SCoPE: Sightline-Coordinate Positional Encoding for Video Diffusion Transformers

Minghao Yin[1*] Jiahao Lu[2*] Wenbo Hu[3†] Wang Zhao[3] Ying Shan[3] Kai Han[1‡]
[1]The University of Hong Kong [2]The Hong Kong University of Science and Technology
[3]ARC Lab, Tencent

Project Page: https://visual-ai.github.io/scope/

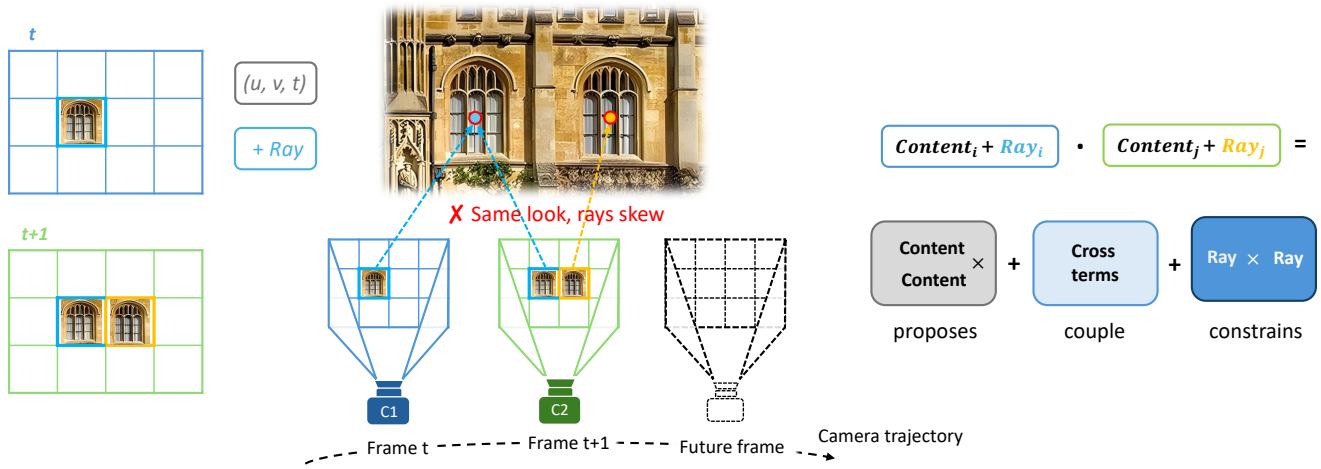

Figure 1. **Camera control as a property of the coordinate system.** SCoPE gives every token a second address, its camera ray (left). Rays are fixed by the user's trajectory while the scene, dashed, is not yet generated (center). Appearance cannot choose between the two identical windows in frame $t{+}1$. Only one of their rays meets the ray from frame t. Added to the pretrained queries and keys, the ray gives the score exactly this test, zero when rays meet, growing with the miss (right). Content proposes matches, ray pairing constrains compatible sightlines, and cross terms couple token features with camera rays.

Abstract

Video diffusion transformers address their tokens by position on the pixel-time grid: an address in the tensor, not in the world. The address we would want, the world point a token depicts, lies on a surface not yet generated, while its camera ray is fixed once the user specifies a trajectory. SCoPE therefore treats the ray as a second positional coordinate, and camera control becomes a property of the coordinate system, not an added module. The ray is added to the pretrained attention's queries and keys, and the score gains a term that reads the two rays alone. Its canonical form, the reciprocal product of line geometry, measures how nearly two lines of sight meet. Normalize-Gate-Inject makes a single encoding trainable across metric and up-to-scale pose sources. The retrofit keeps RoPE bit-exact, starts from the unchanged pretrained DiT, and adds under 0.1% *new pa-*

* Equal contribution † Project lead. ‡ Corresponding author.

rameters. On Wan2.2 at 5B and 14B under matched data and budget, SCoPE improvesevery camera-controllability and fidelity metric, leads all closed-loop revisit metrics, and shows widening margins with model size. At 14B, rotation error falls 29% *and FVD* 43% *below the strongest baseline.*

1. Introduction

A transformer knows only as much about space as its positional encoding supplies. In video diffusion transformers (DiTs) [23, 31, 61, 76], that encoding is typically a factorized RoPE [21, 53] over the (u, v, t) pixel-time grid: *an address in the tensor, not in the world*. Two tokens imaging the same surface from different viewpoints therefore receive positions related only by grid offset, and the model must recover 3D correspondence from appearance alone.

Ideally, each token would sit at the world point it depicts, but that point is where its line of sight meets a surface the model has yet to generate. World-point methods must first reconstruct existing content [7, 63, 73, 79], and new content, as in text-to-video, offers nothing to reconstruct. The line of sight, the token's ray, is another matter. *A ray belongs to the camera, not to the content*. Once the user specifies a trajectory [5, 10, 17], every future token's ray is fixed before a single pixel is synthesized. A ray is admittedly less than a point, fixing the line of sight but not the depth along it. Yet it supplies the constraint appearance alone cannot, since two tokens can depict the same point only if their rays nearly meet (Figure 1). We therefore give each token its ray as a second address, and camera control stops being an auxiliary signal bolted onto the model and becomes *a property of the coordinate system itself.*

A coordinate, though, is just half of a positional encoding. *The other half is the rule by which a pair of coordinates enters the attention score*. On the pixel-time grid, that rule is RoPE [53], whose rotations collapse two absolute indices into their offset, the grid's relative quantity. For camera rays, a line of work running from multi-view synthesis [32, 37, 44] into video diffusion [36, 67, 81] extends this recipe, applying relative camera transforms *multiplicatively* to queries and keys. We build on this line and part ways with it over one question. *What must the geometric score measure?* Deciding whether two tokens could see the same point makes three demands of the score: (i) it must relate individual tokens, not frames; (ii) it must register camera translation, since parallax lives in the baseline; and (iii) it must stay readable from the two rays alone, so one test serves any content. Each demand rules out an established design. Frame-level transforms give every token in a frame the same geometry (i) [32, 36, 37, 44]. Direction-only encodings are blind to translation (ii) [67]. Multiplicative application routes geometry entirely through content, leaving the score with no term that reads the two rays by themselves (iii) [81]. The first two fix the coordinate, and the third asks for a different way into the score.

We present SCoPE (Sightline-Coordinate Positional Encoding), which meets all three demands with *a single addition*. Each token's full Plücker ray $\mathbf{r}$, whose moment carries the translation, is added to its query and key after the pretrained RoPE. Because the ray enters as a summand rather than a transform, the dot product expands into four terms: content with content, the two content–ray cross terms, and ray with ray. The last term reads the two rays and no content at all, the pairing that demand (iii) calls for. Unlike grid indices, rays admit no subtraction, so their relative quantity is not an offset but incidence. Line geometry's classical incidence measure is the *reciprocal product*, bilinear in the two rays and exactly the shape of the new term. It vanishes when two rays meet, or more generally lie in one plane, and grows with how far they miss. Learned projections lift each ray into query-key space, so the pairing is a bilinear form, not a hard-wired formula. It reduces to the reciprocal product when the projections match each ray's direction with the other's moment. The content term identifies candidate matches from visual evidence, while the ray pairing filters matches with compatible sightlines (Figure 1). With the ray as a second coordinate, a token becomes a joint state $z = (x, \mathbf{r})$, combining its learned feature x with its prescribed camera ray $\mathbf{r}$. Scoring content–content and ray–ray compatibility independently imposes a separable score on this joint token state. The cross terms remove this restriction by introducing direct interactions between visual content and viewing geometry.

The moment that carries the translation, however, also carries its scale, set by whatever produced the poses. Poses from metric capture come in meters, while monocular SfM [50] and SLAM [13, 57] recover translation only up to scale, so their unit is a per-clip convention, not a measurement. Prior camera encodings normalize offline to one fixed rule, which costs nothing where the unit is arbitrary. In camera-controlled generation, though, scale is part of the command, and a centimeter dolly should not look like a ten-meter sweep. SCoPE keeps that scale by splitting each ray into a scale-invariant part and an explicit log magnitude, so a change of unit moves one number, not the whole feature. A learned gate reads that magnitude and sets the injection strength for each token, so no single weight serves every convention. Because the backbone normalizes its queries and keys, the injected features receive a matching RMSNorm [80] and enter the score on equal footing. Together, these steps normalize the ray, gate its influence, and inject it, hence the name Normalize-Gate-Inject. The additions total under 0.1% of the parameters, leave RoPE bit-exact, and add no second attention or adapter branch. The injection is zero at initialization, so training starts from the exact

pretrained DiT.

If camera control really is a property of the coordinate system, three things should follow. Adding the ray coordinate, and changing nothing else, should improve control. The improvement should trace to the pairing term the derivation singled out. And it should hold as the backbone grows. We test all three by retrofitting SCoPE onto Wan2.2 [61] at 5B and 14B. Every compared method shares the backbone, data, and training budget, so the camera conditioning is the only thing that differs. All three predictions hold. Control and fidelity improve together. SCoPE leads on every camera-controllability metric and on FVD, FVD_c, and FID. On closed-loop camera trajectories, it also leads all revisit metrics, recovering both the initial camera pose and the conditioning view after the commanded excursion. At 14B, rotation error falls 29% and FVD 43% below the strongest baseline on each metric. Per-term ablations put the largest cost on removing the ray pairing, the very term the multiplicative form does not produce. Without it, rotation error nearly doubles. From 5B to 14B, the margins widen rather than wash out, and the translation-error lead grows from 12% to 25%.

- **Rays as positional coordinates.** A single zero-initialized addition puts each token's Plücker ray into the pretrained queries and keys, giving the score a pure ray pairing whose canonical form is the reciprocal product.
- **Normalize-Gate-Inject.** A scale-invariant split, a per-clip learned gate, and a matched norm confine arbitrary pose units to one number and keep the commanded scale in the encoding.
- **Controlled evidence at scale.** Under matched backbone, data, and budget, SCoPE improves every camera-controllability and fidelity metric, the gains trace to the ray pairing, and the margins grow with model size.

2. Related Work

Camera control by conditioning. Adapter methods attach modules outside the pretrained attention, encoding extrinsics as per-frame tokens [64], rasterizing per-pixel Plücker maps into a ControlNet-style branch [17], concatenating camera embeddings with epipolar cross-view attention [71], or conditioning trainable cross-attention and LoRA modules [3, 4]. Successors condition directly on pose and time [59, 65], steer trajectory or dimension adapters [55, 68, 75], and scale the recipe to dynamic scene exploration [18, 27]. Rendering methods reconstruct a proxy and re-render it along the target trajectory as a pixel-aligned condition [8, 15, 34, 38, 48, 78, 79], or steer a frozen model training-free through warped views and rearranged point clouds [24, 52, 77]. ReCamMaster and multi-camera variants concatenate view tokens and let the native attention discover correspondence on its own [5, 6, 33, 72], while Go-with-the-Flow and Latent-Reframe move the camera into noise and latents, a route orthogonal to ours and combinable with it [10, 86]. These routes pay for control in added modules, extra sequence length, or an upstream reconstruction stage. A related line addresses tokens by world-space coordinates estimated from existing content [7, 63, 73]. The ray needs no reconstruction and is fixed by the trajectory before generation begins.

Camera geometry inside attention. A more direct line moves the camera into the attention operator itself. Epipolar attention did so first as a mask, constraining which pairs may attend [12, 19]. GTA [44] made the geometry an operator instead, applying relative SE(3) transforms multiplicatively to the queries, keys, and values of a multi-view transformer trained from scratch. EscherNet's camera encoding and RePAST follow the same pattern [32, 49]. PRoPE generalizes the transform from extrinsics to full camera frustums while keeping the patch RoPE on separate channels [37], and interactive world models have already adopted it [56]. In video diffusion, UCPE runs GTA-style operators in a parallel zero-initialized bypass branch, at roughly 0.5% added parameters and one extra attention per block [81]. ReRoPE overwrites the low-frequency temporal RoPE phases of a frozen backbone with lifted relative rotations [36]. ViewRope rotates query and key subvectors by viewing-ray direction on the same backbone we use [67]. In each of these, the geometry enters multiplicatively and is routed through content, so the score gains no term that depends on the two rays alone. Scale gets at most a fixed offline rule, near-plane normalization in UCPE, single-convention data in GTA and PRoPE, nothing in ReRoPE. SCoPE stays in this line and changes the geometry's role, from a transform on content to a coordinate of the token, added in the pretrained attention with no branch, no second attention, and a learned gate instead of a fixed rule. The line keeps moving. ReDirector folds cameras into RoPE phase shifts [45], URoPE carries relative embeddings across geometric spaces [69], and DPPE decouples rotation from translation for scaling [30]. CRePE adds depth distributions to UCPE's rays, an upgrade the additive form can adopt unchanged [29], CameraAnything applies ReDirector's rotary phases to intrinsic and extrinsic editing [39], and RayRoPE, no relation despite the name, targets feedforward reconstruction [66].

Positional encodings and ray-space representations. Relative positional encodings enter the score as additive terms in Transformer-XL and T5 [11, 47], or as rotations that collapse indices into offsets in RoPE and its factorized video form [21, 53]. SCoPE extends the additive family to a coordinate whose relative quantity is incidence rather than offset. Rays are also an established representational domain. The plenoptic function and light-field rendering parameterize appearance over rays [1, 14, 35], neural fields from NeRF to light field networks and light-field

transformers learn scene functions on ray inputs [2, 43, 51, 54, 62], and SE(3) equivariant transformers operate in ray space [74]. Camera pose can be estimated as a bundle of rays [82], world models generate autoregressively in ray space [70], and Rays as Pixels folds camera trajectories into the video distribution [28]. These works treat the ray as what the network represents or predicts. SCoPE treats it as where a token stands, a coordinate the pretrained attention compares. Retrofitting favors additions that vanish at initialization, the principle behind ControlNet's zero-initialized residuals [83], and SCoPE carries it into the positional encoding.

3. Background and Notation

This section fixes the notation we use for video diffusion transformers, the camera model, and the Plücker representation of a ray, and recalls the geometric facts that motivate our design.

3.1. Video Diffusion Transformers

We follow the latent-video diffusion / flow-matching setup [9, 22, 42, 61, 76]. A video $V \in \mathbb{R}^{T\times 3\times H\times W}$ is encoded by a 3D VAE into a latent volume $Z \in \mathbb{R}^{T_z\times C\times H_z\times W_z}$, patchified into a sequence of tokens $X \in \mathbb{R}^{N\times d}$ with $N = T_z\cdot H_p\cdot W_p$, where H_p, W_p are the post-patchification spatial sizes. Each token carries an integer index $(u, v, t) \in \{0, \dots, W_p - 1\} \times \{0, \dots, H_p - 1\} \times \{0, \dots, T_z - 1\}$.

A self-attention layer [60] with query/key/value projections $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ produces, for each token i,

$$q_i = W_Q\, x_i, \quad k_i = W_K\, x_i, \quad v_i = W_V\, x_i, \tag{1}$$

and computes the attention output $o_i = \sum_j \mathrm{softmax}_j\big(\frac{1}{\sqrt{d}}\langle q_i, k_j\rangle\big)\, v_j$. Modern video DiTs add (i) RMS-based query/key normalization [20] (often called QKNorm) and (ii) a factorized 3D rotary positional encoding (RoPE) that applies a per-pair rotation to q and k as a function of (u, v, t):

$$\tilde{q}_i = R(u_i, v_i, t_i)\, q_i, \quad \tilde{k}_j = R(u_j, v_j, t_j)\, k_j, \tag{2}$$

where $R(\cdot)$ is a block-diagonal rotation built from per-axis 1D RoPE [21, 53]. The rotation is applied so that the dot product $\langle \tilde{q}_i, \tilde{k}_j\rangle$ depends on $(u_i - u_j, v_i - v_j, t_i - t_j)$, giving the standard relative positional bias on the sampling lattice.

Crucially, (u, v, t) are indices into the camera's sampling grid; they say nothing about *where* in 3D each token's pixel originated. We do not change Eq. (2); SCoPE adds an additive term that is computed from the 3D camera ray of each token.

3.2. Camera Model and Per-Token Rays

Each video clip is annotated with per-frame extrinsics $T_t \in \mathrm{SE}(3)$ (camera-to-world) and per-frame intrinsics $K_t \in \mathbb{R}^{3\times 3}$. We adopt the OpenCV convention (x-right, y-down, z-forward) throughout. For a token whose patch center has pixel coordinate (p_u, p_v) in frame t, the camera-frame ray direction is

$$\mathbf{d}_i^{\mathrm{cam}} \;=\; \frac{K_t^{-1}\,[p_u, p_v, 1]^\top}{\|K_t^{-1}\,[p_u, p_v, 1]^\top\|}, \tag{3}$$

the world-frame ray direction is $\mathbf{d}_i = R_t\,\mathbf{d}_i^{\mathrm{cam}}$ where R_t is the rotation part of T_t, and the world-frame ray origin is $\mathbf{o}_i$, the position of camera t. By construction $\|\mathbf{d}_i\| = 1$.

3.3. Plücker Decomposition Coordinates and Reciprocal Product

A 3D line (or ray) in $\mathbb{R}^3$ is identified by a pair of 3-vectors $(\mathbf{d}, \mathbf{m}) \in \mathbb{R}^6$, where $\mathbf{d}$ is the direction and

$$\mathbf{m} \;=\; \mathbf{o}\times\mathbf{d} \tag{4}$$

is the Plücker moment, with $\mathbf{o}$ any point on the line. Two constraints make $(\mathbf{d}, \mathbf{m})$ a parameterization of an oriented line up to scale: $\|\mathbf{d}\| > 0$ and $\mathbf{d}\cdot\mathbf{m} = 0$. Both are direct consequences of Eq. (4) [16, 46].

Given two rays $\mathbf{r}_i = (\mathbf{d}_i, \mathbf{m}_i)$ and $\mathbf{r}_j = (\mathbf{d}_j, \mathbf{m}_j)$, their *Plücker reciprocal product* (also called Plücker inner product) is

$$\begin{aligned}\langle \mathbf{r}_i,\, \mathbf{r}_j\rangle_{\mathrm{Pl}} &\equiv \mathbf{d}_i\,\cdot\,\mathbf{m}_j + \mathbf{d}_j\,\cdot\,\mathbf{m}_i \\ &= \mathbf{r}_i^\top J\,\mathbf{r}_j, \quad J = \begin{pmatrix} \mathbf{0}_3 & I_3 \\ I_3 & \mathbf{0}_3 \end{pmatrix}.\end{aligned} \tag{5}$$

Three properties of Eq. (5) are central to our design [16, 46]:

- **Bilinearity.** $\langle \mathbf{r}_i,\, \mathbf{r}_j\rangle_{\mathrm{Pl}}$ is linear in $\mathbf{r}_i$ and in $\mathbf{r}_j$ separately; this is the same algebraic shape as the attention dot product $\langle q_i, k_j\rangle$, which is linear in q and k separately.
- **Coplanarity criterion.** $\langle \mathbf{r}_i,\, \mathbf{r}_j\rangle_{\mathrm{Pl}} = 0$ if and only if the two rays are coplanar; equivalently they intersect, are parallel, or coincide. For two rays that observe the *same* 3D point through two different cameras, the two rays meet at that point and so are coplanar, hence $\langle \mathbf{r}_i,\, \mathbf{r}_j\rangle_{\mathrm{Pl}} = 0$.
- **SE(3) invariance.** For $g \in \mathrm{SE}(3)$ acting jointly on $\mathbf{r}_i$ and $\mathbf{r}_j$, $\langle g\cdot\mathbf{r}_i,\, g\cdot\mathbf{r}_j\rangle_{\mathrm{Pl}} = \langle \mathbf{r}_i,\, \mathbf{r}_j\rangle_{\mathrm{Pl}}$; the reciprocal product depends only on the relative geometry of the two rays, not on the choice of world frame.

The bilinear shape of Eq. (5) and the bilinear shape of $\langle q, k\rangle$ are the algebraic match that SCoPE exploits.

3.4. Scale Behavior of Plücker Coordinates

A subtlety of Plücker coordinates that becomes important in practice is their behavior under rescaling of the scene. Suppose all camera positions are rescaled by a factor $s > 0$: $\mathbf{o} \to s\mathbf{o}$. Then $\mathbf{d}$ is unchanged (it is a unit direction) while $\mathbf{m} = \mathbf{o}\times\mathbf{d} \to s\mathbf{m}$. Substituting into Eq. (5) gives

$$\langle \mathbf{r}_i,\, \mathbf{r}_j\rangle_{\mathrm{Pl}} \to s\cdot\langle \mathbf{r}_i,\, \mathbf{r}_j\rangle_{\mathrm{Pl}}, \tag{6}$$

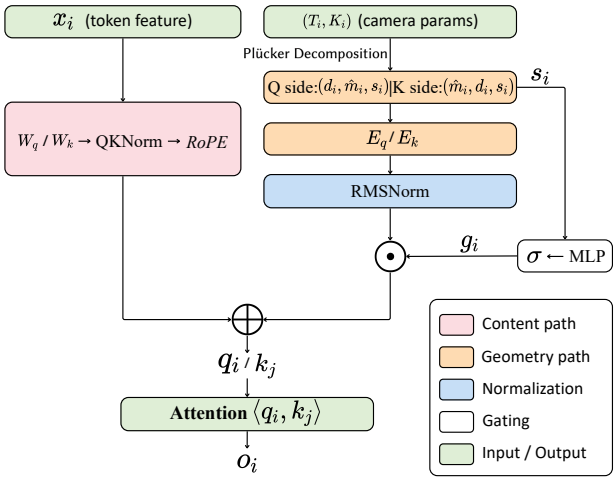

Figure 2. **Applying SCoPE to a self-attention layer.** The content path (pink) preserves the pretrained $W_q/W_k \rightarrow$ QKNorm $\rightarrow$ RoPE pipeline unchanged. The geometry path (orange/blue) computes per-token Plücker coordinates from camera parameters, decomposes them into scale-invariant direction and log-magnitude, projects the resulting features through E_q/E_k, normalizes (RMSNorm), and gates by a learned function of the log-magnitude.

so the reciprocal product scales linearly with translation magnitude. Different videos in the wild come with very different translation scales — normalized scene volumes, metric meters, or DROID-SLAM's internal scale [57] — and a naive raw-Plücker encoding would produce wildly different geometry-channel magnitudes across the dataset. Section 4.2 introduces our Normalize-Gate-Inject step that is designed to address exactly this issue.

4. Method

SCoPE adds a ray-derived positional term to the self-attention layers of a pretrained video diffusion transformer, as illustrated in Figure 2. The method has two parts. First, Plücker Flip PE injects each token's camera ray additively into the query and key, preserving the pretrained RoPE while introducing ray–ray and mixed feature–ray terms into the attention score (Section 4.1). Second, Normalize–Gate–Inject controls the scale and magnitude of this added signal across heterogeneous pose sources (Section 4.2). Implementation details are given in Section 4.3.

4.1. Plücker Flip Positional Encoding

The key observation is that the Plücker reciprocal product and the attention score have the same algebraic form. For two rays $\mathbf{r}_i = (\mathbf{d}_i, \mathbf{m}_i)$ and $\mathbf{r}_j = (\mathbf{d}_j, \mathbf{m}_j)$,

$$\langle \mathbf{r}_i, \mathbf{r}_j \rangle_{\mathrm{Pl}} = \mathbf{d}_i \cdot \mathbf{m}_j + \mathbf{d}_j \cdot \mathbf{m}_i \tag{7}$$

is bilinear in the two rays, while the attention score $\langle q_i, k_j \rangle$ is bilinear in query and key. This suggests adding ray-derived features to q and k in an order that makes the geometric reciprocal product appear as part of the attention score.

Each token i has a camera ray with Plücker coordinate $\mathbf{r}_i = (\mathbf{d}_i, \mathbf{m}_i) \in \mathbb{R}^6$ (Section 3.2). We use two per-layer projections $E_q, E_k \in \mathbb{R}^{d\times 6}$ and add the projected ray features after the usual content projection and RoPE:

$$q_i = R(u_i, v_i, t_i)\, W_Q x_i \;+\; E_q\,(\mathbf{d}_i,\, \mathbf{m}_i), \tag{8}$$
$$k_j = R(u_j, v_j, t_j)\, W_K x_j \;+\; E_k\,(\mathbf{m}_j,\, \mathbf{d}_j). \tag{9}$$

Here we use the shorthand $\tilde{W}_Q x_i \equiv R(u_i, v_i, t_i) W_Q x_i$ and $\tilde{W}_K x_j \equiv R(u_j, v_j, t_j) W_K x_j$ for the RoPE-encoded content query and key, respectively. Two choices matter here. The geometry term is additive, so the pretrained (u, v, t) RoPE remains unchanged. The Plücker halves are also flipped between the two sides: the query receives $(\mathbf{d}, \mathbf{m})$, while the key receives $(\mathbf{m}, \mathbf{d})$. We discuss the role of this flip below; this subsection uses the raw 6D coordinate to make the algebra explicit, and Section 4.2 gives the normalized 7D feature used in practice.

Attention score decomposition. Expanding $\langle q_i, k_j \rangle$ using Eqs. (8)–(9) yields four terms:

$$\begin{aligned}\langle q_i, k_j \rangle = &\underbrace{\langle \tilde{W}_Q x_i,\, \tilde{W}_K x_j \rangle}_{\text{(A) content}} + \underbrace{\langle \tilde{W}_Q x_i,\, E_k \tilde{\mathbf{r}}_j \rangle}_{\text{(B) content}\to\text{geom}} \\ &+ \underbrace{\langle E_q \mathbf{r}_i,\, \tilde{W}_K x_j \rangle}_{\text{(C) geom}\to\text{content}} + \underbrace{\langle E_q \mathbf{r}_i,\, E_k \tilde{\mathbf{r}}_j \rangle}_{\text{(D) geometry}},\end{aligned} \tag{10}$$

where $\tilde{\mathbf{r}}_j = (\mathbf{m}_j, \mathbf{d}_j)$ is the flipped coordinate of token j. Term (A) is the original content attention. Terms (B) and (C) admit the same reading as (D). A dot product against a projected ray is a linear functional on Plücker coordinates, and because the reciprocal product is a nondegenerate pairing, every such functional equals the reciprocal product with a fixed element of line space, a line or, in general, a screw [46]; the flip makes this identification componentwise. Each cross term therefore defines a content-conditioned linear functional over the ray at the other token. These mixed interactions allow the compatibility between two tokens to depend jointly on their learned representations and prescribed viewing rays, rather than on independent content–content and ray–ray scores alone. The terms enter as separate summands, so each can be switched off and its contribution measured (Section 5.4.2). Term (D) is the geometry-only interaction:

$$\text{(D)} = \mathbf{r}_i^\top M\, \tilde{\mathbf{r}}_j, \qquad M = E_q^\top E_k \in \mathbb{R}^{6\times 6}. \tag{11}$$

Flip as a canonical basis. The role of the flip is best stated as a choice of basis on the geometric Q/K projection space rather than an algorithmic necessity:

Proposition 1. *For the raw 6D construction, if* $M = E_q^\top E_k = I_6$*, term (D) becomes*

$$(D) = \mathbf{d}_i \cdot \mathbf{m}_j + \mathbf{m}_i \cdot \mathbf{d}_j = \langle \mathbf{r}_i, \mathbf{r}_j \rangle_{\mathrm{Pl}}.$$

(Proof in Appendix A.4.) For unconstrained learnable E_q, E_k, the flipped and unflipped parameterizations realize the same hypothesis class — any E_k in one form is equivalent to $E_k P$ in the other, where P swaps the $(\mathbf{d}, \mathbf{m})$ blocks (Appendix A.5). We do not claim the flip enlarges the function space the model can express. Its role is to choose a coordinate system in which the symmetric, geometrically meaningful configuration $E_q = E_k = I_6$ corresponds to the textbook Plücker reciprocal product, so that M admits a direct post-hoc reading: $\|M - I_6\|$ measures how far the learned bilinear bias has drifted from the ray-incidence form. The empirical gap between flipped and unflipped configurations is small (Section 5.4.2).

Parameter and compute cost. In the practical NGI version, each layer adds two $d \times 7$ Q/K projections, lightweight gates, and RMSNorms. The per-token projection cost remains $O(N \cdot d)$, which is small compared with the $O(N^2 d)$ cost of attention.

4.2. Normalize-Gate-Inject (NGI)

The flip above defines where geometry enters attention. To use it on mixed real-world data, we still need to control the scale of the injected signal. This is where NGI is used.

The first issue is **scale heterogeneity**. The moment $\mathbf{m} = \mathbf{o} \times \mathbf{d}$ scales linearly with the camera translation magnitude $\|\mathbf{o}\|$. Different datasets use different scale conventions, such as SfM-normalized scenes [50], DROID-SLAM [13, 57] internal scale, or metric meters. Even within one dataset, different baselines can produce moments with very different magnitudes. A raw projection $E\mathbf{r}$ would therefore mix a unit direction $\mathbf{d}$ with a scale-dependent moment $\mathbf{m}$, making the geometry term unstable across batches.

The second issue is **magnitude imbalance** with the content branch. Modern video DiTs apply QKNorm (RMS normalization) to q and k before the dot product, giving the content branch a controlled magnitude. An unnormalized $E\mathbf{r}$ added on top does not respect this normalization. A single global scalar is also too coarse: it cannot handle both per-clip pose scale and the local balance between geometry and content.

NGI handles these two issues in three steps:

Step 1: Direction/magnitude decomposition. We decouple the Plücker coordinate into a scale-invariant directional part and a scalar log-magnitude:

$$\hat{\mathbf{m}} = \mathbf{m} / \max(\|\mathbf{m}\|, \epsilon), \qquad s = \log \max(\|\mathbf{m}\|, \epsilon), \quad (12)$$

where ϵ is a small numerical constant. The 7D geometric input to E_q is $f_i^{(q)} = (\mathbf{d}_i, \hat{\mathbf{m}}_i, s_i)$, and to E_k (after the Q/K flip) is $f_j^{(k)} = (\hat{\mathbf{m}}_j, \mathbf{d}_j, s_j)$. The first six dimensions are invariant to a global rescaling of camera positions; the final dimension retains the log moment magnitude as an explicit pose-scale cue.

Step 2: Scale-aware gating. A small two-layer MLP G_s maps the per-token log-magnitude to a per-channel sigmoid gate:

$$g_i = \sigma\big(G_s(s_i)\big) \in (0, 1)^d, \quad (13)$$

with biases initialized so that $g \approx 0.5$ at $s = 0$. The gate lets the model adjust the geometry injection according to the encoded moment magnitude, rather than using one fixed strength across all pose-scale conventions.

Step 3: Normalize and inject. We project the 7D feature, apply a learnable RMSNorm analogous to the content-side QKNorm, and modulate the result by the gate:

$$\mathrm{pe}_i^q = g_i \odot N_q\big(E_q\, f_i^{(q)}\big), \qquad \mathrm{pe}_j^k = g_j \odot N_k\big(E_k\, f_j^{(k)}\big), \quad (14)$$

where N_q, N_k are per-layer learnable RMSNorms. A scalar α, initialized to zero, controls the overall geometry/content balance:

$$q_i = \mathrm{QKNorm}(\tilde{W}_Q x_i) + \alpha \cdot \mathrm{pe}_i^q, \quad (15)$$

$$k_j = \mathrm{QKNorm}(\tilde{W}_K x_j) + \alpha \cdot \mathrm{pe}_j^k. \quad (16)$$

Since both branches are normalized, α has a consistent meaning across layers and training examples.

Log-scale augmentation. We also apply a simple scale augmentation during training. When computing the gate, we optionally perturb s by a clip-level offset:

$$\tilde{s}_i = s_i + \delta, \quad \delta \sim \mathrm{Uniform}(-1.2,\, 1.6), \text{ with prob. } 0.3, \quad (17)$$

shared across all tokens of a clip. This simulates a global rescaling $\mathbf{o} \to e^\delta \mathbf{o}$ of the camera trajectory without changing the supervision. Only the gate receives $\tilde{s}_i$; the projection $E_q f^{(q)}$ still uses the true s_i.

4.3. Architectural Discussion

Pseudocode. The full one-step procedure is given in Algorithm 1. The per-token Plücker computation, decomposition $(\mathbf{d}, \hat{\mathbf{m}}, s)$, scale gate, and Q/K flip are applied independently in each attention layer.

Relation to alternative camera-aware positional encodings. Unlike camera-aware positional encodings that replace or partition RoPE [36, 37, 81], SCoPE preserves the pretrained 3D RoPE and injects geometric information as an additional Q/K bias term. Quantitative comparisons with these alternatives are provided in Section 5.

Algorithm 1 One forward-backward step of SCoPE on top of a pretrained video DiT.

Require: Latent video z_0, text embedding c, per-frame extrinsics $\{T_t\}$ and intrinsics $\{K_t\}$, dataset id ds, near-depth z_n (or $\perp$).

Apply trajectory rescaling: $\mathbf{o}_t \leftarrow (\eta_{\text{ds}}/\max(z_n, \epsilon)) \cdot \mathbf{o}_t$ ▷ Appendix C.2–C.3
Compute per-token rays $\{(\mathbf{d}_i, \mathbf{m}_i)\}$ from $\{T_t, K_t\}$ and the latent patch grid ▷ Section 3.2
Sample noise $\epsilon \sim \mathcal{N}(0, I)$, timestep τ, build noisy latent z_τ
for each DiT block $\ell = 1 \dots L$ **do**
    $q, k, v \leftarrow W_Q^\ell h,\ W_K^\ell h,\ W_V^\ell h$ ▷ Pretrained projections, possibly fine-tuned
    $q, k \leftarrow \text{QKNorm}(q),\ \text{QKNorm}(k)$
    $q, k \leftarrow R(u, v, t) \cdot q,\ R(u, v, t) \cdot k$ ▷ Standard 3D RoPE
    $f^{(q)} \leftarrow (\mathbf{d}, \hat{\mathbf{m}}, s),\ f^{(k)} \leftarrow (\hat{\mathbf{m}}, \mathbf{d}, s)$ where $s = \log \|\mathbf{m}\|$
    $g \leftarrow \sigma(G_s^\ell(\tilde{s}))$ ▷ $\tilde{s}$ optionally aug. per Eq. (17)
    $q \leftarrow q + \alpha \cdot g \odot N_q^\ell(E_q^\ell f^{(q)})$
    $k \leftarrow k + \alpha \cdot g \odot N_k^\ell(E_k^\ell f^{(k)})$
    $h \leftarrow h + \text{Attn}(q, k, v)\, W_O^\ell$
    $h \leftarrow h + \text{CrossAttn}(h, c) + \text{FFN}(h)$
end for
Compute diffusion / flow loss between predicted noise/velocity and target and backpropagate.

5. Experiments

We evaluate SCoPE along three axes: (i) video generation quality, (ii) camera-trajectory controllability, and (iii) cross-frame geometric consistency. We compare SCoPE with four camera-conditioned video generation baselines and four alternative camera-aware positional encoding designs on the same backbone, isolating the contribution of the proposed Plücker-flip/NGI formulation.

5.1. Setup

Backbone. For the main comparisons (Table 1), we evaluate our method on two official variants from Wan-2.2 [61]: the 5B TI2V model and the 14B I2V model. To ensure computational efficiency, all ablation studies are conducted exclusively using the Wan-2.2-TI2V-5B model. Across all runs, videos are generated at 480×832 resolution with $T = 81$ frames ($T_z = 21$ latent frames), and the models are trained on the four-dataset mixture (Appendix C). All variants share their respective pretrained weights; only the newly integrated camera-conditioning module differs.

Baselines. We compare against five external methods: CameraCtrl [17] (Plücker maps via ControlNet encoder), ReCamMaster [5] (video-to-video re-rendering with camera control), ProPE [37] (deployed via a zero-initialized parallel branch), UCPE [81] (full camera geometry via spatial-attention adapter), and ReRoPE [36] (low-frequency temporal RoPE replaced by relative camera pose). Where official checkpoints are unavailable we re-implement on our backbone under the same data and budget (Appendix A). We additionally train four *FreqSplit-RoPE* variants that replace the low-frequency half of temporal RoPE *multiplicatively* with (i) a 4×4 projective transform, (ii) camera-plane near/far UV coordinates, (iii) learned Plücker-to-phase mapping, or (iv) analytical (θ, ϕ, u, v) phases. All four share our backbone and budget; SCoPE differs by adding Plücker features *additively* without modifying RoPE.

Evaluation. We evaluate on a held-out RE10K [84] split (image-to-video: first GT frame + GT trajectory) and report eight metrics. *Quality:* **CLIP**↑ (inter-frame similarity). *Camera controllability:* **RotErr**↓ (geodesic rotation error), **TransErr**↓ (L2 translation error), **CamMC**↓ (per-frame pose Frobenius norm), **ATE**↓ (RMS absolute trajectory error), all on per-frame poses anchored to frame 0. *Distribution:* **FVD**↓ / **FVD**$_c$↓ (full-frame / center-cropped Fréchet Video Distance [58]), **FID**↓ (per-frame Fréchet Inception Distance). Trajectories are estimated from both GT and generated videos using ViPE [25] and compared *without rescaling*. Unlike the common protocol of normalizing trajectories by their maximum translation magnitude, we preserve absolute-scale fidelity. This avoids artificially inflating the apparent controllability of methods that produce only small motions. Results under the conventional rescaled protocol are provided in the Appendix; the ranking is unchanged. All methods use 50 denoising steps with CFG scale 5.0 (text) / 1.0 (camera), matching Appendix C.4.

5.2. Main Comparison

Table 1 reports the main quantitative comparison and Figures 3–8 show qualitative results. Consistent with the design motivation in Sections 4.1 and 4.2, SCoPE improves all four pose-error metrics substantially while also improving CLIP, FVD, FVD$_c$, and FID. We organize the qualitative evaluation along three progressively harder axes. *In-domain comparison* (Figure 3): on the RE10K validation split, we show the same clip generated by ReCamMaster, UCPE, and SCoPE under identical camera conditioning; SCoPE follows the target trajectory more faithfully across frames while the baselines drift. *In-domain gallery* (Figure 7): a broader set of RE10K scenes where each row is a different first frame driven by a distinct trajectory, illustrating consistent camera following across indoor walkthroughs, outdoor pans, and forward-moving corridors. *Out-of-distribution generalization* (Figures 5, 6, and 8): we test on inputs that never appear in the training data. Figure 5 compares baselines on artistic paintings, where SCoPE realizes the intended camera motion with correct parallax while base-

Figure 3. Qualitative comparison on RealEstate10K. Each row shows frames from the same clip generated under the same target camera trajectory. Top to bottom: ground truth, ReCamMaster [5], UCPE [81], and SCoPE (ours). SCoPE follows the target trajectory more faithfully across frames; ReCamMaster and UCPE drift from the camera path.

	Quality	Camera controllability				Distribution		
Method	CLIP↑	RotErr↓	TransErr↓	CamMC↓	ATE↓	FVD↓	FVD_c ↓	FID↓
Wan-2.2 5B Scale								
CameraCtrl [17]	25.04	0.152	1.292	1.355	1.501	824.37	805.62	64.08
ProPE [37]	24.90	0.117	1.653	1.694	1.915	717.2	611.5	58.33
ReCamMaster [5]	24.97	0.131	1.226	1.279	1.460	874.30	890.52	62.53
ReRoPE [36]	25.20	0.137	1.109	1.215	1.350	684.57	650.31	60.77
UCPE [81]	25.39	0.113	0.856	0.909	0.990	703.41	755.83	61.50
SCoPE (ours)	**26.05**	**0.085**	**0.751**	**0.802**	**0.884**	**543.17**	**588.62**	**57.83**
Wan-2.2 14B Scale								
ReCamMaster [5]	25.31	0.109	0.976	1.012	0.995	675.23	697.10	59.21
ReRoPE [36]	25.85	0.114	0.820	0.905	0.859	493.30	525.61	49.18
UCPE [81]	25.72	0.082	0.693	0.760	0.788	529.42	558.70	54.75
SCoPE (ours)	**26.30**	**0.058**	**0.517**	**0.530**	**0.605**	**280.17**	**354.52**	**41.01**

Table 1. Main comparison on the RE10K held-out split across different backbone scales. Higher is better for CLIP; lower is better for the rest. Pose metrics (RotErr, TransErr, CamMC, ATE) are computed on raw, unscaled ViPE trajectories estimated identically on both GT and generated videos (Section 5.1); rescaled-alignment results matching the conventional protocol are in Appendix A.

lines fail to follow the trajectory. Figure 6 extends this to a wider set of hand-painted stylized first frames, showing that SCoPE preserves artistic style while accurately executing the prescribed camera path. Figure 8 fixes two cinematic movie-still first frames and drives each through several distinct trajectories, demonstrating that the geometric encoding generalizes across both content domain and trajectory diversity. Additional evaluations of dynamic-scene quality and inference efficiency are provided in Appendix B.1 and B.3, respectively.

5.3. Closed-Loop Revisit Consistency

To evaluate long-horizon closed-loop consistency and view recovery, we introduce a Revisit protocol in which each method starts from the same conditioning image, moves away from the initial view, and finally returns to the starting pose. For each conditioning image, we generate four closed-loop camera paths: truck, orbit, dolly, and a non-palindromic loop. We evaluate an away window near the maximum camera excursion (frames 36–40) and a return window after the loop is completed (frames 76–80). Camera trajectories are estimated with ViPE [25] and expressed relative to frame 0.

Return Translation Ratio is the mean camera-center displacement from frame 0 over the return window, divided by the maximum displacement over the full generated trajectory. **Return Rotation** is the mean geodesic rotation from the initial orientation over the return window. **Normalized ATE** compares the predicted and commanded trajectories after anchoring each to frame 0 and independently normalizing it by its maximum translation magnitude. For appearance consistency, **Return LPIPS** and **DINOv2 Similarity** compare the conditioning image with the frames in the return window, using AlexNet LPIPS and DINOv2 global-feature cosine similarity, respectively. We further define **LPIPS Recovery** $= 1 - \mathrm{LPIPS}_{\mathrm{return}}/\mathrm{LPIPS}_{\mathrm{away}}$,

Figure 4. **Closed-loop revisit comparison.** Starting from the same reference image (Ref), each method follows the same camera loop, moving away from and then returning to the initial view. Right: input camera trajectory of each generated video.

	Camera return			Appearance recovery		
Method	Return Trans. Ratio ↓	Return Rot. (°) ↓	Norm. ATE ↓	Return LPIPS ↓	LPIPS Recovery ↑	DINOv2 Sim. ↑
ProPE [37]	0.784	4.90	0.533	0.603	−0.176	0.874
ReCamMaster [5]	0.859	3.80	0.625	0.653	0.094	0.890
UCPE [81]	0.433	3.42	0.391	0.403	0.102	0.910
SCoPE (ours)	**0.090**	**1.89**	**0.275**	**0.230**	**0.587**	**0.952**

Table 2. Revisit evaluation on the test set. Pose metrics measure whether the generated camera returns to its initial pose, while appearance metrics measure recovery of the conditioning view.

Figure 5. Out-of-distribution qualitative comparison. Inputs are artistic paintings (rather than natural photographs), conditioned on simple camera trajectories such as a forward dolly. No ground-truth video exists in this setting, so each row shows only the generated frames of a single method. SCoPE realizes the intended camera motion (e.g., zoom-in with correct foreground/ background parallax) on stylized inputs, while ReCamMaster [5] and UCPE [81] fail to follow the trajectory.

which measures the fraction of the appearance discrepancy at maximum excursion that is recovered after returning. Window-based scores are first averaged over the corresponding five frames, then over the four trajectories of each conditioning image, and finally across cases. Appearance metrics are interpreted jointly with the pose metrics, since similarity to the conditioning image alone does not demonstrate that the prescribed camera motion was executed.

As shown in Table 2, SCoPE ranks first on all six metrics. Relative to the strongest baseline for each metric, it reduces Return Translation Ratio by 79%, Return Rotation by 45%, Normalized ATE by 30%, and Return LPIPS by 43%. It also raises LPIPS Recovery from 0.102 to 0.587 and DINOv2 Similarity from 0.910 to 0.952. The simultaneous gains in pose return and revisited-view fidelity indicate that SCoPE recovers the initial view after executing the commanded excursion, rather than preserving appearance by suppressing camera motion.

Figure 4 illustrates the resulting behavior: all methods are driven away from the same conditioning view by the same closed-loop command. ProPE and ReCamMaster accumulate substantial return error, while UCPE only partially restores the initial view. In contrast, SCoPE returns both the camera and the generated content closest to their

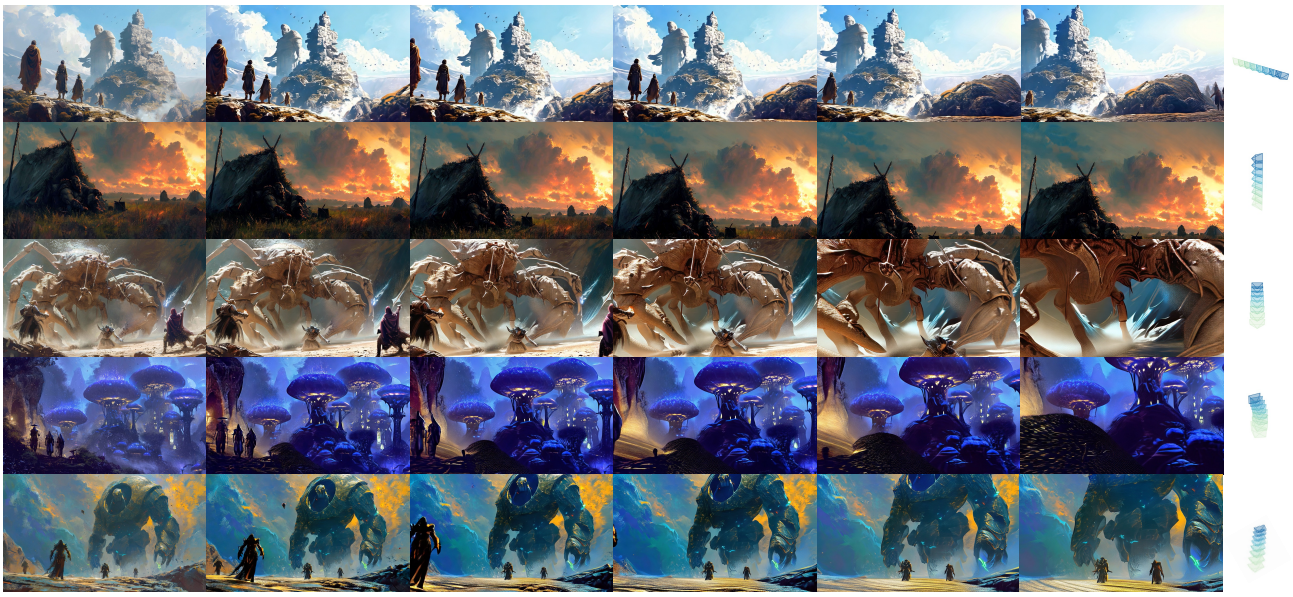

Figure 6. Generalization to stylized inputs. Each row is a different hand-painted concept-art first frame driven by a distinct target camera trajectory with SCoPE; frames go left to right along time, and the rightmost column shows the input trajectory as a stack of camera frusta.

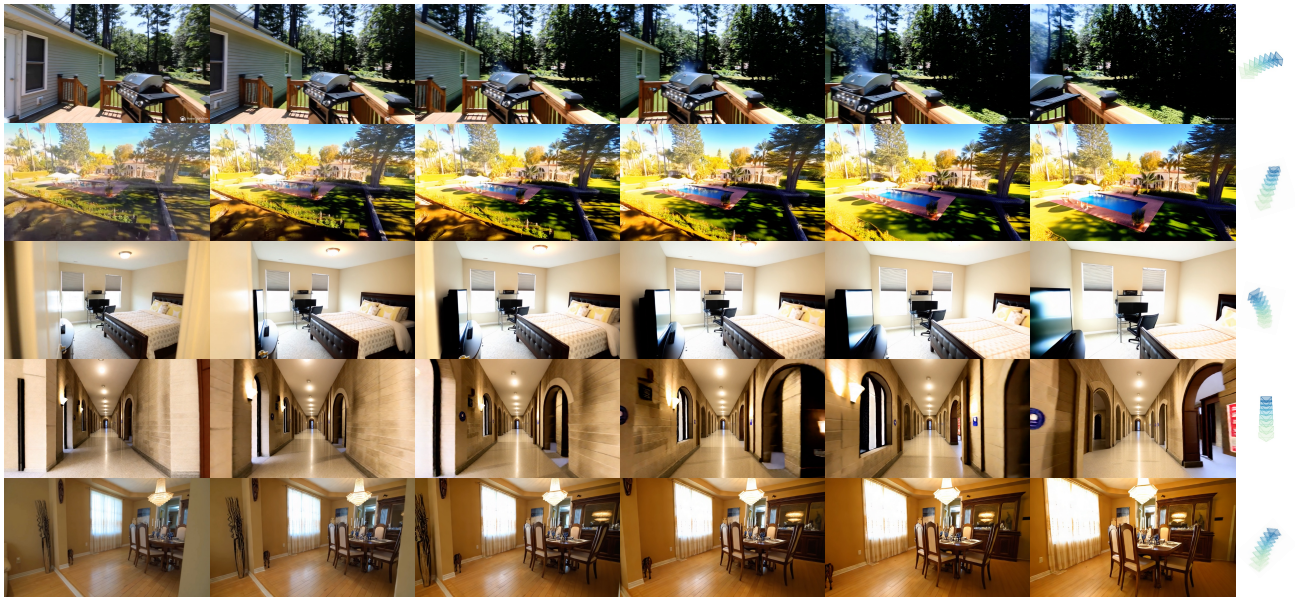

Figure 7. Gallery of SCoPE on diverse scenes. Each row is a different first frame driven by a distinct target camera trajectory; frames go left to right along time, and the rightmost column shows the input trajectory as a stack of camera frusta.

initial states.

5.4. Ablations

We organize ablations into two complementary tables. The first contrasts SCoPE against four alternative camera-aware PE designs that share our backbone (Section 5.4.1). The second removes one component at a time from SCoPE, isolating the contribution of each design element (Section 5.4.2).

5.4.1. Camera-Aware PE Design Space

A direct multiplicative camera PE would replace the pretrained RoPE across all frequency channels. In our experiments, this full-frequency replacement failed to yield a usable camera-controlled model under the matched finetuning budget, suggesting disruption of the pretrained positional prior. We therefore follow the frequency-splitting strategy of ReRoPE [36]: each alternative retains the high-frequency half of temporal RoPE and replaces its low-frequency half with a camera-derived transformation. This

Figure 8. Multi-trajectory gallery on cinematic out-of-distribution inputs. We fix two movie-still first frames and drive each through several distinct camera trajectories (one trajectory per row within a scene), all generated by SCoPE.

	Quality	Camera controllability				Distribution		
PE design	CLIP↑	RotErr↓	TransErr↓	CamMC↓	ATE↓	FVD↓	FVD_c ↓	FID↓
FreqSplit-RoPE + 4×4 Proj.	25.09	0.175	1.227	1.308	1.584	675.03	723.90	62.98
FreqSplit-RoPE + Camera-UV	25.39	0.141	1.205	1.394	1.602	685.20	720.11	60.74
FreqSplit-RoPE + Plücker-RoPE	25.61	0.159	1.018	1.323	1.447	671.45	659.03	59.31
FreqSplit-RoPE + 4DOF Plücker	25.52	0.137	0.976	1.228	1.419	638.31	645.29	58.54
SCoPE (ours)	**26.05**	**0.085**	**0.751**	**0.802**	**0.884**	**543.17**	**588.62**	**57.83**

Table 3. Camera-aware PE design space on RE10K. All rows share the Wan2.2-TI2V-5B backbone, the four-dataset training mixture and the same training budget; the only difference is how the camera enters attention. SCoPE adds an *additive* 6D Plücker term to q, k, while the four FreqSplit-RoPE variants *multiplicatively replace* the low-frequency half of the temporal RoPE.

partial replacement provides a stable common scaffold for comparing different multiplicative camera encodings.

Within this common FreqSplit interface, we test four ways of representing camera geometry: a 4×4 projective transform, near/far camera-plane coordinates, learned rotary phases from the full 6D Plücker ray, and analytical phases from its four intrinsic degrees of freedom. All variants share the same backbone, data, training budget, and frequency allocation. This controlled design study asks whether a stable multiplicative retrofit, equipped with either projective or ray-level geometry, can recover the benefit of SCoPE. The Plücker-RoPE variant provides the closest comparison: it receives the same full ray coordinate as SCoPE, but maps that coordinate to multiplicative rotary phases in the replaced frequency bands.

SCoPE takes a different retrofit route. It preserves every channel of the pretrained RoPE and adds a zero-initialized Plücker term to q and k. It therefore starts from the exact pretrained attention operator while introducing the content–geometry cross terms and geometry-only pairing derived in Section 4.1. As shown in Table 3, SCoPE outperforms all four viable multiplicative alternatives on every metric. The gap to Plücker-RoPE is especially informative: using the same ray coordinate through rotary phases is not sufficient to reproduce the gain of the additive, RoPE-preserving formulation.

Variant	CLIP↑	RotErr↓	CamMC↓	ATE↓	FVD↓
Full SCoPE	**26.05**	**0.085**	**0.802**	**0.884**	**543.17**
w/o Q/K flip	25.93	0.091	0.817	0.929	579.83
w/o Normalize-Gate-Inject	25.86	0.086	0.865	0.933	560.37
w/o PE RMSNorm	25.62	0.090	0.874	0.913	601.39
w/o log-scale aug.	25.94	0.083	0.823	0.890	557.92
w/o zero init	25.68	0.086	0.837	0.899	635.42
with V transform	25.57	0.090	0.855	0.949	724.10
w/o (B)+(C) content↔geometry	25.21	0.135	1.174	1.358	695.41
w/o (D) geometry↔geometry	24.08	0.159	1.382	1.440	733.08

Table 4. Component ablation of SCoPE on RE10K. "Full" is the configuration used for the main comparison. Each subsequent row removes a single component while keeping the rest unchanged.

5.4.2. Component Ablation

Table 4 reports component ablations of the full SCoPE configuration. Each row removes a single component while keeping all other settings and training recipes unchanged.

What each row tests. The **w/o Q/K flip** row sends $(\mathbf{d}, \mathbf{m})$ on both sides instead of the flipped pair; under arbitrary learnable E_q, E_k the flip is a reparametrization (Appendix A.5), so only a marginal change is expected, which the row confirms. **w/o Normalize-Gate-Inject** reduces the encoding to a raw 6D Plücker projection $E\mathbf{r}$, removing direction/magnitude decomposition, scale gating, and PE RMSNorm together — the largest aggregate ablation of the normalize-gate-inject pipeline. **w/o PE RM-**

SNorm keeps decomposition and gating but drops the PE-side RMSNorm, so the geometry magnitude is no longer matched to the content branch and a single α cannot balance both. **w/o log-scale aug.** removes random log-magnitude augmentation during training, isolating the role of synthetic scale diversity in stabilizing cross-domain generalization; the modest drop suggests the augmentation helps but is not the dominant ingredient. **w/o zero init** initializes E_q, E_k randomly instead of to identity (zero residual at $t = 0$), forcing the model to recover from a non-trivial perturbation of the pretrained attention; the larger FVD drop shows that breaking the pretrained equilibrium at the start of training is costly even when the final solution is ultimately reachable. **with V transform** additionally injects geometry into v, which consistently underperforms Q/K-only injection and confirms that ray geometry is most useful as an attention-bias positional signal rather than as a value-channel feature. The last two rows directly intervene on the attention decomposition of Eq. (10): **w/o (B)+(C) content↔geometry** keeps the content–content (A) and geometry–geometry (D) terms but suppresses the two cross-modal terms that couple appearance with ray geometry, while **w/o (D) geometry↔geometry** removes the pure geometric prior. Both produce sharp drops on every metric.

Complementary roles of mixed and geometric terms. The final two ablations separate the pure ray–ray interaction from the mixed feature–ray interactions in Eq. (10). Term (A) is the standard content–content attention term, while term (D) evaluates the two camera rays. Their sum has the separable form

$$A(x_i, x_j) + D(\mathbf{r}_i, \mathbf{r}_j),$$

where visual affinity and camera compatibility are computed by two independent pairwise functions. For a fixed query ray, the relative preference induced by (D) over candidate rays is therefore independent of the query feature.

Terms (B) and (C) introduce the mixed dependence between token features and camera rays. Let $c_i = \tilde{W}_Q x_i$. The part of the attention logit that depends on the candidate ray can be written as

$$\text{(B)} + \text{(D)} = \left(E_k^\top c_i + M^\top \mathbf{r}_i\right)^\top \tilde{\mathbf{r}}_j.$$

Here, $M^\top \mathbf{r}_i$ gives the ray-dependent coefficient from term (D), while $E_k^\top c_i$ adds a coefficient determined by the query feature. The geometry-related preference over candidate rays can consequently vary with the current query representation, instead of being determined by camera geometry alone. Term (C) provides the complementary interaction, allowing the relevance of a key feature to depend directly on the query ray. Together, (B) and (C) remove the separability between visual content and camera geometry, allowing the geometry-involving contribution to attention to be conditioned on the current token representations.

Training mixture	CLIP↑	RotErr↓	CamMC↓	ATE↓	FVD↓
RE10K only	25.58	0.091	0.897	0.983	586.45
RE10K + DL3DV	25.65	0.089	0.872	0.960	602.13
+ PanShot	25.93	**0.083**	0.836	0.905	550.19
+ OmniWorld (full)	**26.05**	0.085	**0.802**	**0.884**	**543.17**

Table 5. Effect of training-data composition on RE10K-test metrics.

The ablation supports these complementary roles. Removing (B)+(C) degrades RotErr from 0.085 to 0.135, CamMC from 0.802 to 1.174, ATE from 0.884 to 1.358, and FVD from 543.17 to 695.41. Thus, a scene-independent ray prior combined with the original content score does not recover the performance of their explicit mixed interactions. Removing (D) causes a further degradation, confirming that the ray–ray prior remains the primary geometric component. The full model benefits from both: (D) supplies a shared camera-geometric compatibility, while (B) and (C) adapt its use to the representations formed for the current video.

5.4.3. Data composition

Table 5 ablates the effect of training-data composition. We compare training on RE10K alone, RE10K+DL3DV, RE10K+DL3DV+PanShot, and the full RE10K+DL3DV+PanShot+OmniWorld mixture. This highlights the value of *normalize-gate-inject* as more scale-heterogeneous data is added: while methods lacking scale-invariance may degrade as the trajectory-scale distribution widens, SCoPE consistently benefits from the additional data.

6. Conclusion

We presented SCoPE, a sightline-coordinate positional encoding for video diffusion transformers that exploits the algebraic match between the Plücker reciprocal product and the attention dot product. By adding per-token Plücker coordinates to queries and keys with a flip arrangement, SCoPE makes the attention score contain a whose canonical configuration recovers the Klein form, giving the model a built-in 3D inductive bias before any learning. Normalize-Gate-Inject further decouples ray direction from translation magnitude and gates the encoding adaptively, making it stable across datasets with heterogeneous pose scales. The full module adds less than 0.1% parameters to a pretrained 5B video DiT, preserves the original RoPE, and requires no adapter or architectural change. Quantitative and qualitative results show that SCoPE improves camera controllability and cross-view and closed-loop consistency while preserving the generation quality of the pretrained backbone.

References

[1] Edward H. Adelson and James R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*. MIT Press, 1991. 3

[2] Benjamin Attal, Jia-Bin Huang, Michael Zollhöfer, Johannes Kopf, and Changil Kim. Learning neural light fields with ray-space embedding networks. In *CVPR*, 2022. 4

[3] 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*, 2025. 3

[4] Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, David B. Lindell, and Sergey Tulyakov. VD3D: Taming large video diffusion transformers for 3d camera control. In *ICLR*, 2025. 3

[5] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, and Di Zhang. ReCamMaster: Camera-controlled generative rendering from a single video. In *ICCV*, 2025. 2, 3, 7, 8, 9, 17, 18

[6] Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Xiao Fu, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di Zhang. Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints. In *ICLR*, 2025. 3

[7] Yunpeng Bai, Haoxiang Li, and Qixing Huang. Positional encoding field. In *ICLR*, 2026. 2, 3

[8] Weikang Bian, Zhaoyang Huang, Xiaoyu Shi, Yijin Li, Fu-Yun Wang, and Hongsheng Li. GS-DiT: Advancing video generation with dynamic 3D gaussian fields through efficient dense 3D point tracking. In *CVPR*, 2025. 3

[9] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv:2311.15127*, 2023. 4

[10] Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, Michael Ryoo, Paul Debevec, and Ning Yu. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In *CVPR*, 2025. 2, 3

[11] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *ACL*, 2019. 3

[12] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *CVPR*, 2023. 3

[13] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics and Automation Magazine*, 2006. 2, 6

[14] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The Lumigraph. In *SIGGRAPH*, 1996. 3

[15] Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, Wenping Wang, and Yuan Liu. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *SIGGRAPH*, 2025. 3

[16] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 4, 16

[17] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for video diffusion models. In *ICLR*, 2025. 2, 3, 7, 8

[18] 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. In *ICCV*, 2025. 3

[19] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *CVPR*, 2020. 3

[20] Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of EMNLP*, 2020. 4

[21] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *ECCV*, 2024. 2, 3, 4

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 4

[23] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *NeurIPS*, 2022. 2

[24] Chen Hou and Zhibo Chen. Training-free camera control for video generation. In *ICLR*, 2025. 3

[25] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixé, and Sanja Fidler. Vipe: Video pose engine for 3d geometric perception. *arXiv:2508.10934*, 2025. 7, 8

[26] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 17

[27] Zhening Huang, Hyeonho Jeong, Xuelin Chen, Yulia Gryaditskaya, Tuanfeng Y. Wang, Joan Lasenby, and Chun-Hao Huang. Spacetimepilot: Generative rendering of dynamic scenes across space and time. In *CVPR*, 2026. 3

[28] Wonbong Jang, Shikun Liu, Soubhik Sanyal, Juan Camilo Perez, Kam Woh Ng, Sanskar Agrawal, Juan-Manuel Perez-Rua, Yiannis Douratsos, and Tao Xiang. Rays as pixels: Learning a joint distribution of videos and camera trajectories. In *ICML*, 2026. 4

[29] Seonghyun Jin, Youngmin Kim, Sunwoo Park, and Jong Chul Ye. CRePE: Curved ray expectation positional encoding for unified-camera-controlled video generation. *arXiv:2605.12938*, 2026. 3

[30] Shun Kenney and Teppei Suzuki. DPPE: Rethinking camera-based positional encoding for scaling multi-view transformers. *arXiv:2606.31585*, 2026. 3

[31] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, et al. Hunyuan-Video: A systematic framework for large video generative models. *arXiv:2412.03603*, 2024. 2

[32] Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J. Davison. EscherNet: A generative model for scalable view synthesis. In *CVPR*, 2024. 2, 3

[33] Zhengfei Kuang, Shengqu Cai, Hao He, Yinghao Xu, Hongsheng Li, Leonidas Guibas, and Gordon Wetzstein. Collaborative video diffusion: Consistent multi-video generation with camera control. In *NeurIPS*, 2024. 3

[34] Yao-Chih Lee, Zhoutong Zhang, Jiahui Huang, Jui-Hsien Wang, Joon-Young Lee, Jia-Bin Huang, Eli Shechtman, and Zhengqi Li. Generative video motion editing with 3d point tracks. In *CVPR*, 2026. 3

[35] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH*, 1996. 3

[36] Chunyang Li, Yuanbo Yang, Jiahao Shao, Hongyu Zhou, Katja Schwarz, and Yiyi Liao. ReRoPE: Repurposing RoPE for relative camera control. *arXiv:2602.08068*, 2026. 2, 3, 6, 7, 8, 10, 17

[37] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *NeurIPS*, 2025. 2, 3, 6, 7, 8, 9, 18

[38] Teng Li, Guangcong Zheng, Rui Jiang, Shuigen Zhan, Tao Wu, Yehao Lu, Yining Lin, Chuanyun Deng, Yepan Xiong, Min Chen, Lin Cheng, and Xi Li. Realcam-i2v: Real-world image-to-video generation with interactive complex camera control. In *ICCV*, 2025. 3

[39] Yixuan Li, Yanhong Zeng, Ka Leong Cheng, Jiayi Zhu, Hanlin Wang, Wen Wang, Yihao Meng, Hao Ouyang, Qiuyu Wang, Yue Yu, ZiDong Wang, Yiyuan Zhang, Yujun Shen, and Dahua Lin. CameraAnything: Refilming videos with arbitrary camera control. *arXiv:2607.24591*, 2026. 3

[40] Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024. 18

[41] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024. 19

[42] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023. 4

[43] 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. In *ECCV*, 2020. 4

[44] Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. GTA: A geometry-aware attention mechanism for multi-view transformers. In *ICLR*, 2024. 2, 3

[45] Byeongjun Park, Byung-Hoon Kim, Hyungjin Chung, and Jong Chul Ye. ReDirector: Creating any-length video retakes with rotary camera encoding. In *CVPR*, 2026. 3

[46] Helmut Pottmann and Johannes Wallner. *Computational Line Geometry*. Springer, 2001. 4, 5, 16

[47] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020. 3

[48] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *CVPR*, 2025. 3

[49] Aleksandr Safin, Daniel Duckworth, and Mehdi S. M. Sajjadi. RePAST: Relative pose attention scene representation transformer. *arXiv:2304.00947*, 2023. 3

[50] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2, 6, 19

[51] Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Frédo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *NeurIPS*, 2021. 4

[52] Chenxi Song, Yanming Yang, Tong Zhao, Ruibo Li, and Chi Zhang. WorldForge: Taming video models for 3d and 4d generation via zero-shot camera control. In *CVPR*, 2026. 3

[53] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024. 2, 3, 4

[54] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *CVPR*, 2022. 4

[55] Wenqiang Sun, Shuo Chen, Fangfu Liu, Zilong Chen, Yueqi Duan, Jun Zhang, and Yikai Wang. Dimensionx: Create any 3d and 4d scenes from a single image with controllable video diffusion. In *ICCV*, 2025. 3

[56] Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv:2512.14614*, 2025. 3

[57] Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In *NeurIPS*, 2021. 2, 5, 6, 19

[58] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation. In *ICLR Workshops*, 2019. 7

[59] Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *ECCV*, 2024. 3

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4

[61] 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:2503.20314*, 2025. 2, 3, 4, 7, 19

[62] Huan Wang, Jian Ren, Zeng Huang, Kyle Olszewski, Menglei Chai, Yun Fu, and Sergey Tulyakov. R2L: Distilling neural radiance field to neural light field for efficient novel view synthesis. In *ECCV*, 2022. 4

[63] Yifan Wang and Tong He. Warp-as-history: Generalizable camera-controlled video generation from one training video. *arXiv:2605.15182*, 2026. 2, 3

[64] 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 *SIGGRAPH*, 2024. 3

[65] Daniel Watson, Saurabh Saxena, Lala Li, Andrea Tagliasacchi, and David J. Fleet. Controlling space and time with diffusion models. In *ICLR*, 2025. 3

[66] Yu Wu, Minsik Jeon, Jen-Hao Rick Chang, Oncel Tuzel, and Shubham Tulsiani. RayRoPE: Projective ray positional encoding for multi-view attention. In *ECCV*, 2026. 3

[67] Chendong Xiang, Jiajun Liu, Jintao Zhang, Xiao Yang, Zhengwei Fang, Shizun Wang, Zijun Wang, Yingtian Zou, Hang Su, and Jun Zhu. Geometry-aware rotary position embedding for consistent video world model. *arXiv:2602.07854*, 2026. 2, 3

[68] Zeqi Xiao, Wenqi Ouyang, Yifan Zhou, Shuai Yang, Lei Yang, Jianlou Si, and Xingang Pan. Trajectory attention for fine-grained video motion control. In *ICLR*, 2025. 3

[69] Yichen Xie, Depu Meng, Chensheng Peng, Yihan Hu, Quentin Herau, Masayoshi Tomizuka, and Wei Zhan. URoPE: Universal relative position embedding across geometric spaces. In *ECCV*, 2026. 3

[70] Yichen Xie, Chensheng Peng, Mazen Abdelfattah, Yihan Hu, Jiezhi Yang, Eric Higgins, Ryan Brigden, Masayoshi Tomizuka, and Wei Zhan. RAYNOVA: Scale-temporal autoregressive world modeling in ray space. In *CVPR*, 2026. 4

[71] Dejia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. CamCo: Camera-controllable 3D-consistent image-to-video generation. *arXiv:2406.02509*, 2024. 3

[72] Dejia Xu, Yifan Jiang, Chen Huang, Liangchen Song, Thorsten Gernoth, Liangliang Cao, Zhangyang Wang, and Hao Tang. Cavia: Camera-controllable multi-view video diffusion with view-integrated attention. In *ICML*, 2025. 3

[73] Tianxing Xu, Zixuan Wang, Guangyuan Wang, Li Hu, Zhongyi Zhang, Peng Zhang, Bang Zhang, and Songhai Zhang. UCM: Unified modeling of camera control and memory with time-aware positional encoding warping for world models. *arXiv:2602.22960*, 2026. 2, 3

[74] Yinshuang Xu, Jiahui Lei, and Kostas Daniilidis. SE(3) equivariant convolution and transformer in ray space. In *NeurIPS*, 2023. 4

[75] Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Di Zhang, Xiaodong Chen, and Jing Liao. Direct-a-video: Customized video generation with user-directed camera movement and object motion. In *SIGGRAPH*, 2024. 3

[76] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025. 2, 4

[77] Meng You, Zhiyu Zhu, Hui Liu, and Junhui Hou. Nvs-solver: Video diffusion model as zero-shot novel view synthesizer. In *ICLR*, 2025. 3

[78] Wangbo Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. TrajectoryCrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *ICCV*, 2025. 3

[79] 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. *TPAMI*, 2025. 2, 3

[80] Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *NeurIPS*, 2019. 2

[81] Cheng Zhang, Boying Li, Meng Wei, Yan-Pei Cao, Camilo Cruz Gambardella, Dinh Phung, and Jianfei Cai. Unified camera positional encoding for controlled video generation. In *CVPR*, 2026. 2, 3, 6, 7, 8, 9, 17, 18, 19

[82] Jason Y. Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *ICLR*, 2024. 4

[83] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 4

[84] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM TOG*, 2018. 7, 19

[85] Yang Zhou, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Haoyu Guo, Zizun Li, Kaijing Ma, Xinyue Li, Yating Wang, Haoyi Zhu, Mingyu Liu, Dingning Liu, Jiange Yang, Zhoujie Fu, Junyi Chen, Chunhua Shen, Jiangmiao Pang, Kaipeng Zhang, and Tong He. OmniWorld: A multi-domain and multi-modal dataset for 4d world modeling. In *ICLR*, 2026. 19

[86] Zhenghong Zhou, Jie An, and Jiebo Luo. Latent-reframe: Enabling camera control for video diffusion model without training. In *ICCV*, 2025. 3

Appendix

A. Additional Derivations

A.1. Bilinearity of the Plücker Reciprocal Product

We restate the elementary fact used in Section 3.3. Let $\mathbf{r}_i = (\mathbf{d}_i, \mathbf{m}_i)$ and $\mathbf{r}_j = (\mathbf{d}_j, \mathbf{m}_j)$ be two 6-vectors. Define

$$\langle \mathbf{r}_i,\, \mathbf{r}_j \rangle_{\mathrm{Pl}} \;=\; \mathbf{d}_i \cdot \mathbf{m}_j + \mathbf{d}_j \cdot \mathbf{m}_i. \tag{18}$$

For any $\alpha, \beta \in \mathbb{R}$ and any $\mathbf{r}'_i$,

$$\begin{aligned}\langle \alpha\mathbf{r}_i + \beta\mathbf{r}'_i,\, \mathbf{r}_j \rangle_{\mathrm{Pl}} &= (\alpha\mathbf{d}_i + \beta\mathbf{d}'_i)\cdot\mathbf{m}_j + (\alpha\mathbf{m}_i + \beta\mathbf{m}'_i)\cdot\mathbf{d}_j \\ &= \alpha\big(\mathbf{d}_i\cdot\mathbf{m}_j + \mathbf{m}_i\cdot\mathbf{d}_j\big) \\ &\quad + \beta\big(\mathbf{d}'_i\cdot\mathbf{m}_j + \mathbf{m}'_i\cdot\mathbf{d}_j\big) \\ &= \alpha\,\langle \mathbf{r}_i,\, \mathbf{r}_j \rangle_{\mathrm{Pl}} + \beta\,\langle \mathbf{r}'_i,\, \mathbf{r}_j \rangle_{\mathrm{Pl}}.\end{aligned}$$

The same holds in the second argument by symmetry. Hence $\langle \cdot,\, \cdot \rangle_{\mathrm{Pl}}$ is bilinear.

A.2. Coplanarity Criterion

Two rays $\mathbf{r}_i, \mathbf{r}_j$ are coplanar (i.e. they intersect, are parallel, or coincide) if and only if $\langle \mathbf{r}_i,\, \mathbf{r}_j \rangle_{\mathrm{Pl}} = 0$. The proof is standard [16, 46] and rests on the observation that $\mathbf{d}_i\cdot\mathbf{m}_j + \mathbf{d}_j\cdot\mathbf{m}_i$ is, up to a sign, the determinant of a 4×4 matrix whose columns are the homogeneous coordinates of two points on each line; the determinant vanishes iff the four points are coplanar.

A.3. SE(3) Invariance

For a rigid motion $g = (R, \mathbf{t}) \in \mathrm{SE}(3)$, the action on a ray is

$$g\cdot(\mathbf{d}, \mathbf{m}) \;=\; (R\mathbf{d},\; R\mathbf{m} + \mathbf{t} \times R\mathbf{d}). \tag{19}$$

Substituting into Eq. (5),

$$\begin{aligned}\langle g\cdot\mathbf{r}_i,\, g\cdot\mathbf{r}_j \rangle_{\mathrm{Pl}} &= R\mathbf{d}_i\cdot(R\mathbf{m}_j + \mathbf{t}\times R\mathbf{d}_j) \\ &\quad + R\mathbf{d}_j\cdot(R\mathbf{m}_i + \mathbf{t}\times R\mathbf{d}_i) \\ &= \mathbf{d}_i\cdot\mathbf{m}_j + \mathbf{d}_j\cdot\mathbf{m}_i \\ &\quad + R\mathbf{d}_i\cdot(\mathbf{t}\times R\mathbf{d}_j) \\ &\quad + R\mathbf{d}_j\cdot(\mathbf{t}\times R\mathbf{d}_i).\end{aligned}$$

The last two terms cancel because the scalar triple product $a\cdot(b\times c)$ is antisymmetric in any two of its arguments, giving $R\mathbf{d}_i\cdot(\mathbf{t}\times R\mathbf{d}_j) = -R\mathbf{d}_j\cdot(\mathbf{t}\times R\mathbf{d}_i)$. Hence $\langle g\cdot\mathbf{r}_i,\, g\cdot\mathbf{r}_j \rangle_{\mathrm{Pl}} = \langle \mathbf{r}_i,\, \mathbf{r}_j \rangle_{\mathrm{Pl}}$.

A.4. Reduction of Term (D) to the Plücker Reciprocal Product

Recall (Eq. (11)) that, with the Q/K flip,

$$(\mathrm{D}) = \mathbf{r}_i^\top\, M\, \tilde{\mathbf{r}}_j, \qquad \mathbf{r}_i = (\mathbf{d}_i, \mathbf{m}_i), \quad \tilde{\mathbf{r}}_j = (\mathbf{m}_j, \mathbf{d}_j), \tag{20}$$

where $M = E_q^\top E_k \in \mathbb{R}^{6\times6}$. Writing $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ with $A, B, C, D \in \mathbb{R}^{3\times3}$,

$$(\mathrm{D}) = \mathbf{d}_i^\top A\mathbf{m}_j + \mathbf{d}_i^\top B\mathbf{d}_j + \mathbf{m}_i^\top C\mathbf{m}_j + \mathbf{m}_i^\top D\mathbf{d}_j. \tag{21}$$

Setting $A = D = I_3$, $B = C = \mathbf{0}_3$ collapses this to $\mathbf{d}_i\cdot\mathbf{m}_j + \mathbf{m}_i\cdot\mathbf{d}_j = \langle \mathbf{r}_i,\, \mathbf{r}_j \rangle_{\mathrm{Pl}}$. The choice $E_q = E_k = I_6$ in Proposition 1 produces exactly $M = I_6$, so the full M-block decomposition above shows *which* sub-blocks of M correspond to the reciprocal-product component and which sub-blocks generalize beyond it: the off-diagonal blocks A, D control the reciprocal-product-like part, while the diagonal blocks B, C control direction–direction and moment–moment couplings that are not present in the classical reciprocal product.

A.5. Flipped vs. Unflipped Parameterization Are Linearly Equivalent

Let $\mathbf{r} = (\mathbf{d}, \mathbf{m}) \in \mathbb{R}^6$ and let $P = \left(\begin{smallmatrix} 0 & I_3 \\ I_3 & 0 \end{smallmatrix}\right)$ be the orthogonal involution that swaps the $(\mathbf{d}, \mathbf{m})$ blocks ($P^2 = I_6$, $P^\top = P$). With learnable $E_q, E_k \in \mathbb{R}^{d\times6}$, the flipped parameterization in Eqs. (8)–(9) feeds $\mathbf{r}_i$ to E_q and $P\mathbf{r}_j$ to E_k, while the unflipped variant feeds $\mathbf{r}_j$ to E_k directly. The corresponding term (D) in Eq. (11) reads

$$(\mathrm{D})_{\text{flip}} = \mathbf{r}_i^\top E_q^\top E_k\, P\, \mathbf{r}_j, \qquad (\mathrm{D})_{\text{no-flip}} = \mathbf{r}_i^\top E_q^\top E_k\, \mathbf{r}_j.$$

For unconstrained E_k the map $E_k \mapsto E_kP$ is a bijection on $\mathbb{R}^{d\times6}$, so the two parameterizations realize the same set of bilinear forms in $(\mathbf{r}_i, \mathbf{r}_j)$. Under iid Gaussian initialization the columns of E_k and of E_kP also share the same distribution, so ℓ_2 weight decay is invariant under the substitution. The flip is therefore a choice of basis: it singles out a distinguished symmetric configuration ($E_q = E_k = I_6$) under which the bilinear form coincides with the Plücker reciprocal product (Proposition 1), but it does not change the function class realizable by the model.

A.6. Implementation Notes

Numerical safety. We clamp $\|\mathbf{m}\|$ at $\epsilon = 10^{-6}$ before normalization (Eq. (12)) to avoid division by zero in near-static-camera tokens.

Old-checkpoint compatibility. A previous version of the encoding (without PE RMSNorm) shipped α_q, α_k as zero-dimensional scalars, which are not compatible with FSDP's flat-parameter constraint that all sharded tensors have ≥ 1 dimension. Current checkpoints store these as shape-$(1,)$ tensors and the loader silently promotes old scalars at load time.

Two-layer MLP form of E_q, E_k. When the optional MLP form $E_q : \mathbb{R}^7 \to \mathbb{R}^h \to \mathbb{R}^d$ is used, zero-initializing

Method	Quality	Camera controllability				Distribution		
	CLIP↑	RotErr↓	TransErr↓	CamMC↓	ATE↓	FVD↓	FVD_c ↓	FID↓
Wan-2.2 5B Scale								
ReCamMaster [5]	24.97	0.131	0.297	0.402	0.322	874.30	890.52	62.53
ReRoPE [36]	25.20	0.137	0.241	0.366	0.307	684.57	650.31	60.77
UCPE [81]	25.39	0.113	0.180	0.274	0.203	703.41	755.83	61.50
SCoPE (ours)	**26.05**	**0.085**	**0.174**	**0.251**	**0.188**	**543.17**	**588.62**	**57.83**
Wan-2.2 14B Scale								
ReCamMaster [5]	25.31	0.109	0.221	0.335	0.286	675.23	697.10	59.21
ReRoPE [36]	25.85	0.114	0.185	0.290	0.243	493.30	525.61	49.18
UCPE [81]	25.72	0.082	0.163	0.238	0.180	529.42	558.70	54.75
SCoPE (ours)	**26.30**	**0.058**	**0.126**	**0.165**	**0.144**	**280.17**	**354.52**	**41.01**

Table 6. Main comparison on the RE10K held-out split under the *rescaled-alignment* protocol used in prior camera-controllability work. Predicted and ground-truth ViPE trajectories are normalized to a common translation norm before pose metrics (RotErr, TransErr, CamMC, ATE) are computed. Everything else matches Table 1. Higher is better for CLIP; lower is better for the rest.

both layers would produce a permanently dead gradient (the GELU after a zero linear is identically zero, and so is its Jacobian). We therefore zero only the output layer and apply Kaiming initialization to the input layer, which keeps the geometric branch exactly zero at step 0 while leaving its gradient non-zero.

A.7. Reference Results under Rescaled-Alignment

We briefly recap why the main paper reports pose metrics on the *raw, unscaled* ViPE trajectories. A camera-conditioned video model is meant to follow a user-specified trajectory *including its absolute scale*; once the predicted and the ground-truth trajectories are rescaled to a common norm at evaluation time, the resulting numbers no longer measure whether the model honored that scale, only whether the relative shape of the trajectory was reproduced. Two practical consequences follow. First, a degenerate solution that produces only a tiny global motion can still score well under the conventional protocol, because once both trajectories are stretched to the same norm the remaining shape error is small even though the absolute motion is essentially absent; the raw protocol exposes this failure mode directly. Second, rescaled pose numbers are implicitly expressed in units of each clip's own ground-truth norm and are therefore difficult to compare across datasets with different scale conventions (SfM-normalized, SLAM-internal, metric), whereas the raw protocol keeps the unit fixed to ViPE's metric output. We therefore adopt the raw protocol throughout the main paper, and report rescaled-alignment numbers here only as a cross-reference to prior camera-controllability work. Concretely, Table 6 repeats the comparison from Section 5.2 after normalizing the predicted and the ground-truth ViPE trajectories to a common translation norm before computing pose metrics. Everything else — held-out split, trajectory estimator, sampler configuration, and the video-quality metrics CLIP, FVD, FVD_c, and FID — is identical to Table 1; only the alignment step differs. Because rescaling absorbs absolute-scale errors, the four pose metrics are systematically smaller than their unscaled counterparts in the main table. SCoPE remains best on every metric in this setting as well, confirming that its advantage under the unscaled protocol is not an artifact of the alignment choice.

B. Additional Evaluation

B.1. Dynamic-Scene Video Quality

Purpose and setting. The camera-control metrics in the main paper do not directly test whether a model preserves the backbone's ability to generate dynamic content. A method could follow the camera trajectory while producing overly static subjects, or introduce motion at the cost of temporal coherence and visual quality. We therefore conduct an additional dynamic-scene evaluation on a comprehensive held-out set drawn from PanShot, OmniWorld, and Open-Sora-Plan. The evaluation videos do not overlap with the training set. All methods use the Wan2.2-14B backbone and receive the same conditioning inputs and target camera trajectories. We evaluate the generated videos with the official VBench [26] implementation under its default configuration and apply the same metrics to the corresponding real videos, reported as ground truth in Table 7.

We report six complementary VBench dimensions. Subject and Background Consistency measure temporal preservation of the foreground and scene, Motion Smoothness measures temporal regularity, and Dynamic Degree directly measures the amount of generated motion. Aesthetic Quality and Imaging Quality measure perceptual quality. Higher is better for every metric.

Results. As shown in Table 7, SCoPE ranks first among the generated methods on all six dimensions. Most no-

Method	Subject Consistency ↑	Background Consistency ↑	Motion Smoothness ↑	Dynamic Degree ↑	Aesthetic Quality ↑	Imaging Quality ↑
GT	0.935	0.942	0.984	0.863	0.558	0.692
ReCamMaster [5]	0.782	0.841	0.909	0.490	0.386	0.451
PRoPE [37]	0.855	0.903	0.955	0.693	0.451	0.606
UCPE [81]	0.897	0.928	0.979	0.752	0.508	0.622
SCoPE (ours)	**0.918**	**0.945**	**0.983**	**0.855**	**0.561**	**0.689**

Table 7. Dynamic-scene evaluation on the held-out PanShot, OmniWorld, and Open-Sora-Plan [40] mixture using Wan2.2-14B and the official VBench default configuration. "GT" denotes the real evaluation videos. All generated methods receive identical conditioning inputs and target camera trajectories. Higher is better for all metrics; bold marks the best generated result.

Method	RotErr ↓	TransErr ↓	CamMC ↓	ATE ↓	FVD ↓	FVD_c ↓
UCPE [81]	0.136/0.136	1.590/0.338	1.704/0.393	1.881/0.379	694.5	722.1
SCoPE (ours)	**0.089/0.089**	**1.164/0.220**	**1.091/0.279**	**1.217/0.208**	**529.0**	**550.8**

Table 8. Generalization beyond RealEstate10K on a held-out mixture of PanShot, OmniWorld, and Open-Sora-Plan using Wan2.2-5B. Pose cells report raw/rescaled-alignment results; FVD metrics are unaffected by trajectory alignment. Both methods use identical conditioning inputs, target trajectories, and evaluation settings. Lower is better for all metrics.

tably, its Dynamic Degree reaches 0.855, close to the real-video value of 0.863 and substantially above the strongest baseline value of 0.752. At the same time, SCoPE nearly matches the real videos in Motion Smoothness (0.983 versus 0.984), Aesthetic Quality (0.561 versus 0.558), and Imaging Quality (0.689 versus 0.692). Subject and Background Consistency also improve over all baselines. These results show that the gains in camera controllability do not come from suppressing scene motion: SCoPE preserves dynamic content while maintaining temporal coherence and perceptual quality.

B.2. Generalization beyond RealEstate10K

Purpose and setting. The main evaluation uses the standard RealEstate10K benchmark, whose videos are dominated by real-estate walkthroughs. To test whether the advantage of SCoPE transfers to broader scene and motion distributions, we evaluate on the same held-out PanShot, OmniWorld, and Open-Sora-Plan mixture used in Section B.1. This set has no overlap with the training videos and includes more diverse visual content, dynamic scenes, and camera trajectories than RealEstate10K. Both UCPE and SCoPE use the same Wan2.2-5B backbone and are evaluated from the same conditioning frames and target trajectories with the sampling and pose-estimation protocol of Section 5.1.

Following the main evaluation, we estimate the generated and ground-truth trajectories with ViPE and report RotErr, TransErr, CamMC, and ATE. Each pose cell in Table 8 gives the raw result followed by the rescaled-alignment result. We additionally report full-frame FVD and center-cropped FVD_c to measure distribution-level video quality.

Results. Table 8 shows that SCoPE remains better than UCPE on every camera-control and distribution metric. Under the raw protocol, SCoPE reduces RotErr, TransErr, CamMC, and ATE by 35%, 27%, 36%, and 35%, respectively. The advantage is preserved after conventional trajectory rescaling, showing that it does not arise only from absolute motion scale. SCoPE also lowers FVD from 694.5 to 529.0 and FVD_c from 722.1 to 550.8, both by approximately 24%. The consistent improvement on this broader held-out distribution indicates that the ray-coordinate encoding generalizes beyond the real-estate scenes used by the main benchmark.

Method	Camera integration	Step latency (s) ↓	Peak memory (GB) ↓
UCPE [81]	Compressed parallel branch	27.6	21.8
PRoPE [37]	Full parallel branch	34.6	22.3
SCoPE (ours)	Additive Q/K injection	**24.4**	**11.6**

Table 9. Inference efficiency on Wan2.2-TI2V-5B. Step latency is averaged over the 50 denoising steps of a complete inference run. All methods use the same input and inference configuration. Lower is better for both metrics.

B.3. Runtime and Memory

Purpose and setting. We compare the inference cost introduced by different camera- conditioning mechanisms on the Wan2.2-TI2V-5B backbone. All methods generate videos at 480×832 resolution with 81 frames under the same inference configuration. Step latency is the mean wall-clock time of one denoising step, averaged over the 50 denoising steps of a complete inference run. Peak memory is measured during the same run.

Results. SCoPE projects the ray features directly into the existing queries and keys, requiring neither a parallel branch nor an additional attention map. Its added per-token com-

putation is $\mathcal{O}(Nd)$, which is small relative to the $\mathcal{O}(N^2d)$ self-attention computation. UCPE instead uses a parallel geometry branch with an attention-compression ratio of eight. For PRoPE, direct substitution of the pretrained positional transformation did not yield stable fine-tuning under the matched budget; we therefore use a zero-initialized full parallel branch for the trainable video-DiT configuration reported here.

As shown in Table 9, SCoPE has the lowest inference cost. It reduces step latency by 12% relative to UCPE and 30% relative to PRoPE. Its peak memory is 11.6 GB, compared with 21.8 GB for UCPE and 22.3 GB for PRoPE, reducing memory consumption by approximately 47% and 48%, respectively. These results show that the direct Q/K injection retains the efficiency of the pretrained attention path while avoiding the cost of a parallel camera-conditioning branch.

C. Cross-Domain Training Strategy

This section describes the data sources we use, the steps we take to make their camera-translation scales comparable, and the optimization recipe we apply on top of the pretrained backbone.

C.1. Data Sources

We train on a concatenation of four datasets that span quite different camera-motion distributions and pose-estimation pipelines:

- **RealEstate10K** (RE10K) [84]: about 10K real-estate walkthrough clips with COLMAP-style SfM poses [50]. Smooth, primarily forward camera motion through indoor scenes; the canonical benchmark for camera-controllable video synthesis.
- **DL3DV** [41]: a more diverse multi-scene dataset with both indoor and outdoor scenes and a wider range of camera motions; poses provided by the dataset.
- **PanShot** [81] (in-house): an internal dataset of synthesized panoramic shots, where camera trajectories are known by construction (we use them as ground-truth) and the field of view is known per clip.
- **OmniWorld** [85]: open-world video with relatively dynamic content, with camera extrinsics estimated by DROID-SLAM [57]. We use the released text captions as conditioning prompts. DROID-SLAM applies an internal scale normalization that is unrelated to either metric scale or the SfM normalization of RE10K/DL3DV.

We do not perform any further re-estimation of camera pose; SCoPE treats the per-frame extrinsics and intrinsics as inputs and is agnostic to which estimator produced them. The four data sources span SfM poses, deep SLAM poses, and ground-truth-by-construction poses, which provides a stress test for whether the encoding generalizes across pose distributions.

C.2. Per-Dataset Trajectory-Scale Alignment

The four data sources differ substantially in absolute translation magnitude. To make their Plücker moments comparable in distribution, we apply a single per-dataset multiplicative factor $\eta \in \mathbb{R}_{>0}$ to the camera positions:

$$\tilde{\mathbf{o}}_t \;=\; \eta \cdot \mathbf{o}_t, \qquad \tilde{\mathbf{m}}_i = \tilde{\mathbf{o}}_t \times \mathbf{d}_i = \eta \cdot \mathbf{m}_i. \tag{22}$$

The values of η for each dataset are chosen so that the median moment magnitude $\mathrm{median}(\|\mathbf{m}\|)$ over a sample of clips falls in a comparable range across datasets; in our default configuration $\eta_{\mathrm{RE10K}} = \eta_{\mathrm{DL3DV}} = \eta_{\mathrm{PanShot}} = 1.0$ and $\eta_{\mathrm{OmniWorld}} = 20.0$, the latter chosen to bring DROID-SLAM's internally normalized translations into approximately the same band as RE10K. Concrete moment-magnitude statistics are reported in Section 5.

C.3. Optional Per-Clip Near-Depth Normalization

A constant per-dataset η does not absorb *within*-dataset variation: a long-corridor RE10K clip and a tight-room RE10K clip can still differ by an order of magnitude in scene depth. When a per-clip near-depth estimate z_n is available, we additionally divide the clip's translation by z_n:

$$\tilde{\mathbf{o}}_t \;=\; \frac{\eta}{z_n} \cdot \mathbf{o}_t. \tag{23}$$

The near-depth estimate z_n is obtained offline (from a depth estimator on a small set of frames) and serves as a per-clip proxy for the scene scale. When z_n is unavailable for a clip we fall back to η alone. For OmniWorld we use only η in this work.

These two normalizations (per-dataset η and per-clip $1/z_n$) together aim to make the moment-magnitude distribution roughly homogeneous across the training mixture. The Normalize-Gate-Inject mechanism (Section 4.2) provides robustness for the residual within-distribution variation that this preprocessing does not remove.

C.4. Optimization Recipe

Backbone and conditioning. We start from a pretrained 5B-parameter Wan2.2-TI2V-5B backbone [61] and use it in its image-to-video mode (the first frame of the training clip is conditioned via the standard first-frame VAE-fuse path of the backbone). All experiments are trained at 480×832 spatial resolution and $T = 81$ raw frames (corresponding to $T_z = 21$ latent frames).

Zero initialization. The new Q/K geometry branch is zero-initialized, so $\mathrm{pe}^q = \mathrm{pe}^k = 0$ at step 0 and the network matches the pretrained DiT. PE RMSNorm weights start at 1, scale-gate biases at 0 (so $g \approx 0.5$), and α starts at 0. When E_q, E_k are implemented as two-layer MLPs, we zero only the output layer and use Kaiming initialization for the input layer.

Trainable parameters and optimizer. We jointly train the new SCoPE modules (E_q, E_k, PE RMSNorms, scale gates, α) together with the backbone self-attention, FFN, and modulation-norm layers; text cross-attention, the VAE, and the text encoder remain frozen. The newly added SCoPE parameters account for less than 0.1% of the model. We use AdamW with a single learning rate of $2{\times}10^{-5}$ across all trainable groups, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 10^{-2}, and a cosine schedule that anneals to 10% of the starting value at $T_{\max}$ steps. We tried per-group asymmetric learning rates that protect the pretrained weights more aggressively but observed no consistent improvement, so we kept the simpler single-LR recipe.

Mixed precision and parallelism. We train in BF16 with FP32 master weights, gradient checkpointing on the DiT blocks, and DDP across GPUs. Gradient clipping by norm at 1.0 is applied at the DDP root level.

Validation-time generation. Every $5{,}000$ steps we run inference on a small held-out validation set (a 100-clip RE10K-test split) and save (i) the per-step training and validation loss, (ii) the synthesized videos, and (iii) ground-truth videos for visual comparison. Validation videos are generated with the standard 50-step CFG sampler at $\mathrm{cfg\_scale} = 5.0$.