SCoPE: Sightline-Coordinate Positional Encoding for Video Diffusion Transformers

Anonymous authors
Paper under double-blind review

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 raw-ray form recovers the reciprocal product, a classical coplanarity test for lines of sight. 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 parameters. On Wan2.2 at 5B and 14B under matched data and budget, SCoPE improves every 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.

Anonymous project page: https://scope-review.github.io/

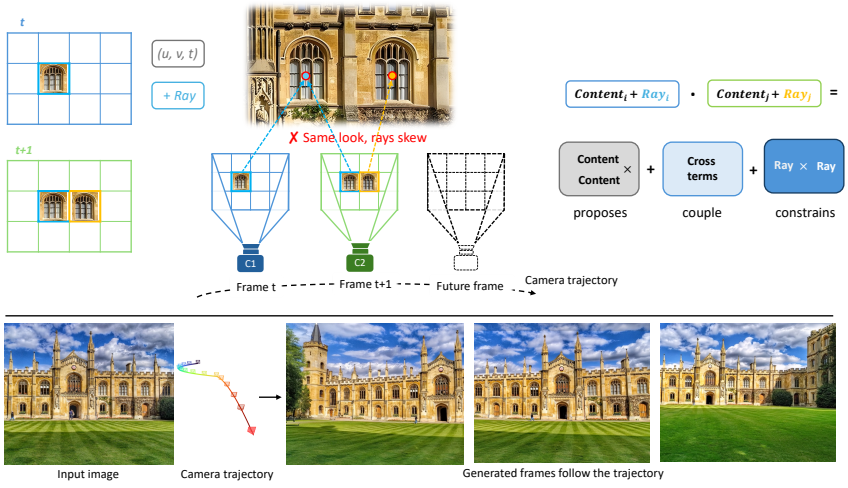

Figure 1: **Camera control as a property of the coordinate system.** SCoPE gives every token a second address, its camera ray (left). Rays are fixed by the user's trajectory while the scene, dashed, is not yet generated (center). Appearance cannot choose between the two identical windows in frame $t{+}1$. Only one of their rays meets the ray from frame t. Added to the pretrained queries and keys, the rays give attention a direct way to compare sightlines (right). Content proposes matches, ray pairing scores geometric compatibility, and cross terms couple appearance with viewing geometry.

1 Introduction

A transformer knows only as much about space as its positional encoding supplies. In video diffusion transformers (DiTs) (Ho et al., 2022; Yang et al., 2025; Kong et al., 2024a; Wan et al., 2025), that encoding is typically a factorized RoPE (Su et al., 2024; Heo et al., 2024) over the (u, v, t)

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

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, where its line of sight meets a surface not yet generated. World-point methods must first reconstruct content (Bai et al., 2026; Xu et al., 2026; Wang & He, 2026; Yu et al., 2025b), 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 (He et al., 2025a; Bai et al., 2025a; Burgert et al., 2025), every future token's ray is fixed before a single pixel is synthesized. A ray fixes the line of sight but not its depth. 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 (Su et al., 2024), whose rotations collapse two absolute indices into their offset, the grid's relative quantity. For camera rays, work from multi-view synthesis (Miyato et al., 2024; Kong et al., 2024b; Li et al., 2025a) to video diffusion (Zhang et al., 2026; Li et al., 2026a; Xiang et al., 2026) applies relative camera transforms *multiplicatively* to queries and keys. We build on this line, but ask: *What must the geometric score measure?* Deciding whether two tokens could see the same point imposes three requirements: (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) (Miyato et al., 2024; Kong et al., 2024b; Li et al., 2025a; 2026a). Direction-only encodings are blind to translation (ii) (Xiang et al., 2026). Multiplicative application routes geometry entirely through content, leaving the score with no term that reads the two rays by themselves (iii) (Zhang et al., 2026). 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 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 yields four terms: content with content, the two content–ray cross terms, and ray with ray. The last term reads only the two rays, as demand (iii) requires. For camera rays, we focus on incidence: whether their supporting lines meet. The classical *reciprocal product* is bilinear in the two rays and vanishes when their supporting lines are coplanar, including when the rays intersect. This motivates a geometry-only pairing in attention. For raw Plücker inputs, learned linear projections yield a bilinear form that recovers the reciprocal product in a canonical configuration. The content term identifies candidate matches from visual evidence, while the ray pairing scores geometric compatibility (Figure 1). The ray coordinate makes each token 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 also carries the translation scale, set by the pose source. Poses from metric capture come in meters, while monocular SfM (Schönberger & Frahm, 2016) and SLAM (Durrant-Whyte & Bailey, 2006; Teed & Deng, 2021) 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, at no cost 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 (Zhang & Sennrich, 2019) 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 use under 0.1% of the parameters, leave RoPE bit-exact, and add no second attention or adapter branch. Zero-initialized injection starts training from the exact pretrained DiT.

If camera control really is a property of the coordinate system, three predictions 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 these by retrofitting SCoPE onto Wan2.2 (Wan et al., 2025) at 5B and 14B. All methods share the backbone, data, and training budget, differing only in camera conditioning. 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 is 29% and FVD 43% below the strongest baseline on each metric. Per-term ablations put the largest cost on removing the ray pairing, which the multiplicative form lacks. Without it, rotation error nearly doubles. From 5B to 14B, the margins widen: 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 geometry-only pairing that recovers the reciprocal product in a canonical raw-ray configuration.
- **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. Camera adapters encode extrinsics as frame tokens, pixel-aligned Plücker maps, or cross-attention and LoRA conditions (Wang et al., 2024; He et al., 2025a; Xu et al., 2024; Bahmani et al., 2025a;b). Other methods re-render a reconstructed proxy along the target trajectory (Yu et al., 2025b;a), steer frozen generators through warped views (Hou & Chen, 2025), or concatenate view tokens for native attention to find correspondences (Bai et al., 2025a; Xu et al., 2025). Noise- and latent-based camera control is orthogonal and combinable with our encoding (Burgert et al., 2025; Zhou et al., 2025). These routes introduce modules, sequence length, or reconstruction; SCoPE uses rays fixed by the commanded trajectory before generation.

Camera geometry inside attention. Epipolar attention constrains token pairs with a geometric mask (He et al., 2020; Du et al., 2023). GTA applies relative SE(3) transforms multiplicatively to Q/K/V in a transformer trained from scratch (Miyato et al., 2024); EscherNet and RePAST use related camera operators (Kong et al., 2024b; Safin et al., 2023). PRoPE extends them to full camera frustums while retaining patch RoPE on separate channels (Li et al., 2025a). For video diffusion, UCPE uses a zero-initialized parallel bypass with roughly 0.5% added parameters and an extra attention per block (Zhang et al., 2026); ReRoPE replaces low-frequency temporal RoPE with relative projective camera transforms (Li et al., 2026a), and ViewRope rotates Q/K subvectors by viewing-ray direction (Xiang et al., 2026). These multiplicative encodings route geometry through content, without a score term depending on the two rays alone. SCoPE instead adds a ray coordinate in the pretrained attention, preserving RoPE and producing mixed content–ray and ray–ray terms without a second branch or attention. ReRoPE normalizes away global translation scale; SCoPE retains log-magnitude and learns its gate. Extended camera-encoding comparisons are provided in Appendix F.

Positional encodings and ray representations. Transformer-XL and T5 add relative attention terms; RoPE encodes relative offsets through rotations (Dai et al., 2019; Raffel et al., 2020; Su et al., 2024; Heo et al., 2024). SCoPE extends the additive family to ray incidence. Ray-space rendering, scene representations, and equivariant transformers (Levoy & Hanrahan, 1996; Sitzmann et al., 2021; Xu et al., 2023) treat rays as represented or predicted quantities; SCoPE uses them as positional coordinates compared by pretrained attention. Its vanishing residual follows the initialization principle of ControlNet (Zhang et al., 2023). Appendix F extends this literature review.

3 BACKGROUND AND NOTATION

3.1 TOKENS AND CAMERA RAYS

In latent-video diffusion / flow matching (Ho et al., 2020; Lipman et al., 2023; Blattmann et al., 2023; Yang et al., 2025; Wan et al., 2025), patchified VAE latents give $X \in \mathbb{R}^{N\times d}$, $N = T_z H_p W_p$, with T_z latent frames and an $H_p \times W_p$ spatial grid. Token i has feature $x_i \in \mathbb{R}^d$ and index (u_i, v_i, t_i). Pretrained projections are $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ and 3D RoPE is $R(u, v, t)$ (Su et al., 2024; Heo et al., 2024); attention uses QKNorm (Henry et al., 2020). Grid indices do not specify a 3D origin. SCoPE preserves RoPE and adds camera rays to Q/K; full backbone notation is in Appendix E.

Each frame has camera-to-world extrinsics $T_t \in \mathrm{SE}(3)$ and intrinsics $K_t \in \mathbb{R}^{3\times3}$, using OpenCV axes ($x$ right, y down, z forward). For a patch center (p_u, p_v), let R_t and $\mathbf{o}_i$ be the rotation and camera position from T_t. The unit camera/world directions and Plücker moment are

$$\mathbf{d}_i^{\mathrm{cam}} = \frac{K_t^{-1}[p_u, p_v, 1]^\top}{\|K_t^{-1}[p_u, p_v, 1]^\top\|}, \qquad \mathbf{d}_i = R_t \mathbf{d}_i^{\mathrm{cam}}, \qquad \mathbf{m}_i = \mathbf{o}_i \times \mathbf{d}_i. \tag{1}$$

3.2 RAY GEOMETRY AND SCALE

A ray is represented by $\mathbf{r} = (\mathbf{d}, \mathbf{m}) \in \mathbb{R}^6$; $\mathbf{o}$ may be any point on its supporting line. The constraints $\|\mathbf{d}\| > 0$ and $\mathbf{d} \cdot \mathbf{m} = 0$ parameterize an oriented line up to scale, with $\|\mathbf{d}_i\| = 1$ in our construction (Hartley & Zisserman, 2004; Pottmann & Wallner, 2001). For two rays, the Plücker reciprocal (inner) product is

$$\langle \mathbf{r}_i, \mathbf{r}_j \rangle_{\mathrm{Pl}} \equiv \mathbf{d}_i \cdot \mathbf{m}_j + \mathbf{d}_j \cdot \mathbf{m}_i = \mathbf{r}_i^\top J \mathbf{r}_j, \quad J = \begin{pmatrix} \mathbf{0}_3 & I_3 \\ I_3 & \mathbf{0}_3 \end{pmatrix}. \tag{2}$$

It is bilinear in the two rays, matching the query–key dot product. It vanishes exactly when the supporting lines are coplanar, including intersecting, parallel, or coincident lines. It is invariant to a joint rigid-frame change: $\langle g \cdot \mathbf{r}_i, g \cdot \mathbf{r}_j \rangle_{\mathrm{Pl}} = \langle \mathbf{r}_i, \mathbf{r}_j \rangle_{\mathrm{Pl}}$ for $g \in \mathrm{SE}(3)$. Proofs are in Appendices J.1–J.3.

Under a global positive translation rescaling $\mathbf{o} \to s\mathbf{o}$, unit direction is unchanged and $\mathbf{m} \to s\mathbf{m}$, so $\langle \mathbf{r}_i, \mathbf{r}_j \rangle_{\mathrm{Pl}} \to s \cdot \langle \mathbf{r}_i, \mathbf{r}_j \rangle_{\mathrm{Pl}}$. Raw geometry magnitudes therefore vary across pose-scale conventions; NGI addresses this mismatch in Section 4.2.

4 METHOD

Figure 2 summarizes SCoPE. Plücker Flip PE adds camera-ray features to Q/K while preserving pretrained RoPE, introducing ray–ray and mixed feature–ray attention terms (Section 4.1). NGI controls their scale and magnitude across heterogeneous pose sources (Section 4.2). Implementation and training pseudocode are in Appendices J.6 and H.

4.1 PLÜCKER FLIP POSITIONAL ENCODING

The Plücker reciprocal product in Eq. (2) is bilinear in the two rays, matching the algebraic form of the query–key dot product. This motivates an additive construction whose geometry-only term can recover that pairing in a canonical configuration.

Each token i has a camera ray with Plücker coordinate $\mathbf{r}_i = (\mathbf{d}_i, \mathbf{m}_i) \in \mathbb{R}^6$ (Section 3.1). We use two per-layer projections $E_q, E_k \in \mathbb{R}^{d\times6}$ and add the projected ray features after the usual content projection and RoPE:

$$q_i = R(u_i, v_i, t_i)\, W_Q x_i \;+\; E_q\,(\mathbf{d}_i,\, \mathbf{m}_i), \qquad k_j = R(u_j, v_j, t_j)\, W_K x_j \;+\; E_k\,(\mathbf{m}_j,\, \mathbf{d}_j). \tag{3}$$

Here $\tilde{W}_Q x_i = R(u_i, v_i, t_i) W_Q x_i$ and $\tilde{W}_K x_j = R(u_j, v_j, t_j) W_K x_j$ denote RoPE-encoded content. The additive geometry preserves pretrained (u, v, t) RoPE; Q receives $(\mathbf{d}, \mathbf{m})$ and K receives $(\mathbf{m}, \mathbf{d})$. For this algebraic exposition we omit content QKNorm and use raw 6D rays; the full content path and normalized 7D inputs follow in Section 4.2.

Attention score decomposition. Expanding $\langle q_i, k_j \rangle$ using Eq. (3) yields four terms:

$$\langle q_i, k_j \rangle = \underbrace{\langle \tilde{W}_Q x_i,\, \tilde{W}_K x_j \rangle}_{\text{(A) content}} + \underbrace{\langle \tilde{W}_Q x_i,\, E_k \tilde{\mathbf{r}}_j \rangle}_{\text{(B) content}\to\text{geom}} + \underbrace{\langle E_q \mathbf{r}_i,\, \tilde{W}_K x_j \rangle}_{\text{(C) geom}\to\text{content}} + \underbrace{\langle E_q \mathbf{r}_i,\, E_k \tilde{\mathbf{r}}_j \rangle}_{\text{(D) geometry}}, \tag{4}$$

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 (Pottmann & Wallner, 2001); 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 5.4.1). Term (D) is the geometry-only interaction:

$$\text{(D)} = \mathbf{r}_i^\top M\, \tilde{\mathbf{r}}_j, \qquad M = E_q^\top E_k \in \mathbb{R}^{6\times 6}. \tag{5}$$

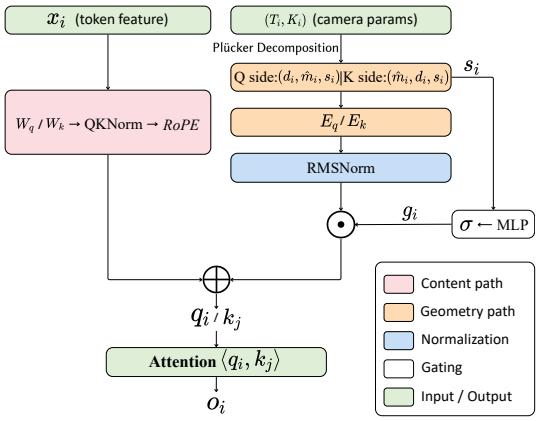

Figure 2: **SCoPE in self-attention.** The content path (pink) preserves pretrained $W_q/W_k \to \mathrm{QKNorm} \to \mathrm{RoPE}$. The geometry path (orange/blue) projects normalized Plücker features through E_q/E_k, applies RMSNorm, and gates them by a learned function of log-magnitude.

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*

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

Proof: Appendix J.4. For unconstrained E_q, E_k, flipped and unflipped forms share a hypothesis class: E_k in one corresponds to $E_k P$ in the other, where P swaps the $(\mathbf{d}, \mathbf{m})$ blocks (Appendix J.5). The flip chooses a basis without enlarging the function space. The condition $E_q^\top E_k = I_6$ gives the textbook reciprocal product, and $\|M - I_6\|$ quantifies departure from this canonical form in the raw linear setting. The empirical flip/no-flip gap is small (Section 5.4.1).

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)

NGI controls two mismatches in pretrained attention. First, the moment $\mathbf{m} = \mathbf{o} \times \mathbf{d}$ depends on translation magnitude: SfM-normalized scenes (Schönberger & Frahm, 2016), DROID-SLAM (Durrant-Whyte & Bailey, 2006; Teed & Deng, 2021), metric datasets, and different camera baselines produce heterogeneous scales. Raw $E\mathbf{r}$ mixes unit direction with this scale-dependent moment. Second, unnormalized geometry does not match the QKNorm-controlled content magnitude; one global scalar cannot handle per-clip pose scale and local geometry/content balance.

Normalize. We separate moment direction and log-magnitude:

$$\hat{\mathbf{m}} = \mathbf{m} / \max(\|\mathbf{m}\|, \epsilon), \qquad s = \log \max(\|\mathbf{m}\|, \epsilon), \tag{6}$$

where ϵ is a small numerical constant. The 7D inputs are $f_i^{(q)} = (\mathbf{d}_i, \hat{\mathbf{m}}_i, s_i)$ and, after the Q/K flip, $f_j^{(k)} = (\hat{\mathbf{m}}_j, \mathbf{d}_j, s_j)$. 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.

Gate and inject. A two-layer MLP G_s maps per-token log-magnitude to a per-channel sigmoid gate, with biases initialized so that $g \approx 0.5$ at $s = 0$. We project the 7D inputs, normalize them using per-layer learnable RMSNorms N_q, N_k, and gate their contribution to attention:

$$g_i = \sigma\big(G_s(s_i)\big) \in (0,1)^d, \tag{7}$$

$$\mathrm{pe}_i^q = g_i \odot N_q\big(E_q f_i^{(q)}\big), \qquad \mathrm{pe}_j^k = g_j \odot N_k\big(E_k f_j^{(k)}\big), \tag{8}$$

$$q_i = R_i\,\mathrm{QKNorm}(W_Q x_i) + \alpha \cdot \mathrm{pe}_i^q, \qquad k_j = R_j\,\mathrm{QKNorm}(W_K x_j) + \alpha \cdot \mathrm{pe}_j^k. \tag{9}$$

Here $R_i = R(u_i, v_i, t_i)$ applies RoPE after QKNorm, as in Figure 2. The gate adapts injection to moment magnitude, and α controls geometry/content balance; normalizing both branches gives it consistent meaning across layers and training examples. NGI retains the four-term score decomposition, with a learned geometry-only interaction $\alpha^2\langle \mathrm{pe}_i^q, \mathrm{pe}_j^k\rangle$; Proposition 1 characterizes the canonical raw-ray case.

Log-scale augmentation. During training, the gate optionally receives a perturbed log-magnitude:

$$\tilde{s}_i = s_i + \delta, \quad \delta \sim \mathrm{Uniform}(-1.2,\, 1.6), \text{ with prob. } 0.3, \tag{10}$$

where all tokens in a clip share the offset. This simulates a global rescaling $\mathbf{o} \to e^{\delta}\mathbf{o}$ without changing supervision. Only the gate receives $\tilde{s}_i$; the projection $E_q f^{(q)}$ uses the true s_i.

Each attention layer independently applies per-token Plücker computation, decomposition $(\mathbf{d}, \hat{\mathbf{m}}, s)$, the scale gate, and Q/K flip. SCoPE preserves pretrained 3D RoPE and adds geometric features to Q/K; comparisons with alternative camera-aware encodings (Li et al., 2025a; Zhang et al., 2026; Li et al., 2026a) are in Section 5 and Appendix B. See Appendix H, Algorithm 1, for the full procedure.

5 Experiments

We evaluate generation quality, camera-trajectory controllability, and cross-frame geometric consistency. Camera-conditioned baselines and four alternative camera-aware PE designs share our backbone, isolating the proposed Plücker-flip/NGI formulation.

5.1 Setup

Backbone. Main comparisons use the official Wan-2.2-TI2V-5B and I2V-14B variants (Wan et al., 2025); all ablations use TI2V-5B for efficiency. All runs generate 480×832 videos with $T = 81$ frames ($T_z = 21$ latent frames) and use the four-dataset training mixture (Appendix G). Variants share their respective pretrained weights and differ only in the integrated camera module.

Baselines. We compare CameraCtrl (He et al., 2025a) (Plücker maps via ControlNet), ReCamMaster (Bai et al., 2025a) (camera-controlled video-to-video re-rendering), ProPE (Li et al., 2025a) (a zero-initialized parallel branch), UCPE (Zhang et al., 2026) (full camera geometry via a spatial-attention adapter), and ReRoPE (Li et al., 2026a) (relative camera pose replacing low-frequency temporal RoPE). Without official checkpoints, we re-implement on our backbone with matched data and budget (Appendix J). The four FreqSplit-RoPE variants are compared in Appendix B.

Evaluation. Held-out RE10K (Zhou et al., 2018) uses the first GT frame and GT trajectory for image-to-video generation. We report eight metrics: **CLIP**↑ (inter-frame similarity); **RotErr**↓ (geodesic rotation), **TransErr**↓ (L2 translation), **CamMC**↓ (per-frame pose Frobenius norm), and **ATE**↓ (RMS absolute trajectory error); **FVD**↓/**FVD$_c$**↓ (full-frame/center-cropped Fréchet Video Distance (Unterthiner et al., 2019)) and **FID**↓ (per-frame Fréchet Inception Distance). ViPE (Huang et al., 2025) estimates both GT and generated poses, anchored to frame 0 and compared *without rescaling*. This preserves absolute scale and avoids rewarding artificially small motions. Conventional maximum-translation-normalized results are in Appendix A.1, with unchanged ranking. All methods use 50 denoising steps and CFG 5.0 (text)/1.0 (camera), as in Appendix G.4.

5.2 Main Comparison

Table 1 shows that SCoPE improves all four pose-error metrics as well as CLIP, FVD, FVD$_c$, and FID. On RE10K, Figure 3 compares the same clip under identical camera conditioning: SCoPE follows the target trajectory more faithfully than ReCamMaster and UCPE, which drift. Appendix

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.1.

	Quality	Camera controllability				Distribution		
Method	CLIP↑	RotErr↓	TransErr↓	CamMC↓	ATE↓	FVD↓	FVD_c ↓	FID↓
Wan-2.2 5B Scale								
CameraCtrl	25.04	0.152	1.292	1.355	1.501	824.37	805.62	64.08
ProPE	24.90	0.117	1.653	1.694	1.915	717.2	611.5	58.33
ReCamMaster	24.97	0.131	1.226	1.279	1.460	874.30	890.52	62.53
ReRoPE	25.20	0.137	1.109	1.215	1.350	684.57	650.31	60.77
UCPE	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	25.31	0.109	0.976	1.012	0.995	675.23	697.10	59.21
ReRoPE	25.85	0.114	0.820	0.905	0.859	493.30	525.61	49.18
UCPE	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**

CameraCtrl (He et al., 2025a); ProPE (Li et al., 2025a); ReCamMaster (Bai et al., 2025a); ReRoPE (Li et al., 2026a); UCPE (Zhang et al., 2026).



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 (Bai et al., 2025a), UCPE (Zhang et al., 2026), and SCoPE (ours). SCoPE follows the target trajectory more faithfully across frames; ReCamMaster and UCPE drift from the camera path.

Figure A2 extends the in-domain evaluation to indoor walkthroughs, outdoor pans, and forward-moving corridors under distinct trajectories.

Table 2: Dynamic-scene evaluation on the held-out PanShot, OmniWorld, and Open-Sora-Plan (Lin et al., 2024) 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	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	0.782	0.841	0.909	0.490	0.386	0.451
PRoPE	0.855	0.903	0.955	0.693	0.451	0.606
UCPE	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**

ReCamMaster (Bai et al., 2025a); PRoPE (Li et al., 2025a); UCPE (Zhang et al., 2026).

For out-of-distribution inputs absent from training, Figure 4 shows that SCoPE preserves hand-painted style while executing the prescribed camera path. Appendix Figure A1 compares methods on artistic paintings, where SCoPE produces the intended motion and parallax while baselines fail to follow the trajectory; cinematic multi-trajectory results are given in Appendix C.3. Table 2 evaluates dynamic-scene quality with VBench (Huang et al., 2024): SCoPE leads the generated methods on all six metrics. Evaluation details and inference efficiency are in Appendices A.2 and A.4, respectively.

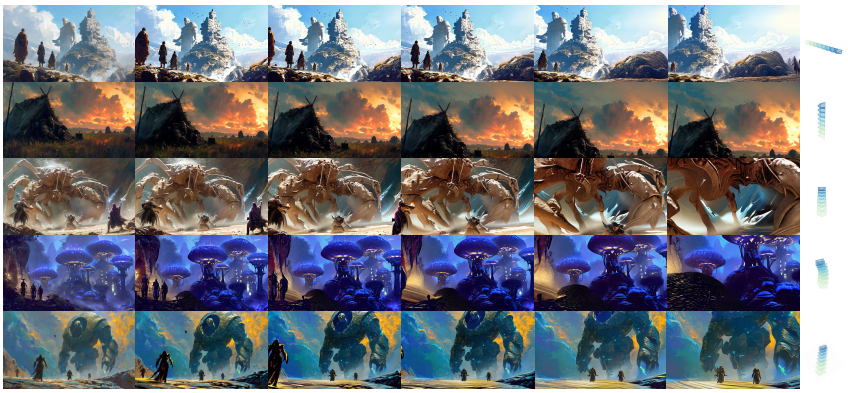

Figure 4: Generalization to stylized inputs. Each row is a different hand-painted concept-art first frame driven by a distinct target camera trajectory with SCoPE; frames go left to right along time, and the rightmost column shows the input trajectory as a stack of camera frusta.

5.3 Closed-Loop Revisit Consistency

From each conditioning image, we command four camera loops—truck, orbit, dolly, and non-palindromic—that leave and return to the starting pose. We evaluate frames 36–40 near maximum excursion (away window) and 76–80 after completion (return window). ViPE (Huang et al., 2025) estimates poses relative to frame 0.

Return Translation Ratio divides mean return-window displacement by the maximum over the generated trajectory; **Return Rotation** is the mean return-window geodesic rotation from frame 0. **Normalized ATE** compares frame-0-anchored commanded and predicted trajectories, each normalized by its own maximum translation magnitude. **Return LPIPS** (AlexNet) and **DINOv2 Similarity** (global-feature cosine) compare return frames with the conditioning image; **LPIPS Recovery** $= 1 - \mathrm{LPIPS}_{\mathrm{return}}/\mathrm{LPIPS}_{\mathrm{away}}$ is the fraction of away-window appearance discrepancy recovered. Scores average over each five-frame window, then the four paths per image, then cases. Pose and appearance are interpreted jointly: similarity alone cannot verify commanded motion.



Figure 5: **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.

Table 3: 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.

	Camera return			Appearance recovery		
Method	Return Trans. Ratio ↓	Return Rot. (°) ↓	Norm. ATE ↓	Return LPIPS ↓	LPIPS Recovery ↑	DINOv2 Sim. ↑
ProPE	0.784	4.90	0.533	0.603	−0.176	0.874
ReCamMaster	0.859	3.80	0.625	0.653	0.094	0.890
UCPE	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**

ProPE (Li et al., 2025a); ReCamMaster (Bai et al., 2025a); UCPE (Zhang et al., 2026).

Leading all six revisit metrics (Table 3), SCoPE reduces Return Translation Ratio by 79% and Return LPIPS by 43% versus the strongest baseline per metric. Joint pose-return and appearance gains indicate recovery after the commanded excursion, rather than apparent consistency from suppressed motion. Figure 5 shows large return errors for ProPE and ReCamMaster, partial recovery for UCPE, and the closest pose and content recovery for SCoPE.

5.4 Ablations

5.4.1 Component Ablation

Table 4 isolates each component of SCoPE, with all other settings and training recipes unchanged. Alternative camera-aware PE designs are compared in Appendix B, Table A4. **What each row tests.** Table 4 varies the Q/K flip, the NGI pipeline, PE RMSNorm, log-scale augmentation, initialization, and whether geometry is also injected into v. Appendix I provides the configuration and discussion for these six variants. The w/o Normalize-Gate-Inject row removes decomposition, scale gating, and PE RMSNorm together, while the flip comparison tests a reparametrization under unconstrained E_q, E_k (Appendix J.5). The last two rows directly intervene on the attention decomposition of Eq. (4): **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. The drops in Table 4 show that both the (B)+(C) cross terms and the (D) geometry term are needed for full performance; Appendix I.1 provides the full analysis.

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.

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 NGI	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)	25.21	0.135	1.174	1.358	695.41
w/o (D)	24.08	0.159	1.382	1.440	733.08

5.4.2 Data composition

Table 5 adds DL3DV, PanShot, and OmniWorld sequentially to RE10K alone. The full mixture achieves the best CLIP, CamMC, ATE, and FVD, while the three-dataset mixture has the lowest RotErr. Gains are not monotonic across all metrics, and this comparison does not isolate the effect of scale handling.

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

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**

6 Conclusion

We presented SCoPE, a sightline-coordinate positional encoding for video diffusion transformers built on the algebraic match between the Plücker reciprocal product and the attention dot product. Adding per-token ray features to queries and keys introduces a geometry-only interaction alongside content–geometry cross terms. Its canonical raw-ray form recovers the Klein form, connecting the attention score to classical line geometry. Normalize-Gate-Inject decouples ray direction from translation magnitude and adaptively gates the encoding for stability across heterogeneous pose scales. The module adds less than 0.1% parameters to a pretrained 5B video DiT, preserves RoPE without an adapter or architectural change. Quantitative and qualitative results show improved camera control, cross-view and closed-loop consistency, with pretrained generation quality preserved.

AI USE STATEMENT

We used a generative AI assistant to support manuscript writing, including polishing language, organizing the LaTeX source files, and identifying typographical errors. AI was not used to develop research ideas, design experiments, or derive or prove mathematical claims. The authors reviewed all AI-assisted changes and take responsibility for the final content of this work.

ETHICS STATEMENT

SCoPE generates camera-controlled videos and could be misused to produce misleading synthetic media. Camera control and geometric consistency do not establish the truth of generated content. Applications should account for consent, privacy, and the licenses of the data and pretrained models used.

REPRODUCIBILITY STATEMENT

Section 4 defines the geometric construction and NGI pipeline, and Section 5.1 specifies the backbones, baselines, and evaluation protocol. Appendix J provides derivations and implementation notes; Appendix G describes the training data and optimization recipe; and Algorithm 1 gives the training pseudocode. Appendices B and I detail the alternative encodings and ablation configurations.

REFERENCES

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.

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

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*, 2025a.

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*, 2025b.

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*, 2025a.

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*, 2025b.

Yunpeng Bai, Haoxiang Li, and Qixing Huang. Positional encoding field. In *ICLR*, 2026.

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.

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.

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.

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.

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

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

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

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.

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

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*, 2025a.

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*, 2025b.

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

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

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *ECCV*, 2024.

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

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

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

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.

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.

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)*, pp. 21807–21818. IEEE, 2024.

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.

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.

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

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv:2412.03603*, 2024a.

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

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.

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.

Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH*, 1996.

Chunyang Li, Yuanbo Yang, Jiahao Shao, Hongyu Zhou, Katja Schwarz, and Yiyi Liao. ReRoPE: Repurposing RoPE for relative camera control. *arXiv:2602.08068*, 2026a.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *NeurIPS*, 2025a.

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*, 2025b.

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*, 2026b.

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.

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.

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

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.

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

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

Helmut Pottmann and Johannes Wallner. *Computational Line Geometry*. Springer, 2001.

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.

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.

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

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

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.

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

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

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

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

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*, 2025b.

Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. In *NeurIPS*, 2021.

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.

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.

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.

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.

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.

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

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.

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

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.

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.

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.

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*, 2026a.

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*, 2026b.

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.

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.

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.

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

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.

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.

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

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

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*, 2025b.

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

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.

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.

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

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

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.

Zhenghong Zhou, Jie An, and Jiebo Luo. Latent-Reframe: Enabling camera control for video diffusion models without training. In *ICCV*, 2025.

APPENDIX

A ADDITIONAL EVALUATION

A.1 REFERENCE RESULTS UNDER RESCALED-ALIGNMENT

Table A1: 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.

	Quality	Camera controllability				Distribution		
Method	CLIP↑	RotErr↓	TransErr↓	CamMC↓	ATE↓	FVD↓	FVD_c ↓	FID↓
Wan-2.2 5B Scale								
ReCamMaster	24.97	0.131	0.297	0.402	0.322	874.30	890.52	62.53
ReRoPE	25.20	0.137	0.241	0.366	0.307	684.57	650.31	60.77
UCPE	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	25.31	0.109	0.221	0.335	0.286	675.23	697.10	59.21
ReRoPE	25.85	0.114	0.185	0.290	0.243	493.30	525.61	49.18
UCPE	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**

ReCamMaster (Bai et al., 2025a); ReRoPE (Li et al., 2026a); UCPE (Zhang et al., 2026).

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 A1 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.

A.2 VBENCH EVALUATION OF MOTION, CONSISTENCY, AND VISUAL 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 (Huang et al., 2024) implementation under its default configuration and apply the same metrics to the corresponding real videos, reported as ground truth in Table 2.

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 2, SCoPE ranks first among the generated methods on all six dimensions. Most notably, 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.

A.3 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 A.2. 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 A2 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.

Table A2: 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.

Method	RotErr ↓	TransErr ↓	CamMC ↓	ATE ↓	FVD ↓	FVD_c ↓
UCPE	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**	**1.252/0.279**	**1.217/0.208**	**529.0**	**550.8**

UCPE (Zhang et al., 2026).

Results. Table A2 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%, 27%, 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.

A.4 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 computation is $\mathcal{O}(Nd)$, which is small relative to the $\mathcal{O}(N^2 d)$ self-attention computation. UCPE instead uses a parallel

Table A3: 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.

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

UCPE (Zhang et al., 2026); PRoPE (Li et al., 2025a).

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 A3, 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.

B Camera-Aware PE Design Space

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.

Table A4: 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.

	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**

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 fine-tuning budget, suggesting disruption of the pretrained positional prior. We therefore follow the frequency-splitting strategy of ReRoPE (Li et al., 2026a): each alternative retains the high-frequency half of temporal RoPE and replaces its low-frequency half with a camera-derived transformation. This 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 A4, 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.

C Additional Qualitative Results

C.1 Out-of-Distribution Comparison



Figure A1: 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 (Bai et al., 2025a) and UCPE (Zhang et al., 2026) fail to follow the trajectory.

C.2 In-Domain Gallery

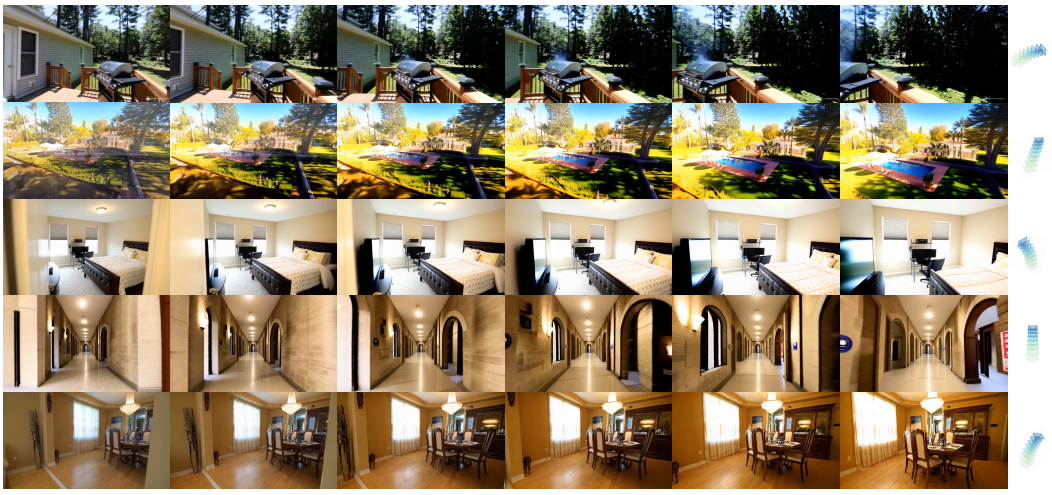

Figure A2: Gallery of SCoPE on diverse scenes. Each row is a different first frame driven by a distinct target camera trajectory; frames go left to right along time, and the rightmost column shows the input trajectory as a stack of camera frusta.

C.3 Cinematic Multi-Trajectory Gallery

Figure A3 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.



Figure A3: 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.

D Analysis of Ray Encoding and Attention

Geometry-only responses to camera motion. Using a checkpoint saved at training step 36,000, we inspect the learned geometry-only interaction by fixing a query ray and evaluating the key ray at the same image coordinates in a moved camera. For each layer–head pair h, we plot $\Delta D_h(b) = D_h(b) - D_h(0)$, where b denotes the swept rotation or translation and $b = 0$ is the static reference setting. Figure A4 shows sign-changing rotational responses in many heads and nonzero responses to pure translation. Under pure translation, ray directions remain fixed while camera origins change, so the latter responses demonstrate sensitivity to camera position carried by the Plücker moments. The variation in response sign and magnitude across heads shows that camera motion modulates their geometry-only scores in different ways. The learned interaction thus supplies distinct responses to changes in both viewing direction and camera position.

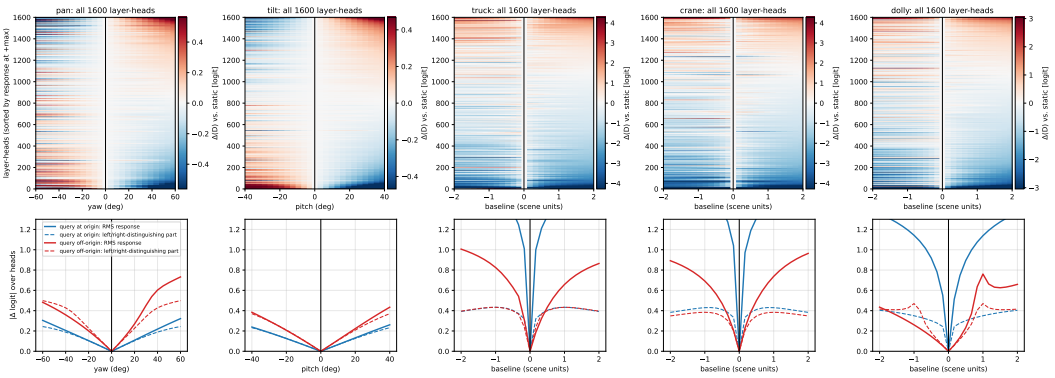

Figure A4: **Geometry-only responses to five camera motions.** Columns show pan, tilt, truck, crane, and dolly motions. For the three translation motions, "baseline" denotes signed camera displacement along the respective axis in scene units. Top: each row represents one of 1,600 layer–head pairs, ordered by response at the positive sweep endpoint. Red/blue indicates an increase/decrease in the geometry-only score relative to the static setting; each panel has its own color scale. Bottom: solid curves show RMS response magnitudes across heads; dashed curves summarize the component that distinguishes opposite motion directions. Blue curves use a query camera at the coordinate origin; red curves use one away from it.

Spatial preference of full-model attention. We next examine attention under pan and truck motions in four scenes. Figure A5 reports the normalized preference $P = (R - L)/(R + L)$, where R and L denote right- and left-side attention mass. Relative motion is measured between the key and query cameras. At denoising step 13, the layer 16–32 summary is positive for negative relative yaw or lateral translation and negative for positive relative motion. The same sign pattern appears in both left-moving and right-moving runs, suggesting that the preference tracks the relative camera configuration across trajectories. The layer-wise maps show coherent spatial biases across a range of layers, with weaker responses near the input. These trends are consistent with the use of camera geometry to organize attention across frames, complementing the quantitative component ablations in Table 4.

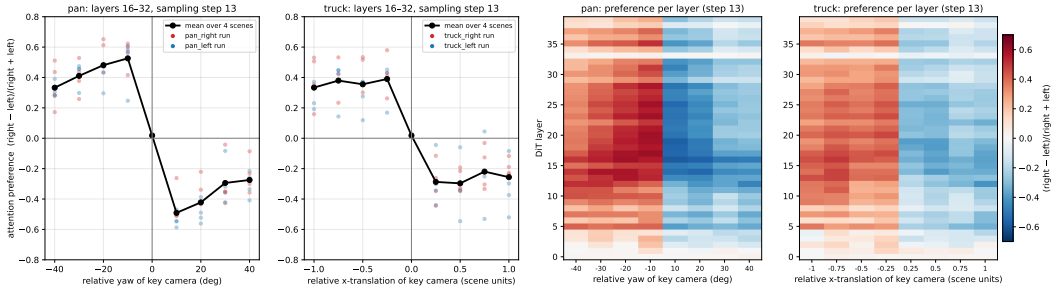

Figure A5: **Spatial attention under relative camera motion.** Left two panels: attention preference for pan and truck motions, summarized over layers 16–32 at denoising step 13. Black curves show means over four scenes; red and blue points denote right-moving and left-moving runs, respectively. Horizontal axes give the key camera's yaw or lateral displacement relative to the query camera. Right two panels: each row shows the preference at one DiT layer, in network order. Heatmap colors encode attention preference: red (positive) indicates more attention to the right, blue (negative) more to the left, and white approximately balanced attention.

E Backbone Notation

We follow the latent-video diffusion / flow-matching setup (Ho et al., 2020; Lipman et al., 2023; Blattmann et al., 2023; Yang et al., 2025; Wan et al., 2025). 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 (Vaswani et al., 2017) with query/key/value projections $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ produces, for each token i,

$$q_i = W_Q\, x_i, \quad k_i = W_K\, x_i, \quad v_i = W_V\, x_i, \tag{11}$$

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 (Henry et al., 2020) (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):

$$\tilde{q}_i = R(u_i, v_i, t_i)\, q_i, \quad \tilde{k}_j = R(u_j, v_j, t_j)\, k_j, \tag{12}$$

where $R(\cdot)$ is a block-diagonal rotation built from per-axis 1D RoPE (Su et al., 2024; Heo et al., 2024). 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. (12); SCoPE adds an additive term that is computed from the 3D camera ray of each token.

F Extended Related Work

Camera control by conditioning. Adapter methods attach modules outside the pretrained attention, encoding extrinsics as per-frame tokens (Wang et al., 2024), rasterizing per-pixel Plücker maps

into a ControlNet-style branch (He et al., 2025a), concatenating camera embeddings with epipolar cross-view attention (Xu et al., 2024), or conditioning trainable cross-attention and LoRA modules (Bahmani et al., 2025a;b). Successors condition directly on pose and time (Van Hoorick et al., 2024; Watson et al., 2025), steer trajectory or dimension adapters (Yang et al., 2024; Xiao et al., 2025; Sun et al., 2025a), and scale the recipe to dynamic scene exploration (He et al., 2025b; Huang et al., 2026). Rendering methods reconstruct a proxy and re-render it along the target trajectory as a pixel-aligned condition (Yu et al., 2025b;a; Gu et al., 2025; Bian et al., 2025; Ren et al., 2025; Li et al., 2025b; Lee et al., 2026), or steer a frozen model training-free through warped views and rearranged point clouds (You et al., 2025; Hou & Chen, 2025; Song et al., 2026). ReCamMaster and multi-camera variants concatenate view tokens and let the native attention discover correspondence on its own (Bai et al., 2025a;b; Kuang et al., 2024; Xu et al., 2025), while Go-with-the-Flow and Latent-Reframe move the camera into noise and latents, a route orthogonal to ours and combinable with it (Burgert et al., 2025; Zhou et al., 2025). 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 (Bai et al., 2026; Xu et al., 2026; Wang & He, 2026). 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 (He et al., 2020; Du et al., 2023). GTA (Miyato et al., 2024) 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 (Kong et al., 2024b; Safin et al., 2023). PRoPE generalizes the transform from extrinsics to full camera frustums while keeping the patch RoPE on separate channels (Li et al., 2025a), and interactive world models have already adopted it (Sun et al., 2025b). 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 (Zhang et al., 2026). ReRoPE replaces low-frequency temporal RoPE with relative projective camera transforms and fine-tunes the backbone's self-attention layers (Li et al., 2026a). ViewRope rotates query and key subvectors by viewing-ray direction on the same backbone we use (Xiang et al., 2026). 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 handling includes near-plane normalization in UCPE and maximum-translation normalization in ReRoPE. The latter removes global translation scale from the pose representation; SCoPE instead retains log moment magnitude as an explicit scale cue and learns how it gates geometric features. SCoPE changes geometry from a transform on content to a coordinate of the token, added in the pretrained attention with no parallel branch or second attention. The line keeps moving. ReDirector folds cameras into RoPE phase shifts (Park et al., 2026), URoPE carries relative embeddings across geometric spaces (Xie et al., 2026a), and DPPE decouples rotation from translation for scaling (Kenney & Suzuki, 2026). CRePE adds depth distributions to UCPE's rays, an upgrade the additive form can adopt unchanged (Jin et al., 2026), CameraAnything applies ReDirector's rotary phases to intrinsic and extrinsic editing (Li et al., 2026b), and RayRoPE, no relation despite the name, targets feedforward reconstruction (Wu et al., 2026).

Positional encodings and ray-space representations. Relative positional encodings enter the score as additive terms in Transformer-XL and T5 (Dai et al., 2019; Raffel et al., 2020), or as rotations that collapse indices into offsets in RoPE and its factorized video form (Su et al., 2024; Heo et al., 2024). 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 (Adelson & Bergen, 1991; Levoy & Hanrahan, 1996; Gortler et al., 1996), neural fields from NeRF to light field networks and light-field transformers learn scene functions on ray inputs (Mildenhall et al., 2020; Sitzmann et al., 2021; Wang et al., 2022; Suhail et al., 2022; Attal et al., 2022), and SE(3) equivariant transformers operate in ray space (Xu et al., 2023). Camera pose can be estimated as a bundle of rays (Zhang et al., 2024), world models generate autoregressively in ray space (Xie et al., 2026b), and Rays as Pixels folds camera trajectories into the video distribution (Jang et al., 2026). 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 (Zhang et al., 2023), and SCoPE carries it into the positional encoding.

G 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.

G.1 Data Sources

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

- **RealEstate10K** (RE10K) (Zhou et al., 2018): about 10K real-estate walkthrough clips with COLMAP-style SfM poses (Schönberger & Frahm, 2016). Smooth, primarily forward camera motion through indoor scenes; the canonical benchmark for camera-controllable video synthesis.
- **DL3DV** (Ling et al., 2024): 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** (Zhang et al., 2026) (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** (Zhou et al., 2026): open-world video with relatively dynamic content, with camera extrinsics estimated by DROID-SLAM (Teed & Deng, 2021). 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.

G.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:

$$\tilde{\mathbf{o}}_t \;=\; \eta \cdot \mathbf{o}_t, \qquad \tilde{\mathbf{m}}_i = \tilde{\mathbf{o}}_t \times \mathbf{d}_i = \eta \cdot \mathbf{m}_i. \tag{13}$$

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.

G.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:

$$\tilde{\mathbf{o}}_t \;=\; \frac{\eta}{z_n} \cdot \mathbf{o}_t. \tag{14}$$

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.

G.4 Optimization Recipe

Backbone and conditioning. We start from a pretrained 5B-parameter Wan2.2-TI2V-5B backbone (Wan et al., 2025) 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 $\alpha = 1$ at initialization. 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 2×10^{-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 $\mathrm{cfg_scale} = 5.0$.

H Training Pseudocode

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: $\mathbf{o}_t \leftarrow (\eta_{\mathrm{ds}} / \max(z_n, \epsilon)) \cdot \mathbf{o}_t$ ▷ Appendix G.2–G.3
Compute per-token rays $\{(\mathbf{d}_i, \mathbf{m}_i)\}$ from $\{T_t, K_t\}$ and the latent patch grid ▷ Section 3.1
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
  $q, k \leftarrow \mathrm{QKNorm}(q),\ \mathrm{QKNorm}(k)$
  $q, k \leftarrow R(u, v, t) \cdot q,\ R(u, v, t) \cdot k$ ▷ 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. (10)
  $q \leftarrow q + \alpha \cdot g \odot N_q^\ell(E_q^\ell f^{(q)})$
  $k \leftarrow k + \alpha \cdot g \odot N_k^\ell(E_k^\ell f^{(k)})$
  $h \leftarrow h + \mathrm{Attn}(q, k, v)\, W_O^\ell$
  $h \leftarrow h + \mathrm{CrossAttn}(h, c) + \mathrm{FFN}(h)$
end for
Compute diffusion / flow loss between predicted noise/velocity and target and backpropagate.

I Component Ablation Details

The following discussion expands the component ablations in Table 4; Section 5.4.1 defines the interventions on terms (B), (C), and (D), whose complementary roles are analyzed in Appendix I.1.

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 J.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 RMSNorm** 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.

I.1 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. (4). 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.

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.

J Additional Derivations

J.1 Bilinearity of the Plücker Reciprocal Product

We restate the elementary fact used in Section 3.2. 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

$$\langle \mathbf{r}_i, \mathbf{r}_j \rangle_{\mathrm{Pl}} = \mathbf{d}_i \cdot \mathbf{m}_j + \mathbf{d}_j \cdot \mathbf{m}_i. \tag{15}$$

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.

J.2 Coplanarity Criterion

The supporting lines of two rays $\mathbf{r}_i, \mathbf{r}_j$ are coplanar (intersecting, parallel, or coincident) if and only if $\langle \mathbf{r}_i,\, \mathbf{r}_j\rangle_{\mathrm{Pl}} = 0$. The proof is standard (Hartley & Zisserman, 2004; Pottmann & Wallner, 2001) 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.

J.3 SE(3) Invariance

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

$$g\cdot(\mathbf{d}, \mathbf{m}) \;=\; (R\mathbf{d},\, R\mathbf{m} + \mathbf{t}\times R\mathbf{d}). \tag{16}$$

Substituting into Eq. (2),

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

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 $\langle g\cdot\mathbf{r}_i,\, g\cdot\mathbf{r}_j\rangle_{\mathrm{Pl}} = \langle \mathbf{r}_i,\, \mathbf{r}_j\rangle_{\mathrm{Pl}}$.

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

Recall (Eq. (5)) that, with the Q/K flip,

$$(\mathrm{D}) = \mathbf{r}_i^\top\, M\, \tilde{\mathbf{r}}_j, \qquad \mathbf{r}_i = (\mathbf{d}_i, \mathbf{m}_i), \quad \tilde{\mathbf{r}}_j = (\mathbf{m}_j, \mathbf{d}_j), \tag{17}$$

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

$$(\mathrm{D}) = \mathbf{d}_i^\top A\mathbf{m}_j + \mathbf{d}_i^\top B\mathbf{d}_j + \mathbf{m}_i^\top C\mathbf{m}_j + \mathbf{m}_i^\top D\mathbf{d}_j. \tag{18}$$

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 condition $E_q^\top E_k = I_6$ in Proposition 1 gives $M = I_6$, 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 diagonal blocks A, D control the reciprocal-product-like part, while the off-diagonal blocks B, C control direction–direction and moment–moment couplings that are not present in the classical reciprocal product.

J.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 Eq. (3) 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. (5) reads

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

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 canonical configuration ($E_q^\top E_k = I_6$) 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.

J.6 Implementation Notes

Numerical safety. We clamp $\|\mathbf{m}\|$ at $\epsilon = 10^{-6}$ before normalization (Eq. (6)) 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 both weight matrices with zero biases would block gradients to the weight matrices: the output layer receives zero hidden activations, and its zero weights block gradients to the input layer. We therefore zero only the output layer and use Kaiming initialization for the input layer. With $\alpha = 1$ at initialization, the branch starts at zero while gradients can reach its output layer.