CRePE: Curved Ray Expectation Positional Encoding for Unified-Camera-Controlled Video Generation

Seonghyun Jin[1,*] **Youngmin Kim**[1,2,*,‡] **Sunwoo Park**[1,*] **Jong Chul Ye**[1,†]
[1]Kim Jaechul Graduate School of Artificial Intelligence, KAIST, Seoul, Republic of Korea
[2]Korea University, Seoul, Republic of Korea
{jinotter3,sunwoo_p,jong.ye}@kaist.ac.kr zeromin03@korea.ac.kr

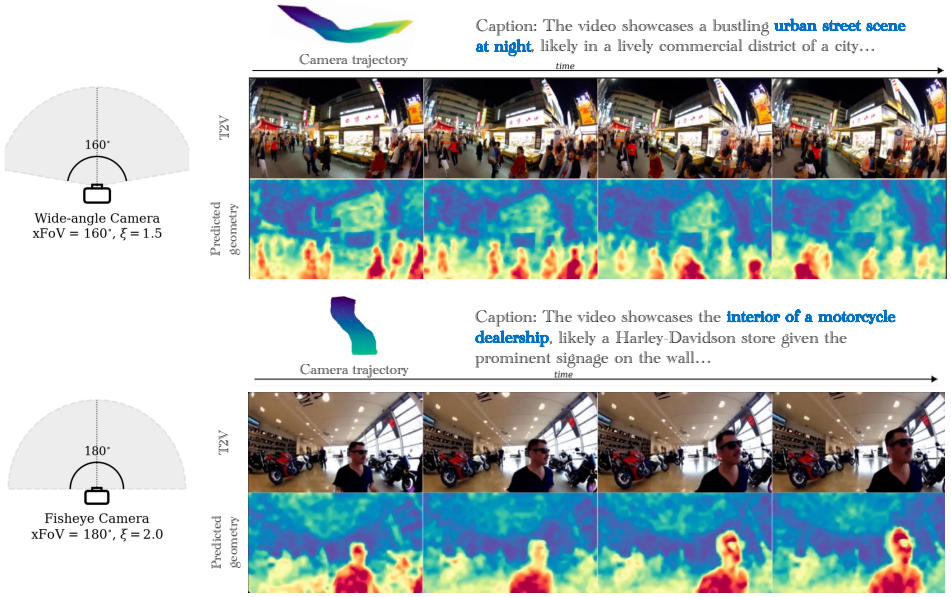


Figure 1: **Fisheye and wide-lens camera-conditioned video generation.** Each row shows a video generated by **CRePE** under a non-pinhole camera trajectory, alongside the internal geometry map the model estimates and utilizes during generation — the per-token expected radial distance along its viewing ray, integrated into the rotary phasor at each denoising step. Top: wide-angle camera trajectory with the corresponding generated frames and geometry map. Bottom: fisheye trajectory with its generated frames and geometry map. The geometry map is internal to CRePE rather than an external input: it is a latent positional variable decoded from the geometry head, not the output of a depth decoder or a metric depth estimate.

Abstract

Camera-conditioned video generation requires positional encoding that remains reliable under changes in camera motion, lens configuration, and scene structure. However, existing attention-level camera encodings either provide ray-only camera signals or rely on pinhole camera geometry, limiting their applicability to general camera control under the Unified Camera Model, including wide-angle and fisheye lenses. To address this limitation, we propose **Curved Ray Expectation Positional Encoding (CRePE)**. CRePE represents each image token as a depth-aware positional distribution along its source ray, providing a

*Equal contribution.
†Corresponding author.
‡Work done during an internship at KAIST.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Unified Camera Model-compatible positional encoding that naturally captures the curved-ray geometry of wide-angle and fisheye cameras. CRePE is implemented through a **Geometric Attention Adapter** added to frozen video DiTs, injecting token-wise scene-distance information into selected attention layers and stabilizing it with pseudo supervision from a monocular geometry foundation model. This design leads to more stable camera control with competitive video quality in camera-conditioned video generation under the Unified Camera Model, and the resulting model transfers zero-shot to real fisheye and diverse pinhole videos. Controlled positional-encoding ablations show improvements over a RayRoPE-style endpoint approximation, demonstrating the effectiveness of curved-ray-aware positional encoding, particularly for non-pinhole cameras. Furthermore, by extending the same positional-encoding pathway to external geometry control through **Radial MixForcing**, CRePE supports external radial-map control for scene-geometry-conditioned generation and source-video motion transfer beyond camera control. Project page: `https://crepe-neurips2026.github.io/`; code: `https://github.com/jinotter3/CRePE`.

1 Introduction

Camera-conditioned video generation [1] requires models to synthesize videos that follow specified camera trajectories and lens configurations. Recent studies have expanded the scope of camera conditioning from direct camera conditioning to attention-level positional encoding [2, 3]. By incorporating camera geometry directly into positional encoding, video diffusion transformers [4, 5] can more explicitly model the relative positional relationships among latent tokens under view changes. In particular, UCPE [3] encodes the viewing ray of each token based on a unified central-camera representation, demonstrating the possibility of camera control under the Unified Camera Model (UCM), including wide-angle and fisheye cameras. This marks an important step toward handling diverse lens models within a unified positional-encoding framework.

However, relative-ray-based positional encoding still has a limitation. A viewing ray indicates the direction associated with a token, but it does not specify where the underlying scene content lies along that ray. When the camera moves, token correspondences depend not only on ray direction but also on the scene position along the ray. Thus, while ray-only encoding provides a stable way to inject camera geometry, it remains limited in representing more refined positional relationships coupled with scene geometry. This limitation becomes more pronounced for wide-angle and fisheye cameras, where projection geometry changes nonlinearly.

To address this problem, we propose **Curved Ray Expectation Positional Encoding (CRePE)**. Instead of representing each token only by its viewing ray, CRePE models it as a depth-aware positional distribution along the source ray, reflecting possible scene positions associated with the token. CRePE is designed to be compatible with the Unified Camera Model, naturally capturing the nonlinear projection geometry that arises under diverse lens configurations. This allows the positional encoding to account for how tokens may align with the underlying scene structure and to refine token relationships under view changes. CRePE builds on RayRoPE's principle of averaging RoPE over ray-position uncertainty [2], replacing its pinhole endpoint interval with a log-radial distribution projected through the UCM (Table 1).

To apply CRePE to camera-conditioned video generation, we introduce a **Geometric Attention Adapter** for frozen video DiTs. The adapter predicts token-wise scene distance and incorporates it into the positional relationships used in attention. Pseudo radial-distance supervision from a monocular geometry foundation model serves as an auxiliary geometric anchor, helping the predicted geometry signal remain stable during training; it is not required at inference. Through this design, CRePE extends ray-only positional encoding into a scene-geometry-aware positional encoding while preserving the adapter-based camera-conditioning structure.

We evaluate CRePE against camera-conditioned video generation baselines [1, 3] and analyze the effect of the positional encoding itself through controlled ablations. CRePE improves overall camera control under the Unified Camera Model with competitive video quality, while showing a generally favorable performance trend over a RayRoPE-style endpoint approximation in the controlled setting. Furthermore, inspired by YoNoSplat [6], we extend the same positional-encoding pathway to

Table 1: **Geometry-aware positional encodings for camera-conditioned video DiTs.** The RayRoPE column describes the original pinhole design; our $K = 2$ endpoint ablation instead shares CRePE's UCM projection. "–" marks axes not applicable to a method without a learnable depth axis.

Property	UCPE [3]	RayRoPE [2]	CRePE (ours)
Non-pinhole / UCM projection	✓	✗	✓
Ray-position-adaptive phase	✗	✓	✓
Curved projected-path integration	–	✗	✓
Pseudo-GT geometric grounding	–	✗	✓
Joint camera+depth control interface	✗	✗	✓

external geometry control through **Radial MixForcing**; CRePE aligns tokens with geometry-aware positional information in a compatible form, suggesting its potential for external radial-map control in scene-geometry-conditioned generation and source-video motion transfer. Quantitative DAVIS and non-pinhole PanShot experiments verify adherence to the supplied maps.

Our contributions are summarized as follows:

- We propose **Curved Ray Expectation Positional Encoding**, a Unified Camera Model-compatible positional encoding that considers not only ray direction but also possible scene positions along the ray, enabling depth-aware positional encoding for curved-ray geometry.
- We apply CRePE to camera-conditioned video generation through a **Geometric Attention Adapter** for frozen video DiTs, incorporating token-wise scene distance into attention and stabilizing geometry prediction with pseudo radial-distance supervision.
- We provide controlled comparisons showing that CRePE exhibits a generally favorable performance trend over existing camera-conditioned baselines and a RayRoPE-style endpoint approximation, supported by supervision-free and zero-shot analyses, and further demonstrate that the same pathway can be extended through **Radial MixForcing** toward external radial-map control for scene-geometry-conditioned generation and source-video motion transfer.

2 Related Work

Camera-conditioned video generation. Camera-conditioned video models control generated motion using pose embeddings, camera trajectories, rendered geometric guidance, or dedicated camera branches [7–10, 1]. ReCamMaster [1] is a strong external baseline for trajectory-conditioned generation, but its formulation is tied to a pinhole setting. UCPE [3] instead conditions video diffusion with unified-camera rays and is therefore the closest baseline for our setting. CRePE builds on the UCPE interface but adds a token-wise ray-position variable inside attention. By contrast, depth-conditioned video generators inject dense structural features into a generative backbone [11–15].

Geometric positional encoding. Rotary positional encoding [16] makes attention depend on relative position. Multi-view variants replace image-grid position with camera-aware quantities: CaPE [17], GTA [18], and PRoPE [19] encode relative camera geometry; raymap methods concatenate ray origins or directions to features [20–23]; RayRoPE [2] predicts a token-wise ray-position interval under pinhole projection and applies expected RoPE after projection. CRePE shares this principle of averaging RoPE over ray-position uncertainty but differs from RayRoPE in two ways: it uses UCM projection for unified central cameras, and it integrates the phasor over a curved projected trajectory rather than a single endpoint-defined interval.

Central-camera geometry and radial distance. Wide-angle and fisheye videos are better described by central-camera models such as UCM than by pinhole intrinsics. Radial distance—distance from the camera center along the viewing ray—is a natural scalar for such cameras because it remains defined regardless of lens distortion. Recent geometric foundation models provide dense monocular scene geometry across diverse cameras [24–27]. We use these predictions only as pseudo radial-distance anchors for CRePE's geometry head.

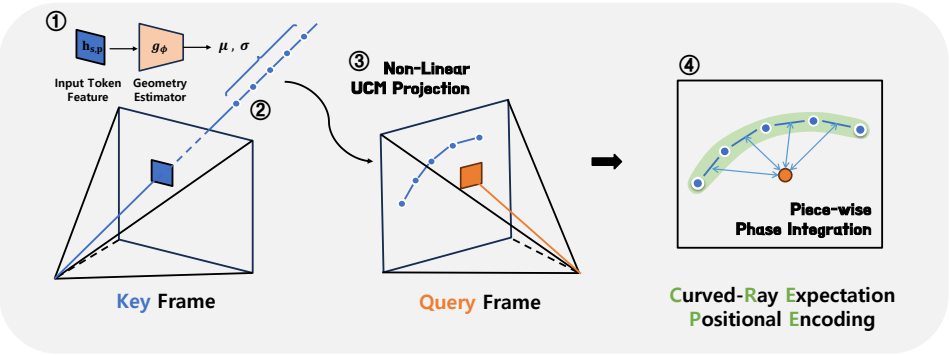


Figure 2: **CRePE.** A shared geometry head g_ϕ predicts a log-distance center μ and uncertainty $|\sigma|$ from each key-side token feature $\mathbf{h}_{s,p}$, defining K breakpoints $\{r_k\}$ along the source viewing ray (①). The breakpoints are unprojected to 3D via the source UCM model $\Phi_{\xi_s}^{\mathrm{UCM}}$ (②), transported into the query frame and forward-projected via $\Pi_{\xi_q}^{\mathrm{UCM}}$, tracing a curved trajectory $\{\tilde{\mathbf{x}}^{(k)}\}$ in the query view (③). CRePE aggregates the expected RoPE phasor along this trajectory by analytically integrating each of the $K-1$ piecewise-linear segments (④), yielding a token-level modulation $\bar{\rho}_{f,c}$ that captures ray-position uncertainty under non-pinhole projection.

3 Method

3.1 Overview

We start from a frozen video diffusion transformer with a UCPE-style camera adapter. For each token at layer ℓ, source frame s, and patch p, the adapter has feature $\boldsymbol{h}_{\ell,s,p}$. CRePE adds a geometry head g_ϕ that predicts a log radial-distance center μ and interval width σ:

$$(\mu_{\ell,s,p}, \sigma_{\ell,s,p}) = g_\phi(\boldsymbol{h}_{\ell,s,p}). \tag{1}$$

These predictions define a token-wise radial-distance distribution. CRePE converts this distribution into RoPE coefficients by lifting radial-distance breakpoints along the source ray and projecting them into each query camera. The resulting coefficients modulate attention in a selected set of middle DiT layers. All other adapter layers keep UCPE's relative-ray modulation (Figure 2).

3.2 Curved-Ray Expected RoPE

CRePE interprets the predicted interval as a uniform distribution in log radial distance. Specifically, for each source token we define $z = \log r \sim \mathcal{U}(\mu - |\sigma|, \mu + |\sigma|)$, where r denotes the radial distance measured from the source camera center along the token's UCM ray. We then discretize this log-radial interval with K uniformly spaced breakpoints and exponentiate them:

$$z_k = \mu - |\sigma| + \frac{k-1}{K-1}(2|\sigma|), \qquad r_k = \exp(z_k), \qquad k = 1, \ldots, K. \tag{2}$$

Each r_k is lifted along the source UCM ray, transformed into the query camera, and projected through the query UCM to obtain a query-side coordinate $\tilde{\mathbf{x}}^{(k)}$. For a fixed RoPE frequency ω_f and coordinate channel c, define $\theta_{f,c}^{(k)} = \omega_f \tilde{x}_c^{(k)}$. CRePE treats consecutive breakpoints as piecewise-linear phase segments and integrates the phasor analytically:

$$\bar{c}_{f,c} = \frac{1}{K-1}\sum_{k=1}^{K-1} \frac{\sin\theta_{f,c}^{(k+1)} - \sin\theta_{f,c}^{(k)}}{\theta_{f,c}^{(k+1)} - \theta_{f,c}^{(k)}}, \quad \bar{s}_{f,c} = \frac{1}{K-1}\sum_{k=1}^{K-1} \frac{\cos\theta_{f,c}^{(k)} - \cos\theta_{f,c}^{(k+1)}}{\theta_{f,c}^{(k+1)} - \theta_{f,c}^{(k)}}. \tag{3}$$

The pair $(\bar{c}_{f,c}, \bar{s}_{f,c})$ forms the rotary coefficient $\bar{\rho}_{f,c} = \bar{c}_{f,c} + i\,\bar{s}_{f,c}$ for that coordinate and frequency. Eq. (3) therefore approximates $\mathbb{E}_{z\sim\mathcal{U}(\mu-|\sigma|,\mu+|\sigma|)}\left[\exp\left(i\omega_f \tilde{x}_c(z)\right)\right]$, where $\tilde{x}(z)$ is the query-view projection of the 3D point obtained by lifting radial distance $r = \exp(z)$ along the source UCM ray. Importantly, the expectation is taken over log radial distance, not over image coordinates or metric distance directly. CRePE thus computes the expected phasor along the projected trajectory, rather than applying RoPE at a single averaged position $\exp(i\omega_f \mathbb{E}[\tilde{x}_c])$. When $K = 2$, the path degenerates

to an endpoint interval; with $K > 2$, CRePE captures non-affinity of the projected phase, including the curvature induced by UCM projection (Appendix I). We use $K = 5$ by default. Implementation details, including the UCM projection equations, bounded query-view coordinates, patch-offset rays, and numerical handling of small phase differences, are given in Appendices A and B.

CRePE inside attention. We apply the expected phasor in the geometric attention branch, following the relative-transform pattern of the UCPE adapter (Figure 6). Let $i = (q, p_q)$ denote a query token in query frame q, and let $j = (s, p_s)$ denote a key/value token in source frame s. For a CRePE-enabled layer ℓ, the geometry head predicts $(\mu_{\ell,s,p_s}, \sigma_{\ell,s,p_s})$ from the key-side feature $\boldsymbol{h}_{\ell,s,p_s}$. Given the relative camera transform from source frame s to query frame q, Eqs. (2)–(3) produce a query-conditioned expected rotary coefficient $\bar{\rho}_{\ell,q\leftarrow s,p_s} = \{(\bar{c}_{f,c}, \bar{s}_{f,c})\}_{f,c}$. We convert this coefficient into the corresponding block-diagonal rotary operator $\bar{R}_j \equiv \bar{R}_{\ell,q\leftarrow s,p_s}$; the query token uses its own coefficient in its own frame, $\bar{R}_i \equiv \bar{R}_{\ell,q\leftarrow q,p_q}$. Queries, keys, values, and outputs are then transformed as

$$
\begin{aligned}
Q_{\ell,i} &= \bar{R}_i^\top W_Q \boldsymbol{h}_{\ell,i}, \qquad K_{\ell,j} = \bar{R}_j^\top W_K \boldsymbol{h}_{\ell,j}, \qquad V_{\ell,j} = \bar{R}_j^\top W_V \boldsymbol{h}_{\ell,j}, \\
A^{\mathrm{crepe}}_{\ell,i,j} &= \mathrm{softmax}_j\left(\frac{Q_{\ell,i}^\top K_{\ell,j}}{\sqrt{d}}\right), \qquad O^{\mathrm{crepe}}_{\ell,i} = \bar{R}_i \sum_j A^{\mathrm{crepe}}_{\ell,i,j} V_{\ell,j}.
\end{aligned} \tag{4}
$$

Each logit thus depends on $\bar{\rho}_{\ell,q\leftarrow q,p_q}\bar{\rho}^*_{\ell,q\leftarrow s,p_s}$, the relative expected phase between the query and the key in the query view. For a unit-magnitude phasor, $\bar{R}^\top = \bar{R}^{-1}$, which recovers the $D^\top/D^{-1}/D$ pattern of Figure 6; because $|\bar{\rho}_{f,c}| \le 1$, we use the transpose. The output of this branch is passed through the same zero-initialized output projection used by the adapter and added residually to the frozen DiT feature. CRePE is used only in the selected middle layers; the remaining layers keep the UCPE relative-ray modulation. This preserves the stable ray-only camera signal while injecting depth-aware projected-path modulation where radial-distance information is most recoverable.

3.3 Geometric Attention Adapter and Radial-Distance Supervision

The geometry head in Eq. (1) affects attention directly, so incorrect predictions can become a harmful positional shortcut. We therefore supervise it with pseudo radial-distance maps from a monocular geometry foundation model, UniK3D [27]. We first apply validity filtering to the raw metric pseudo distance $\hat{r}^{\mathrm{m}}$: a target is invalid if UniK3D fails, is marked unreliable by our validity filters, returns a non-finite or non-positive value, falls outside the supported distance range, or predicts $\hat{r}^{\mathrm{m}} > r_{\max} = 20\,\mathrm{m}$. Invalid or far-field values are not clipped into pseudo ground truth. Only valid metric targets are normalized by a per-clip near-distance statistic, pooled to the token grid, and used as normalized targets $\hat{r}$ in the loss. From (μ, σ) we decode the predicted normalized radial distance $\bar{r} = \exp(\mu)$ and an uncertainty scale s derived from the log interval. The auxiliary loss and the full objective are

$$
\mathcal{L}_{\mathrm{rad}} = \frac{1}{|\mathcal{T}|}\sum_{n\in\mathcal{T}}\left(\frac{|\bar{r}_n - \hat{r}_n|}{s_n} + \alpha \log s_n\right), \qquad \mathcal{L} = \mathcal{L}_{\mathrm{diff}} + \lambda_{\mathrm{rad}}\mathcal{L}_{\mathrm{rad}}, \tag{5}
$$

where $\mathcal{T}$ is the valid token set indexed by n, $\alpha = 1$, $\mathcal{L}_{\mathrm{diff}}$ is the backbone diffusion objective, and $\lambda_{\mathrm{rad}} = 10^{-3}$. We supervise the geometry head at a single selected layer, which trains the head shared across all CRePE-modulated layers, and keep the video DiT backbone frozen. Only 0.061% of pixels are true UniK3D failures; after the 20 m far-field cut, 56.3% of pixels (55.7% of tokens) remain valid, and the excluded rays (mostly sky) use the model's own prediction. In practice, this radial-distance anchor is important: diffusion-only CRePE can learn plausible ray-position maps early, but degrades under longer optimization (Section 4.5).

3.4 Radial MixForcing

Radial MixForcing is a training-time teacher-substitution algorithm for the CRePE ray-position pathway. Its purpose is to teach the adapter to accept externally supplied radial maps while preserving a safe camera-only fallback. Let $(\mu_{\mathrm{pred}}, \sigma_{\mathrm{pred}})$ be the geometry-head output, let $\hat{r}^{\mathrm{m}}$ be the raw metric pseudo radial distance, let $\hat{r}$ be its per-clip-normalized value when valid, let $\mu_{\mathrm{gt}} = \log \hat{r}$, and let M be a sampled teacher-substitution mask. We define pseudo-GT validity on the raw metric output

as V_{gt} below; distances above the validity threshold are therefore treated as unknown, not as valid targets clipped to $20\,\mathrm{m}$. CRePE consumes the effective radial distribution

$$V_{\mathrm{gt}} = \mathbf{1}[\mathrm{valid}_{\mathrm{UniK3D}}(\hat{r}^{\mathrm{m}}) \wedge 0 < \hat{r}^{\mathrm{m}} \leq r_{\max}], \qquad (\mu_{\mathrm{eff}}, \sigma_{\mathrm{eff}}) = \begin{cases} (\mu_{\mathrm{gt}}, \sigma_T) & \text{if } M \wedge V_{\mathrm{gt}}, \\ (\mu_{\mathrm{pred}}, \sigma_{\mathrm{pred}}) & \text{otherwise,} \end{cases} \tag{6}$$

where σ_T is a narrow teacher interval. The second branch covers every invalid ray: no teacher geometry is injected and that ray uses the model prediction. The auxiliary loss in Eq. (5) is still applied only on valid tokens, so the head learns from reliable pseudo radial distances while invalid regions remain self-predicted. At inference, an external radial map follows the same rule after validity filtering and scale normalization: valid external rays may replace the predicted tensor, while invalid or missing rays fall back to $(\mu_{\mathrm{pred}}, \sigma_{\mathrm{pred}})$. We use the supervised CRePE model for the main camera-only benchmark and the MixForcing-trained model (CRePE-Mix) for external-radial-map control. The schedule and teacher-width details are in Appendix F; an inference-time sweep of σ_T (trained at 0.1) over a $20\times$ range changes CamMC by only 3.9% (Appendix K).

3.5 Layer Placement

Applying CRePE to every layer is unnecessary and can degrade visual quality. We probe the frozen DiT layers for recoverable radial-distance information and, with a placement ablation under the reduced-budget protocol of Section 4.4 (1K steps, $\lambda_{\mathrm{rad}} = 0$, pose stride 4), find that middle layers provide the best tradeoff between geometric control and generation quality. Our default model applies CRePE to the middle 10 blocks and retains UCPE modulation elsewhere. The probing curve and placement ablation are in Appendix C.

4 Experiments

4.1 Setup

Backbone and training. We use Wan2.1-T2V-1.3B [5] as the video diffusion backbone and keep its weights frozen. CRePE is trained through the camera-conditioning adapter for 10K steps using AdamW. The default setting applies CRePE to the middle 10 DiT blocks, uses $K = 5$ radial-distance breakpoints, and supervises the shared geometry head at a single selected layer. Full hyperparameters are in Appendix B.

Dataset. We train and evaluate on PanShot [3], a multi-camera video dataset covering panning shots with pinhole, wide-angle, and fisheye camera settings, using its released train/test split. Each clip has a camera trajectory and UCM lens parameters. We evaluate all 340 test clips without filtering by camera-motion magnitude; reduced-budget ablations use a fixed 50-clip subset of the test split (13 pinhole, 37 non-pinhole). Pseudo radial-distance maps are produced offline with UniK3D [27] and used only for training supervision or optional external-radial-map interventions. Dataset details are in Appendix G.

Metrics. We evaluate camera controllability with rotation error (RotErr), scale-aligned translation error (TransErr), and CamMC, recovering relative pose from generated videos with ViPE [28] following UCPE [3]. We evaluate generation quality with FVD, FVD-C, FID, Inception Score, VBench [29], and CLIP-based image/text scores. Lens and orientation diagnostics, estimated with GeoCalib [30], include radial-distortion-coefficient errors (k_1, k_2) and pitch, gravity, and up-vector errors. Main comparisons recover pose at frame stride 1; the reduced-budget ablations and zero-shot studies use stride 4, so absolute pose values are comparable only within a protocol.

4.2 Baselines

The main evaluation compares against **ReCamMaster** [1] and **UCPE** [3]. ReCamMaster is the external camera-controlled generation baseline; UCPE is the controlled baseline with the same Wan backbone and adapter interface but without CRePE or radial-distance supervision. We evaluate a **RayRoPE-style endpoint PE** [2] in a separate positional-encoding ablation, where it isolates the difference between endpoint-based expected RoPE and CRePE's UCM-compatible curved-ray integration; it uses CRePE's UCM projection with $K = 2$ and is not the full RayRoPE system.

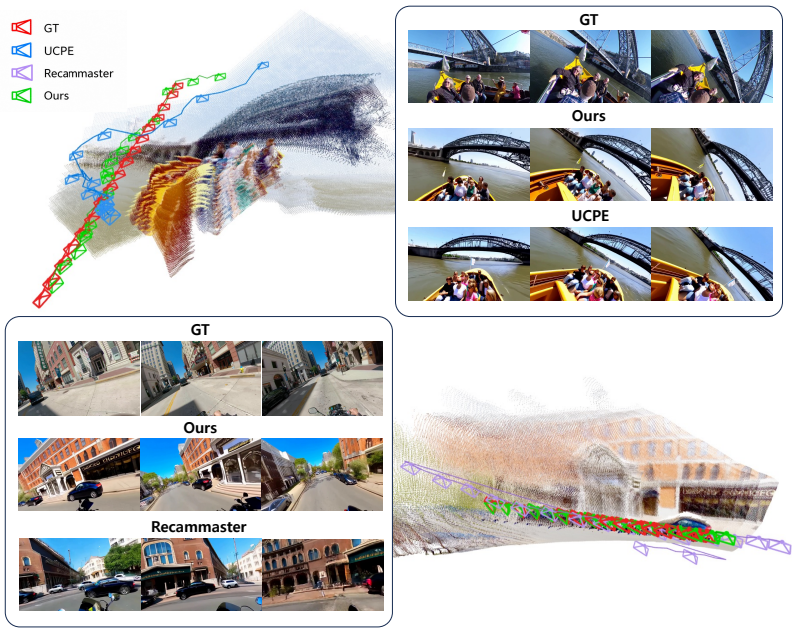


Figure 3: **Qualitative camera-control comparison.** Generated videos under matched prompts and camera trajectories. The main comparison includes ReCamMaster, UCPE, and CRePE. CRePE follows the requested camera trajectory most stably under non-pinhole lenses.

Table 2: **Pinhole and non-pinhole camera-generation comparison** (340 PanShot test clips; pose stride 1). We report representative metrics across pose control, text/image alignment, lens distortion, orientation, and VBench quality. Non-pinhole and pinhole subsets are grouped by rows for readability. CRePE consistently improves over UCPE and ReCamMaster on the representative camera-control, geometry-aware, and VBench quality metrics.

Subset	Method	Pose	Text/Image		Lens		Orientation	VBench	
		CamMC ↓	CS-I ↑	CS-T ↑	k_1 ↓	k_2 ↓	Gravity ↓	Aes. Qual. ↑	BG Cons. ↑
Non-pinhole	ReCamMaster	54.56	98.69	24.60	0.186	0.154	11.19	0.5596	0.9484
	UCPE	20.60	98.80	24.78	0.183	0.154	7.08	0.5616	0.9537
	CRePE	**19.02**	**98.81**	**24.80**	**0.157**	**0.116**	**6.27**	**0.5639**	**0.9542**
Pinhole	ReCamMaster	57.83	98.65	25.58	0.179	0.152	10.98	0.5605	0.9441
	UCPE	18.50	98.59	25.86	0.175	0.151	7.10	0.5619	0.9457
	CRePE	**15.33**	**98.65**	**25.93**	**0.153**	**0.113**	**6.13**	**0.5620**	**0.9471**

4.3 Main Camera-Controlled Generation Results

Table 2 and Figure 3 report the main comparison across lens families. CRePE improves the overall camera-conditioning profile relative to the baselines on both pinhole and non-pinhole subsets. Compared with UCPE, CRePE improves most metrics, while UCPE remains a strong baseline and is slightly better on FVD (463.5 vs. 465.3), FVD-C, and IS (Table 7); we therefore describe CRePE's video quality as competitive rather than improved. This tradeoff suggests that CRePE improves scene-geometry-aware camera following, but the direct ray/lens signal in UCPE is still useful.

4.4 Positional-Encoding Ablation: RayRoPE-Style Endpoint versus CRePE

We use the RayRoPE-style endpoint PE as a controlled ablation of the positional encoding rather than as a main camera-generation baseline. It corresponds to a two-point endpoint approximation of the ray-position interval ($K = 2$), while CRePE performs curved-ray phasor integration by sampling multiple radial-distance breakpoints along the projected trajectory. This comparison isolates whether explicitly integrating over the curved ray path provides benefits beyond the endpoint-based approximation. All variants use the same 1K-step diffusion-only objective ($\lambda_{\text{rad}} = 0$), parameter

Table 3: **Sampling-point ablation for CRePE** (1K steps, $\lambda_{\mathrm{rad}} = 0$, 50-clip subset, pose stride 4). We compare the RayRoPE-style endpoint PE ($K = 2$) and CRePE variants using representative metrics. Avg. Rank is computed among the five settings over 19 metrics, excluding IS Std. CRePE with $K = 5$ achieves the best overall balance, while $K = 3$ leads pose.

Setting	Avg. Rank ↓	CamMC ↓	Rot. ↓	Gravity ↓	CS-T ↑	FID ↓
$K = 1$ (point)	4.53	17.64	6.73	18.22	20.57	242.7
$K = 2$ (endpoint PE)	3.27	14.75	5.65	10.46	25.75	**135.3**
$K = 3$ (CRePE)	2.94	**12.31**	**4.43**	9.90	25.76	139.1
$K = 5$ (CRePE, default)	**1.71**	13.45	4.65	**9.69**	**25.76**	136.6
$K = 7$ (CRePE)	2.56	14.23	4.70	9.80	25.70	135.5

Table 4: **Controls and zero-shot transfer.** Pose stride 1 in A/B (B: 1K runs on the 50-clip subset, 10K runs on all 340 clips), 4 in C. Lower is better except VBench-OC (overall consistency).

A. Geometry injection (10K steps)

Method	CamMC	Rot.	Trans.	FVD
UCPE (no depth)	21.87	6.66	17.54	**463.5**
Depth-Concat	23.21	7.48	18.11	473.0
CRePE (phase)	**18.15**	**6.33**	**13.78**	465.3

B. Supervision and training duration

Metric	1K no	1K sup	10K no	10K sup
CamMC	**52.02**	56.27	22.19	**18.15**
RotErr	**18.00**	18.21	8.11	**6.33**
SSI-L1	0.129	**0.117**	0.133	**0.087**

C. Zero-shot transfer without tuning

Method	KITTI-360 (32 scenes, 180° fisheye)				RealCam-Vid (100 scenes)			
	CamMC	Rot.	Trans.	VBench-OC	CamMC	Rot.	Trans.	VBench-OC
UCPE	22.32	16.20	9.34	0.2097	12.17	7.02	7.00	0.1894
CRePE (no-sup)	18.18	14.01	**5.15**	**0.2198**	12.31	6.68	7.22	0.2146
CRePE	**17.01**	**12.70**	5.59	0.2137	**9.55**	**5.79**	**4.89**	**0.2148**

count, and data, with pose evaluated at stride 4; UniK3D is used neither in training nor at inference, and interior breakpoints are deterministic evaluations of the same predicted (μ, σ).

Table 3 shows that CRePE consistently improves over the endpoint approximation in overall average rank. $K = 1$ (a single point) collapses in quality (FID 242.7), and the largest pose gain comes from $K = 2$ to $K = 3$ (CamMC $14.75 \rightarrow 12.31$). With $K = 5$, CRePE achieves the best balance across pose accuracy, text/image alignment, video quality, lens distortion, and orientation metrics. Increasing the number of samples to $K = 7$ slightly improves some distortion-related metrics, but does not further improve the overall balance. We therefore use $K = 5$ as our default setting. These gains are not a sample-count effect: for an affine phase path the coefficient is independent of K, and the gains appear on both pinhole and non-pinhole clips, so they are not purely due to UCM curvature (Appendix I).

4.5 Supervision, Robustness, and Zero-Shot Transfer

Geometry as phase, not extra information. A matched Depth-Concat control predicts the same pseudo radial signal under the same loss and 10K-step schedule, but concatenates it to every token through a projection MLP (+6M parameters). It reaches comparable quality (FVD 473.0 vs. 465.3) yet trails even depth-free UCPE on pose (Table 4A), so among the tested alternatives the advantage comes from making radial distance a query-dependent phase, not from access to monocular geometry.

Supervision anchors long training. At 1K steps, $\mathcal{L}_{\mathrm{rad}}$ provides no camera-control advantage, so the supervision-free protocol of Section 4.4 isolates the formulation. At 10K steps (full test split; the 1K runs use the 50-clip subset), the unsupervised head's radial self-consistency error (SSI-L1 against depth re-estimated from its own generated video) is 0.133 versus 0.087 with supervision (0.129 vs. 0.117 at 1K), and the CamMC ordering reverses, with the unsupervised model falling below UCPE (Table 4B). Few monocular models provide metric radial distance under the UCM, which motivated UniK3D; UniDepthV2 supervision [31] on the pinhole subset (1K steps, pose stride 4) is comparable to UniK3D (CamMC 15.41 vs. 15.23), suggesting that the results are not specific to one teacher.

Noisy dynamic regions. On 50 PanShot clips with moving-object masks [32] (34 with valid support; one middle-window layer at one denoising step; Appendix J), dynamic tokens disagree more

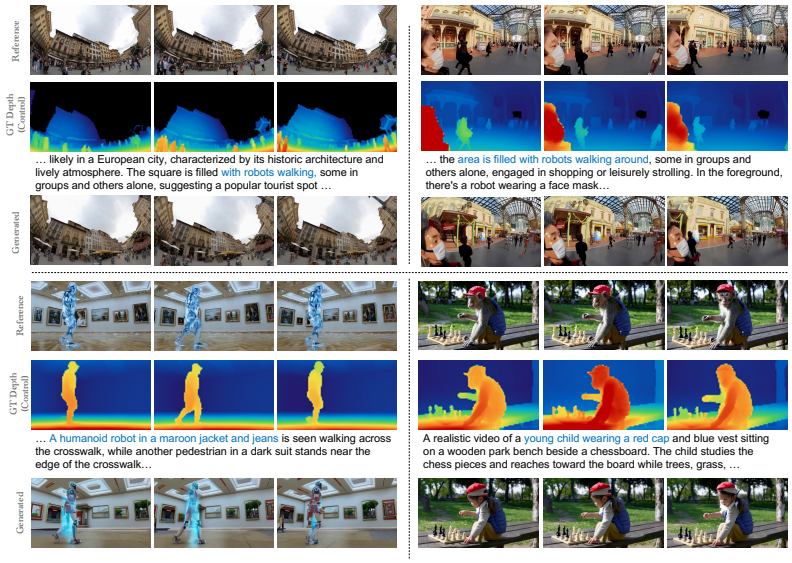


Figure 4: **Qualitative external-radial-map control.** CRePE-Mix can consume externally supplied radial-distance maps through the same positional-encoding pathway used during camera-conditioned generation. We show scene-geometry-conditioned generation and source-video motion transfer qualitative results on PanShot (top) and in-the-wild AI-generated videos (bottom). "GT Depth (Control)" marks supplied pseudo radial maps, not sensor ground truth. The motion-transfer examples (bottom) start denoising from the source video's latents noised to 75% (SDEdit-style [38]) rather than from pure noise, so source appearance can also carry over; the quantitative adherence in Table 5 is measured from pure noise.

with the pseudo target (log-MAE 0.294 vs. 0.208) but receive $1.51\times$ wider predicted uncertainty than static tokens. Across frames, CRePE's median log-radial statistic changes by 0.0156 versus 0.0375 for UniK3D and 0.0084 for the SLAM-consistent ViPE reference [28], and CRePE tracks ViPE's scale changes more closely (drift disagreement 0.020 vs. 0.034; favoring CRePE on 82% of supported clips). CRePE thus filters rather than copies teacher jitter; its latent is not metric moving-object depth.

Generalization beyond PanShot. Without tuning, CRePE improves every pose metric over UCPE on 32 real 180° fisheye KITTI-360 scenes [33] and on 100 RealCam-Vid [34] scenes drawn from RealEstate10K [35], DL3DV [36], and MiraData [37] (Table 4C), reducing CamMC by 24% and 22%, respectively, and also improves VBench overall consistency. The no-supervision model already transfers on KITTI-360, while supervision provides the margin on diverse pinhole videos.

4.6 External Radial-Map Conditioning and Motion Transfer

CRePE exposes the ray-position variable through the positional-encoding pathway. Although our primary benchmark is camera-conditioned generation, this interface also allows intervention: at inference, CRePE-Mix can replace the predicted radial-distance tensor with an external radial map after the same validity filtering and scale normalization used for pseudo targets. We show two qualitative uses in Figure 4. First, the same prompt and camera trajectory can be generated with different radial maps, changing scene layout through the PE pathway. Second, a source video can provide a sequence of radial maps, allowing motion and scene-geometry transfer while using a new text prompt and target camera trajectory. Invalid or missing external radial values do not overwrite the predicted tensor; those rays continue to use the model prediction according to Eq. (6).

Beyond these qualitative examples, we quantify adherence on 33 DAVIS-2016 sequences [39]: all methods receive identical UniK3D source maps, and geometry re-estimated from each generated video is compared at pixel and 16×16-pooled token resolution. CRePE-Mix is best on all token-level metrics and on seven of eight pixel/token comparisons (Table 5; within 0.0008 of VACE on pixel SILog), with about 30% lower Scale Jitter than the strongest baseline, despite modifying only positional coefficients. On 48 valid non-pinhole PanShot scenes, it also leads VACE and OmniVDiff in AbsRel/δ_1/Scale Jitter (0.550/0.381/0.088 vs. 0.833/0.371/0.095 and 0.929/0.097/0.233). In Figure 5,



Figure 5: **DAVIS radial-map control.** Columns: source video, UniK3D control, CRePE-Mix, VACE, OmniVDiff; rows: frames 1/41/81. Black control regions are invalid far-field rays, where CRePE-Mix uses its own prediction.

Table 5: **DAVIS radial-map adherence** (33 sequences). Cells: pixel (top) / token (bottom); best in bold. δ_1 uses mean-scale alignment; Jitter: median frame-to-frame change of the fitted depth scale; UDv2: UniDepthV2 inputs.

Method	SILog ↓	AbsRel ↓	δ_1 ↑	Jitter ↓
	0.3952	0.4018	0.5465	0.2739
ControlVideo [11]	0.3842	0.3873	0.5486	0.2751
	0.4942	0.6305	0.4437	0.4013
Ctrl-Adapter [12]	0.4854	0.6150	0.4476	0.4023
	0.5942	0.8540	0.3140	0.5087
Make-Your-Video [13]	0.5892	0.8426	0.3182	0.5089
	0.3764	0.3902	0.5096	0.3184
OmniVDiff [15]	0.3701	0.3842	0.5133	0.3181
	0.2982	0.3016	0.6220	0.2518
VACE [14]	0.2877	0.2929	0.6259	0.2521
	0.3218	0.3307	0.6348	0.2174
CRePE-Mix (UDv2)	0.3059	0.3161	0.6386	0.2164
	0.2990	**0.2719**	**0.6400**	**0.1781**
CRePE-Mix	**0.2834**	**0.2578**	**0.6439**	**0.1783**

CRePE-Mix reproduces the source layout and motion, whereas VACE adds structure absent from the control map and OmniVDiff drifts in appearance.

4.7 Limitations

CRePE improves geometry-aware camera conditioning, but does not solve all camera-control challenges. Camera recovery from generated videos is itself noisy, and radial adherence is re-estimated with the same UniK3D model that supplies pseudo labels, so pose and geometry metrics should be interpreted together with qualitative examples; main-table values are single-run point estimates. The radial-distance supervision depends on the quality and scale normalization of monocular geometry predictions, which may introduce errors when the estimated geometry is unreliable or poorly calibrated, as on moving objects. In addition, CRePE currently focuses on central-camera models represented by UCM, and extending the formulation to more general non-central or rolling-shutter camera models remains future work. Finally, external-radial-map baselines use different backbones, so we do not claim state-of-the-art depth conditioning; a motion-transfer benchmark is future work.

5 Conclusion

We presented CRePE, a curved-ray expectation positional encoding for unified-camera-controlled video generation. CRePE represents each token as a depth-aware positional distribution along its source ray and integrates the expected RoPE phasor along the UCM-projected curved trajectory, extending ray-only camera conditioning with token-wise scene-geometry awareness while remaining compatible with non-pinhole lenses. A Geometric Attention Adapter realizes this design on a frozen video DiT, with pseudo radial-distance supervision from a monocular geometry foundation model stabilizing the ray-position channel and probe-selected middle-layer placement keeping the adapter compact. In experiments, CRePE improves camera control over ReCamMaster and UCPE across pinhole and non-pinhole lenses with competitive video quality, and outperforms a RayRoPE-style endpoint approximation in controlled positional-encoding ablations, isolating the benefit of projected-path integration over endpoint-based approximations.

Beyond camera-only conditioning, the same positional-encoding pathway, trained with Radial Mix-Forcing, exposes a controllable interface for external scene geometry. This unifies camera and scene-geometry control under a single adapter and enables scene-geometry-conditioned generation with quantitatively verified map adherence and source-video motion transfer with new prompts and target trajectories. We hope CRePE serves as a useful building block for geometry-aware, camera-controllable video generation under diverse lens configurations.

Acknowledgments and Disclosure of Funding

This work was supported by the National Research Foundation of Korea under Grant RS-2024-00336454, the Advanced GPU Utilization Support Program funded by the Government of the Republic of Korea (Ministry of Science and ICT) (02-26-01-0404), the AI Computing Infrastructure Enhancement (GPU Rental Support) User Support Program funded by the Ministry of Science and ICT (MSIT) Republic of Korea (RQT-25-120217), and Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT (MSIT, Korea) & Gwangju Metropolitan City.

References

[1] 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 *IEEE/CVF International Conference on Computer Vision*, pages 14834–14844, 2025.

[2] Yu Wu, Minsik Jeon, Jen-Hao Rick Chang, Oncel Tuzel, and Shubham Tulsiani. RayRoPE: Projective ray positional encoding for multi-view attention. In *European Conference on Computer Vision*, 2026.

[3] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 38027–38037, 2026.

[4] William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL `https://arxiv.org/abs/2212.09748`.

[5] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[6] Botao Ye, Boqi Chen, Haofei Xu, Daniel Barath, and Marc Pollefeys. YoNoSplat: You only need one model for feedforward 3D Gaussian splatting. In *International Conference on Learning Representations*, pages 39852–39871, 2026.

[7] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for text-to-video generation, 2025. URL `https://arxiv.org/abs/2404.02101`.

[8] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. MotionCtrl: A unified and flexible motion controller for video generation, 2024. URL `https://arxiv.org/abs/2312.03641`.

[9] 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 *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, SIGGRAPH '24, pages 1–12. ACM, July 2024. doi: 10.1145/3641519.3657481. URL `http://dx.doi.org/10.1145/3641519.3657481`.

[10] Jason Y. Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion, 2024. URL `https://arxiv.org/abs/2402.14817`.

[11] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. ControlVideo: Training-free controllable text-to-video generation. In *International Conference on Learning Representations*, pages 54441–54461, 2024.

[12] Han Lin, Jaemin Cho, Abhay Zala, and Mohit Bansal. Ctrl-Adapter: An efficient and versatile framework for adapting diverse controls to any diffusion model. In *International Conference on Learning Representations*, pages 101103–101147, 2025.

[13] Jinbo Xing, Menghan Xia, Yuxin Liu, Yuechen Zhang, Yong Zhang, Yingqing He, Hanyuan Liu, Haoxin Chen, Xiaodong Cun, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Make-Your-Video: Customized video generation using textual and structural guidance. *IEEE Transactions on Visualization and Computer Graphics*, 31(2):1526–1541, 2025. doi: 10.1109/TVCG.2024.3365804.

[14] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. VACE: All-in-one video creation and editing. In *IEEE/CVF International Conference on Computer Vision*, pages 17191–17202, 2025.

[15] Dianbing Xi, Jiepeng Wang, Yuanzhi Liang, Xi Qiu, Yuchi Huo, Rui Wang, Chi Zhang, and Xuelong Li. OmniVDiff: Omni controllable video diffusion for generation and understanding. In *AAAI Conference on Artificial Intelligence*, volume 40, pages 10915–10923, 2026. doi: 10.1609/aaai.v40i13.38068.

[16] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

[17] Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J. Davison. EscherNet: A generative model for scalable view synthesis, 2024. URL https://arxiv.org/abs/2402.03908.

[18] Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. GTA: A geometry-aware attention mechanism for multi-view transformers, 2024. URL https://arxiv.org/abs/2310.10375.

[19] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding, 2025. URL https://arxiv.org/abs/2507.10496.

[20] Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering, 2022. URL https://arxiv.org/abs/2106.02634.

[21] Benjamin Attal, Jia-Bin Huang, Michael Zollhoefer, Johannes Kopf, and Changil Kim. Learning neural light fields with ray-space embedding networks, 2022. URL https://arxiv.org/abs/2112.01523.

[22] Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations, 2022. URL https://arxiv.org/abs/2111.13152.

[23] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. CAT3D: Create anything in 3D with multi-view diffusion models, 2024. URL https://arxiv.org/abs/2405.10314.

[24] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy, 2024. URL https://arxiv.org/abs/2312.14132.

[25] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R, 2024. URL https://arxiv.org/abs/2406.09756.

[26] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer, 2025. URL https://arxiv.org/abs/2503.11651.

[27] Luigi Piccinelli, Christos Sakaridis, Mattia Segu, Yung-Hsu Yang, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniK3D: Universal camera monocular 3D estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1028–1039, 2025.

[28] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, et al. ViPE: Video pose engine for 3D geometric perception. *arXiv preprint arXiv:2508.10934*, 2025.

[29] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

[30] Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, and Marc Pollefeys. GeoCalib: Learning single-image calibration with geometric optimization. In *European Conference on Computer Vision*, pages 1–20. Springer, 2024. doi: 10.1007/978-3-031-73661-2_1.

[31] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniDepthV2: Universal monocular metric depth estimation made simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):2354–2367, 2026. doi: 10.1109/TPAMI.2025.3628473.

[32] Nan Huang, Wenzhao Zheng, Chenfeng Xu, Kurt Keutzer, Shanghang Zhang, Angjoo Kanazawa, and Qianqian Wang. Segment any motion in videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3406–3416, 2025.

[33] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3292–3310, 2023. doi: 10.1109/TPAMI.2022.3179507.

[34] Guangcong Zheng, Teng Li, Xianpan Zhou, and Xi Li. RealCam-Vid: High-resolution video dataset with dynamic scenes and metric-scale camera movements. *arXiv preprint arXiv:2504.08212*, 2025.

[35] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM TOG*, 37(4):1–12, 2018.

[36] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024.

[37] Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. MiraData: A large-scale video dataset with long durations and structured captions. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, pages 48955–48970, 2024.

[38] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

[39] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016.

A Standard RoPE and UCM Details

Rotary positional encoding (RoPE) [16] encodes a scalar position x as a block-diagonal rotation

$$\rho_D(x) = \bigoplus_{f=1}^{D/2} \rho_2(\omega_f x), \qquad \rho_2(\omega x) = \begin{bmatrix} \cos(\omega x) & -\sin(\omega x) \\ \sin(\omega x) & \cos(\omega x) \end{bmatrix}. \tag{7}$$

For multi-dimensional coordinates, we allocate separate channel groups to each coordinate.

The Unified Camera Model uses distortion parameter ξ and projection

$$\Pi_\xi^{\mathrm{UCM}}(\mathbf{X}) = \left(f_x \frac{X}{Z + \xi\|\mathbf{X}\|} + c_x,\ f_y \frac{Y}{Z + \xi\|\mathbf{X}\|} + c_y \right), \tag{8}$$

and inverse ray map

$$\Phi_\xi^{\mathrm{UCM}}(x, y) = \mathrm{normalize}\left([\gamma x, \gamma y, \gamma - \xi]^\top \right), \quad \gamma = \frac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{1 + x^2 + y^2}. \tag{9}$$

For $\xi \neq 0$, the projection of a source-ray interval into a query view is generally curved, motivating CRePE's piecewise phasor integration.

B CRePE Implementation Details

Backbone and training. Unless otherwise specified, the following settings describe the default 10K-step model. The backbone is Wan2.1-T2V-1.3B [5] with 30 transformer blocks, kept frozen. We train the adapter on 81-frame clips at 480×832 resolution with global batch size 8, AdamW, learning rate 10^{-4}, gradient checkpointing, and 10K optimization steps. All experiments run on a single NVIDIA B200 GPU; a default 10K-step training run takes approximately 3–7 days of wall-clock time, and the reduced-budget ablations (1K steps) use the same hardware. CRePE is applied to blocks 11–20, numbering blocks 1–30 as in Figure 6; other blocks retain UCPE-style relative-ray modulation. The attention compression factor is 8, following [3]. Figure 6 illustrates the overall Geometric Attention Adapter architecture.

Query–source frame pairs. Training clips are sampled purely by temporal position, not by relative camera motion: each example is a fixed-length clip of contiguous frames at the native frame rate. Attention operates densely over all query–source frame pairs, so every temporal offset up to the full clip span is present. Relative motion therefore ranges from near-degenerate adjacent pairs (median rotation 0.91°) to full-span pairs with substantial displacement (median 16.4°), the regime in which projected-path integration is most active.

Geometry head. The shared geometry head g_ϕ is implemented as an MLP, LayerNorm $\to$ Linear $\to$ SiLU $\to$ Linear, with hidden width $\max(16, D/4)$. Radial-distance supervision is applied to its predictions at a single selected layer, training the same geometry head used across all CRePE-modulated layers. The final layer is initialized with zero weights and bias $[0, 3]$, giving $\mu = 0$ and a broad initial interval $\sigma = 3$. The predicted log radial-distance interval is bounded to $[-3, 3]$, corresponding to normalized radial distances in approximately $[0.05, 20]$. This bound limits the predicted CRePE interval on the normalized scale. It is separate from pseudo-label validity filtering: raw metric pseudo radial-distance targets above $20\,\mathrm{m}$ are marked invalid rather than clipped to the upper bound.

Multi-ray patch encoding. Each token uses $A = 3$ offset rays at the patch corners, with offsets $(0, 0)$, $(1, 0)$, and $(0, 1)$ in patch units. For offset a, the source ray is $\boldsymbol{r}_{s,p,a} = \Phi_{\xi_s}^{\mathrm{UCM}}(u_{p,a}, v_{p,a})$. We compute CRePE coefficients independently for the offsets and allocate disjoint RoPE channel groups to them.

Query-view coordinate. Given lifted point $\mathbf{X}_q^{(k)} = (X, Y, Z)^\top$ in the query camera, let $\beta_q = Z + \xi_q\|\mathbf{X}_q^{(k)}\|$. We define

$$\bar{u}^{(k)} = \frac{f_x}{W}\frac{X}{\beta_q}, \qquad \bar{v}^{(k)} = \frac{f_y}{H}\frac{Y}{\beta_q}, \tag{10}$$

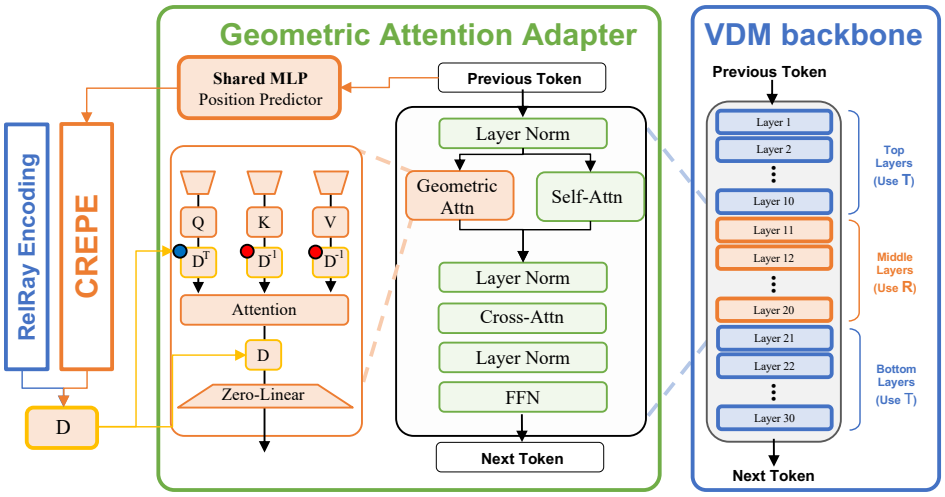


Figure 6: **Architecture of the Geometric Attention Adapter.** Our adapter follows the UCPE-style camera-conditioning structure, but augments the attention pathway with a CRePE-based geometric attention branch. The "Shared MLP Position Predictor" is the geometry head g_ϕ. Following the layer-placement ablation, CRePE is applied to the middle 10 transformer blocks, where recoverable radial-distance information is strongest. The remaining early and late blocks retain the RelRay encoding used in UCPE. This hybrid design preserves the stable ray-only camera signal in the outer blocks while injecting depth-aware curved-ray positional modulation into the middle blocks.

and use the bounded projected coordinate

$$\mathbf{u}^{(k)} = \left(\frac{[\bar{u}^{(k)}, \bar{v}^{(k)}, 1]^\top}{\|[\bar{u}^{(k)}, \bar{v}^{(k)}, 1]\|} \right)_{1:2}. \tag{11}$$

The final CRePE coordinate is $\tilde{\mathbf{x}}^{(k)} = (\mathbf{u}^{(k)}, \|\mathbf{X}_q^{(k)}\|)$. In addition, as in RayRoPE, each key token encodes the source camera center expressed in the query camera frame (3 coordinates); this term does not depend on the radial distribution and uses a single rotary phase, and for query tokens it is the origin. A token therefore carries $3 + A \times (2 + 1) = 12$ coordinates. Each coordinate uses 8 log-spaced RoPE frequencies with periods from P down to $P/3^7$, where $P = 8$ for the bounded image coordinates, $P = 80$ for the radial range (clamped to $[-10, 10]$), and $P = 4$ for the camera center, so the 12 coordinates fill the 192-dimensional geometric-attention head.

Numerical stability. When $\theta^{(k+1)} - \theta^{(k)}$ is small in Eq. (3), we use the limiting value of the segment integral. Invalid projected points and masked pseudo labels are excluded from the auxiliary radial-distance loss.

C Layer Placement Ablation

Probe protocol. For each frozen Wan2.1 block, we train an independent lightweight probe with the same input features used by the adapter to predict the valid, normalized UniK3D radial-distance targets on the token grid. The backbone and video diffusion objective are not updated during probing; the probe is optimized only with the radial-distance supervision loss and evaluated on held-out PanShot clips. Lower probing error indicates that the layer exposes recoverable ray-position information to a small adapter head, not that the final generator has been trained with CRePE at that layer.

Placement selection. The probe curve is used as a selection guide rather than as the sole criterion. We evaluate contiguous CRePE windows on the first 10, middle 10, last 10, and all 30 DiT blocks under the reduced-budget protocol of the breakpoint ablation (1K steps, $\lambda_{\text{rad}} = 0$, pose stride 4, 50 clips). Middle-layer placement provides the best overall rank and the strongest pose/lens/orientation tradeoff, while retaining UCPE-style relative-ray modulation in the remaining blocks. This is why the default model applies CRePE to blocks 11–20 and uses UCPE modulation elsewhere.

Figure 7 reports the layer-wise radial-distance probing results. Figure 8 visualizes decoded radial-distance maps across layers, Figure 9 shows their evolution during denoising, and Table 6 reports the layer-placement ablation.

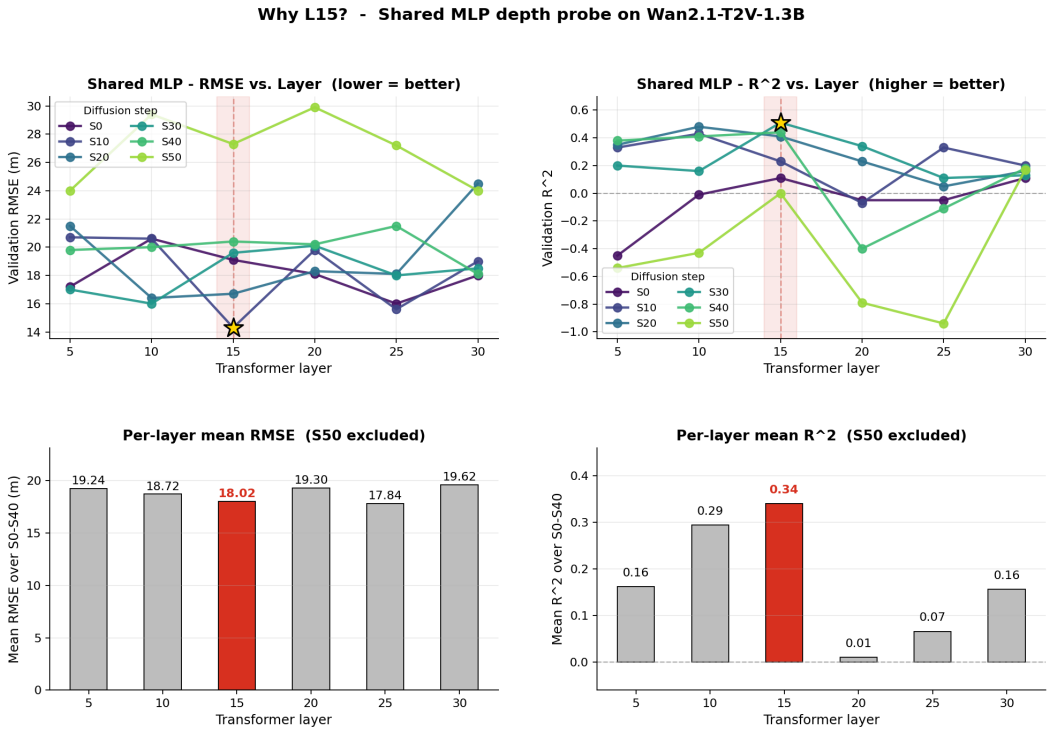


Figure 7: **Layer-wise probing of radial-distance information.** Held-out radial-distance probing error per Wan2.1 layer. The shaded middle-layer window is used for CRePE.

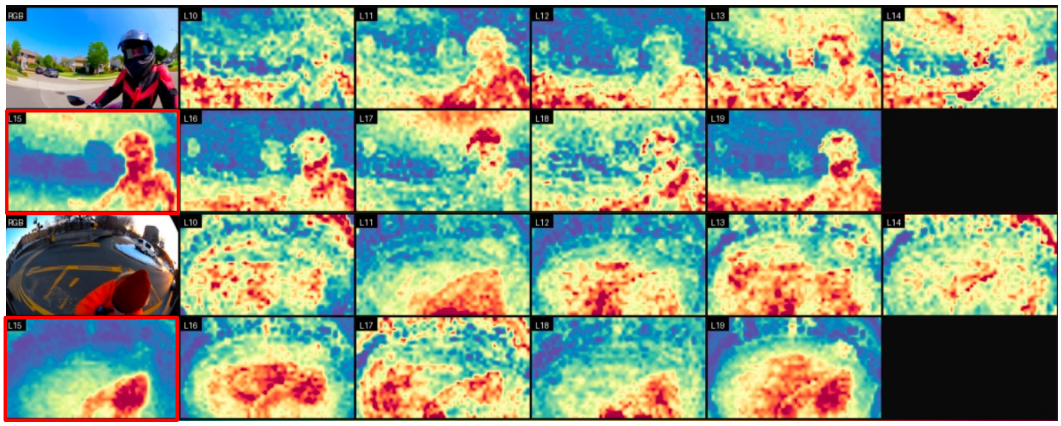


Figure 8: **Layer-wise visualization of predicted radial-distance maps.** We visualize radial-distance maps decoded from different Wan2.1 layers using the shared geometry head. The highlighted map, from the layer used as the target-supervised layer in CRePE, shows the sharpest and most coherent scene structure. This supports our choice of applying radial-distance supervision to the middle-layer representation selected by the probing analysis. The maps visualize the token-resolution expected radial distance $\exp(\mu)$ that modulates attention; they are not depth-decoder outputs and should not be read as metric depth estimates.

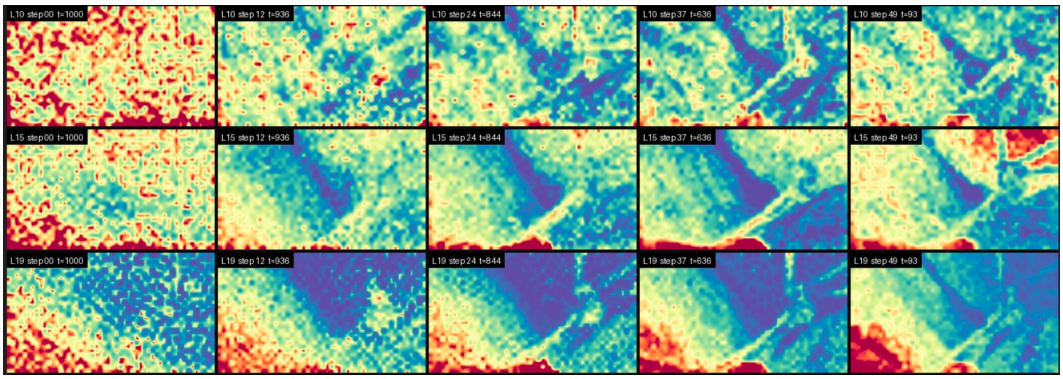


Figure 9: **Evolution of predicted radial-distance maps during denoising.** We visualize CRePE's internal radial-distance maps at multiple denoising steps. Denoising progresses from left to right. The maps evolve from coarse responses to structured scene layouts, suggesting that the ray-position pathway progressively organizes scene geometry during generation. As in Figure 8, these are token-level latent radial distributions, not metric depth predictions.

Table 6: **CRePE layer placement** (reduced-budget protocol: 1K steps, $\lambda_{\text{rad}} = 0$, pose stride 4, 50-clip ablation subset; the Middle10 row is the $K = 5$ run of the breakpoint ablation). Applying CRePE to the middle 10 blocks provides the best overall rank and the strongest pose/lens/orientation tradeoff.

Setting	Avg. Rank	Pose	Text/Image		Video		Lens		Orientation	
		Rot. Err ↓	CS-I ↑	CS-T ↑	FID ↓	FVD ↓	k_1 ↓	k_2 ↓	Pitch ↓	Up ↓
All	3.06	4.877	99.27	25.70	136.8	1457	0.1610	0.09002	6.874	6.918
First10	2.12	5.142	99.16	**25.78**	133.9	**1438**	0.1679	0.1113	6.732	6.902
Middle10	**2.00**	**4.649**	**99.30**	25.76	136.6	1491	**0.1417**	**0.08034**	**6.612**	**6.712**
Last10	2.82	4.843	99.21	25.68	**130.3**	1506	0.1574	0.08684	6.982	6.921

D Overall Camera-Generation Metrics

Table 7 reports the merged pinhole and non-pinhole results across pose, lens, orientation, text/image, video-quality, and VBench metrics.

E Radial-Distance Supervision Details

Pseudo radial-distance maps are generated by UniK3D [27]. We apply validity filtering before normalization. A raw metric pseudo target $\hat{r}^{\text{m}}$ is invalid if UniK3D fails, is marked unreliable by our validity filters, is non-finite or non-positive, falls outside the supported distance range, or exceeds $20\,\text{m}$. These far-field targets are not clipped to $20\,\text{m}$; they are excluded from the loss and, under Radial MixForcing, fall back to the model prediction for that ray. We divide only valid metric maps by a per-clip near-distance statistic to obtain normalized targets, pool them to the token grid, and mask invalid regions. From (μ, σ), the normalized prediction is $\bar{r} = \exp(\mu)$. The uncertainty scale s is derived from the variance of a uniform distribution over the normalized interval $[\exp(\mu - |\sigma|), \exp(\mu + |\sigma|)]$; this interval-width surrogate is separate from the log-uniform measure used for the phasor integration:

$$v_{\text{int}} = \frac{(\exp(\mu + |\sigma|) - \exp(\mu - |\sigma|))^2}{12}. \tag{12}$$

We use $s = \sqrt{\max(v_{\text{int}}, 10^{-6})}$ with a ceiling of 10, $\alpha = 1.0$, and $\lambda_{\text{rad}} = 10^{-3}$. Supervision is gated off for the noisiest 3% of diffusion timesteps.

F Additional Radial MixForcing Details

Section 3.4 gives the main Radial MixForcing algorithm. Here we keep only the implementation choices that are too detailed for the main text.

Table 7: **Overall camera-generation comparison.** We merge pinhole and non-pinhole subsets and report overall metrics for ReCamMaster, UCPE, and CRePE. Panel A reports camera-control, lens-distortion, and orientation metrics; Panel B reports text/image alignment and video-quality metrics; Panel C reports VBench quality and consistency metrics.

Panel A: Pose, lens distortion, and orientation

Method	Pose			Lens		Orientation				
	CamMC ↓	Rot. ↓	Trans. ↓	k_1 ↓	k_2 ↓	Pitch ↓	Roll ↓	Gravity ↓	Lat. ↓	Up ↓
ReCamMaster	55.42	16.99	45.98	0.184	0.153	8.08	6.30	11.13	8.58	7.40
UCPE	21.87	6.66	17.54	0.181	0.153	4.84	4.47	7.09	6.13	6.09
CRePE	**18.15**	**6.33**	**13.78**	**0.156**	**0.115**	**4.57**	**3.61**	**6.24**	**6.02**	**5.88**

Panel B: Text/image alignment and video quality

Method	Text/Image		Video				
	CS-I ↑	CS-T ↑	FID ↓	FVD ↓	FVD-C ↓	IS ↑	IS Std.
ReCamMaster	98.680	24.855	61.15	551.75	504.73	10.04	0.288
UCPE	98.745	25.060	58.71	**463.47**	**440.40**	**11.48**	0.230
CRePE	**98.772**	**25.096**	**58.08**	465.33	444.92	11.43	0.237

Panel C: VBench metrics

Method	VBench				
	Aesthetic ↑	BG Cons. ↑	Img. Qual. ↑	Overall Cons. ↑	Subject Cons. ↑
ReCamMaster	0.5599	0.9473	74.79	0.2142	**0.9467**
UCPE	0.5617	0.9516	74.89	0.2157	0.9427
CRePE	**0.5634**	**0.9524**	**75.48**	**0.2163**	0.9425

Teacher interval width. When a valid pseudo radial target is substituted, we use $\sigma_T = 0.1$. Invalid pseudo targets never create a teacher interval: if the pseudo distance is unavailable, marked unreliable by the validity filters, non-finite or non-positive, outside the supported range, or above $20\,\mathrm{m}$ in raw metric units, the effective CRePE geometry for that ray remains $(\mu_{\mathrm{pred}}, \sigma_{\mathrm{pred}})$.

Schedule. We use a piecewise-linear schedule that starts at full substitution probability, decays between steps 1000 and 7000, and then holds a small floor. For block-frame sampling, the floor is 0.1; for video-level sampling, the floor is 0.5. The mask is sampled once per forward pass and shared across all CRePE layers.

External-radial-map inference. For qualitative scene-geometry conditioning and source-video motion transfer, we intentionally provide an external radial map at inference and route it through the same CRePE pathway. The same validity and scale-normalization rule is used as in training: valid external radial values can overwrite the predicted radial tensor after normalization, while invalid or missing values use the model prediction for the corresponding ray. This changes only the positional coefficients, not the video backbone. For the source-video motion-transfer examples in Figure 4 (bottom), denoising starts from the source video encoded by the VAE and noised to strength 0.75 (SDEdit-style [38], 50 steps) instead of pure noise, so part of the source appearance carries over in addition to the supplied geometry; the DAVIS evaluation in Table 5 starts from pure noise.

G Dataset and Evaluation Details

PanShot [3] contains panning videos with camera trajectories and UCM lens parameters. The train/test split prevents scene overlap. The dataset spans pinhole, wide-angle, and fisheye settings with horizontal FoV and UCM distortion. Camera controllability is measured by recovering camera motion from generated videos and comparing it to the requested trajectory. The main camera-generation comparison uses all 340 clips in the PanShot test split; we do *not* remove the high-motion tail of the distribution (e.g., the top 20% by camera-motion magnitude [3]), since large-trajectory clips are the regime in which curved-ray modeling is expected to matter most. Reduced-budget ablations use a fixed 50-clip subset of the test split (13 pinhole and 37 non-pinhole clips); the external-control subsets are described in Appendix K.

H Number of Breakpoints

Table 8 retains the expanded metric suite for $K \in \{2, 5, 7\}$. These variants follow the 1K-step, diffusion-only, stride-4 protocol in Appendix I; they are distinct from the default 10K-step supervised model. Table 3 additionally includes $K = 1$ and $K = 3$ and reports ranks among all five settings.

Table 8: **Full sampling-point ablation for CRePE.** We report all available non-zero metrics for CRePE with different numbers of radial-distance breakpoints ($K = 2$ is the endpoint approximation). Panels A1–A3 summarize pose, text/image alignment, and video-quality metrics, while Panel B reports lens and orientation metrics. Avg. Rank is computed over all displayed metrics except IS Std., among these three settings (Table 3 ranks five settings). CRePE with $K = 5$ achieves the best overall balance.

Panel A1: Average rank and pose

Setting	Avg. Rank ↓	CamMC ↓	Rot. ↓	Trans. ↓
CRePE, $K = 2$	2.55	14.75	5.648	11.09
CRePE, $K = 5$	**1.35**	**13.45**	**4.649**	**10.55**
CRePE, $K = 7$	2.10	14.23	4.702	11.21

Panel A2: Text/image alignment

Setting	CS-I ↑	CS-T ↑	LPIPS ↓	PSNR ↑	SSIM ↑
CRePE, $K = 2$	99.300	25.750	0.720	8.848	0.262
CRePE, $K = 5$	**99.301**	**25.762**	**0.719**	**8.886**	**0.264**
CRePE, $K = 7$	99.265	25.696	0.720	8.868	0.263

Panel A3: Video quality

Setting	FID ↓	FVD ↓	FVD-C ↓	IS ↑	IS Std.
CRePE, $K = 2$	**135.34**	1563	1589	9.55	0.351
CRePE, $K = 5$	136.61	**1491**	**1465**	**10.43**	0.570
CRePE, $K = 7$	135.46	1550	1502	9.98	0.308

Panel B: Lens distortion and orientation

Setting	Lens		Orientation				
	k_1 ↓	k_2 ↓	Pitch ↓	Roll ↓	Gravity ↓	Lat. ↓	Up ↓
CRePE, $K = 2$	**0.1212**	0.0789	7.333	5.818	10.458	8.684	6.949
CRePE, $K = 5$	0.1417	0.0803	**6.612**	5.598	**9.688**	**8.455**	**6.712**
CRePE, $K = 7$	0.1396	**0.0779**	6.756	**5.462**	9.801	8.588	6.757

I Protocols and Stratified Results for the Curved-Path Analysis

Supervision-free breakpoint protocol. All breakpoint variants in the main-paper ablation ($K \in \{1, 2, 3, 5, 7\}$) are trained for an identical reduced budget of 1K steps with the diffusion objective only ($\lambda_{\text{rad}} = 0$), the same parameter count, and the same data. UniK3D is used neither to train these variants nor at inference; layer probing only selects the placement window. All variants predict the same two interval parameters (μ, σ); for $K \geq 2$, interior breakpoints are deterministic evaluations of this same interval, not additional learned geometry. Pose metrics for these ablations use frame stride 4 and are therefore not comparable in absolute terms to the stride-1 numbers of the main comparison tables.

Camera-model stratification. To examine the lens-dependent component of the curved-path gain, we stratify the supervision-free evaluation into pinhole ($\xi = 0$, 13 clips) and non-pinhole ($\xi \neq 0$, 37 clips) subsets. Table 9 reports the raw errors together with the improvement of each curvature-capable setting relative to the endpoint approximation ($K = 2$) within each subset. At $K = 3$, translation gains are nearly identical across the two subsets (13.2% vs. 13.4%), suggesting a shared, lens-independent benefit of non-affine integration; CamMC and rotation gains are larger and more persistent on non-pinhole clips, consistent with an additional contribution from UCM-induced transverse curvature. The gains saturate rather than grow monotonically with K.

Table 9: **Pinhole/non-pinhole stratification of the supervision-free breakpoint ablation.** Percentages are error changes relative to $K = 2$ within each subset (negative is better); all metrics are lower-is-better.

Metric	Camera model	K=2	K=3	K=5
CamMC	Pinhole	14.7457	13.1519 (−10.8%)	14.7137 (−0.2%)
	Non-pinhole	14.7555	12.0174 (−18.6%)	13.0098 (−11.8%)
RotErr	Pinhole	5.2380	5.0362 (−3.9%)	4.9312 (−5.9%)
	Non-pinhole	5.7914	4.2134 (−27.2%)	4.5505 (−21.4%)
TransErr	Pinhole	11.3287	9.8322 (−13.2%)	11.7906 (+4.1%)
	Non-pinhole	11.0002	9.5238 (−13.4%)	10.1079 (−8.1%)

Why this is not a sample-count effect. For normalized log-radius $t \in [0, 1]$ and phase samples $\theta_k = \theta(t_k)$, the implemented coefficient of Eq. (3), written $\bar{\rho}_K$ for K breakpoints, is the average of exact per-segment integrals,

$$\bar{\rho}_K = \frac{1}{K-1} \sum_{k=1}^{K-1} \frac{e^{i\theta_{k+1}} - e^{i\theta_k}}{i(\theta_{k+1} - \theta_k)}. \tag{13}$$

If $\theta(t) = at + b$ is affine, every segment has the same phase increment, so the sum telescopes to $(e^{i\theta(1)} - e^{i\theta(0)})/[i(\theta(1) - \theta(0))]$, independent of K; extra subdivision helps only when the deployed phase is non-affine. Exponentiating log radius, camera translation, range encoding, and bounded coordinates create a shared non-affinity even for pinhole cameras, and the UCM denominator adds lens-dependent bending. Consistently, in a stratified evaluation (13 pinhole and 37 non-pinhole clips; Table 9), $K = 3$ reduces CamMC over $K = 2$ by 10.8% on pinhole and 18.6% on non-pinhole clips with nearly identical translation gains (13.2% vs. 13.4%), and at $K = 5$ gains persist on non-pinhole clips (CamMC -11.8%) but are small or mixed on pinhole clips. This indicates a shared viewpoint/depth benefit plus a non-pinhole component, rather than a purely UCM-curvature effect.

J Supervision and Robustness Details

Supervision-duration study. We train the $K = 5$ setting for 1K and 10K steps, with and without supervision. SSI-L1 is the scale- and shift-invariant L1 error between the geometry head's prediction and depth re-estimated from its own generated video (lower means more self-consistent). Pose metrics use stride 1, as in the main comparison tables. The 1K-step runs are evaluated on the 50-clip ablation subset and the 10K-step runs on the full 340-clip test split, so we compare settings within a budget. Table 10 shows that at the 1K ablation budget supervision provides no camera-control advantage (the unsupervised point estimates are slightly ahead), so the supervision-free protocol above is a fair isolation of the formulation. Under long optimization the two settings diverge: at 10K steps the supervised head is markedly more self-consistent (SSI-L1 0.087 vs. 0.133), the camera-control ordering reverses, and the unsupervised variant falls below even depth-free UCPE (CamMC 22.19 vs. 21.87). The lightweight anchor ($\lambda_{\text{rad}} = 10^{-3}$) therefore stabilizes the radial channel under long training rather than producing the formulation-level gain.

Table 10: **Role of pseudo radial-distance supervision.** The same $K = 5$ configuration trained with and without $\mathcal{L}_{\text{rad}}$ at two budgets (1K: 50-clip ablation subset; 10K: 340-clip test split; pose stride 1).

Metric	1K no-sup	1K sup	10K no-sup	10K sup
CamMC ↓	52.02	56.27	22.19	**18.15**
RotErr ↓	18.00	18.21	8.11	**6.33**
Gravity ↓	9.69	10.04	7.57	**6.24**
SSI-L1 ↓	0.129	0.117	0.133	**0.087**

Alternate pseudo-depth teacher. Few monocular models provide metric radial prediction under UCM, which motivated our choice of UniK3D. Retraining CRePE with UniDepthV2 supervision [31] (1K steps, same λ_{rad}; pinhole subset, as UniDepthV2 lacks UCM-compatible metric radial distance under strong distortion; stride-4 pose) reaches performance comparable to the UniK3D-supervised

counterpart: CamMC 15.41 vs. 15.23, RotErr 5.72 vs. 5.89, CS-T 25.94 vs. 25.80. Because supervision has little effect at this budget (Table 10), this comparison suggests that the results are not specific to UniK3D rather than showing that the long-training anchor transfers across teachers.

Depth-Concat control. The Depth-Concat baseline receives the same pseudo radial signal under the same auxiliary loss and the full 10K-step protocol: a shared MLP predicts the pseudo radial distance, and the prediction is passed through an additional projection MLP and concatenated to the token features ($+6$M parameters) instead of entering the positional phase. Depth-Concat trains healthily and is comparable to CRePE in quality (FVD 473.0 vs. 465.3; CS-T 25.08 vs. 25.10), yet fails specifically on pose, below even depth-free UCPE (CamMC 23.21 vs. 21.87; CRePE 18.15). What matters is how geometry enters attention—as a phase varying per query with the relative camera transform, not as a query-agnostic feature.

Dynamic-region behavior. Frame-wise monocular pseudo-depth is least reliable on moving objects. On 50 PanShot clips, we evaluate the radial prediction of one middle-window layer at a single denoising step, using moving-object masks from Segment Any Motion [32] pooled to the 30×52 token grid (a token is dynamic if at least 50% of its pixels move, static if none; invalid tokens are excluded). Statistics use identical normalization and pooling, with 95% clip-level bootstrap confidence intervals (2000 resamples; 34/50 clips retain sufficient valid tokens); the metrics measure agreement with the supervision signal, not ground-truth depth. Dynamic tokens disagree more with the pseudo target (log-MAE 0.294 [0.228, 0.342] vs. 0.208 [0.194, 0.236]; δ_1-style agreement 0.381 [0.312, 0.492] vs. 0.533 [0.475, 0.560]) but receive $1.51\times$ wider predicted uncertainty (mean $|\sigma|$ 1.180 vs. 0.779), avoiding static-level confidence in unreliable labels.

Temporal stability of the radial channel. Table 11 reports the median absolute frame-to-frame change of per-frame log-radial statistics over shared valid tokens (median over frames, then clips; brackets are 95% clip-level bootstrap confidence intervals). All four paired CRePE–UniK3D differences are significant, and the SLAM-consistent ViPE reference [28] anchors every row: ViPE $<$ CRePE $<$ UniK3D, so CRePE is ordered between the pose-consistent reference and its frame-wise teacher rather than merely over-smoothed. CRePE's scale changes also track ViPE better than UniK3D's do (median-statistic drift disagreement 0.020 vs. 0.034, favoring CRePE on 82% of supported clips). The radial channel filters, rather than copies, teacher jitter.

Table 11: **Temporal stability of per-frame log-radial statistics.** Median absolute frame-to-frame change (lower is more stable); brackets are 95% bootstrap confidence intervals.

Per-frame statistic	CRePE	UniK3D	ViPE (anchor)
Median	0.0156 [0.0156, 0.0234]	0.0375 [0.0254, 0.0503]	0.0084
Q1	0.0269 [0.0220, 0.0312]	0.0398 [0.0295, 0.0550]	0.0080
Q3	0.0181 [0.0156, 0.0234]	0.0305 [0.0267, 0.0391]	0.0091
IQR	0.0234 [0.0156, 0.0244]	0.0306 [0.0248, 0.0417]	0.0114

K Zero-Shot and External-Control Evaluation Details

Zero-shot protocol and full results. Both zero-shot evaluations use the PanShot-trained models without any additional training or tuning, with pose metrics at stride 4. KITTI-360 [33] provides real 180° fisheye driving videos (32 scenes); RealCam-Vid [34] provides 100 test-split scenes from RealEstate10K, DL3DV, and MiraData9K (34/33/33). Main Table 4C reports the metrics measured on both datasets. CRePE improves every pose metric over UCPE on both, and VBench overall consistency also favors CRePE on both. On KITTI-360, GeoCalib orientation errors measured on the generated videos likewise favor the CRePE variants over UCPE: roll 1.668, gravity 2.688, and up-vector 2.654 for CRePE versus 2.076, 3.060, and 3.016 for UCPE, with pitch on par (1.783 vs. 1.777), and the no-supervision variant reaching pitch 1.478 and gravity 2.500. Even the no-supervision variant outperforms UCPE on KITTI-360 in pose—indicating that the curved-path formulation itself transfers under strong lens distortion—while supervision provides the margin on the diverse pinhole content of RealCam-Vid.

External-control protocol. Radial-map adherence is evaluated on 33 DAVIS-2016 sequences [39] (dynamic pinhole videos disjoint from training), selected by a valid-token-ratio threshold fixed before

comparing methods. All methods receive the same source control maps; geometry is re-estimated from each generated video with the same fixed UniK3D model and compared at the 480×480 pixel level and the 30×30 token level after 16×16 average pooling. Scale Jitter is the median absolute frame-to-frame change of the fitted per-frame depth-scale factor. The non-pinhole control evaluation uses the 48 of 50 non-pinhole PanShot scenes with sufficient valid depth (a set separate from the 50-clip ablation subset), with the radial map as the supplied control.

Teacher-width sweep. In Radial MixForcing, the injected radial map defines the mean of the teacher's radial distribution and σ_T its width. Training uses $\sigma_T = 0.1$; Table 12 sweeps σ_T at inference on the MixForcing-trained model. Per-metric optima spread across the range. Camera-motion error varies mildly over a $20\times$ change (CamMC spread 3.9%), while video-quality and lens-distortion metrics vary more. The range 0.05–0.3 maintains comparable camera-motion control, and raising σ_T at inference can partially relax the rigidity of tightly forced geometry. CS-T rises monotonically but only by $+0.9\%$, so the flexibility–alignment trade-off is minor.

Table 12: **Inference-time teacher-width sweep for Radial MixForcing.**

Metric	σ_T=0.05	σ_T=0.10	σ_T=0.30	σ_T=1.00
CamMC ↓	**20.39**	21.02	21.18	21.14
k_1 ↓	0.148	0.148	**0.116**	0.163
k_2 ↓	0.110	0.101	**0.070**	0.104
FVD ↓	1104	**1062**	1079	1394
CS-T ↑	25.73	25.80	25.81	**25.96**

L Broader Impacts

CRePE makes camera motion and lens geometry more controllable in video generation. Positive uses include content creation with physically plausible camera moves, synthetic data for perception systems that rely on wide-angle and fisheye sensors (e.g., driving and robotics), and more geometrically consistent video simulation. The same controllability could also make synthetic videos more convincing and thereby easier to misuse for misleading or deceptive content. CRePE is a lightweight adapter on a publicly released backbone and inherits that backbone's capabilities and limitations; deployments should follow the backbone's usage policy and use standard safeguards such as content filtering and provenance marking of generated media. CRePE's internal radial distances are token-level latent variables supervised by monocular pseudo labels and should not be used as metric depth in safety-critical applications.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 claim a UCM-compatible projected-path positional encoding, pseudo supervision acting as a training stabilizer, camera-control gains over UCPE and ReCamMaster with competitive video quality, improvement over an endpoint approximation, zero-shot transfer, and external radial-map control with quantitatively verified map adherence (Section 1); these are supported in Sections 4.3–4.6, and their scope is qualified in Section 4.7.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 4.7 discusses single-run estimates, shared-estimator bias of the proxy metrics, reliance on monocular pseudo labels, restriction to central (UCM) cameras, and backbone differences in the external-control comparison.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The only formal statement is the K-invariance of Eq. (13) for affine phase paths; its assumption (a phase affine in the normalized log radius) and its telescoping proof are given in Appendix I.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The method is fully specified in Section 3 and Appendices A–B; training settings, supervision, MixForcing schedules, datasets, metrics, strides, and evaluation subsets are given in Section 4.1 and Appendices E–K.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is publicly available at `https://github.com/jinotter3/CRePE` (project page: `https://crepe-neurips2026.github.io/`). All datasets (PanShot, KITTI-360, RealCam-Vid, DAVIS-2016) and the Wan2.1 backbone are publicly available, and the method and protocols are specified in sufficient detail for reimplementation (Section 3, Appendices B–K).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 and Appendices B, E, F, G, and I give the data splits, optimizer, learning rate, batch size, training budgets, layer placement, loss weights, and evaluation protocols.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The main comparison tables report single-run point estimates because each training and evaluation run of a video diffusion adapter is costly; this is stated in Section 4.7. The dynamic-region and temporal-stability analyses report 95% clip-level bootstrap confidence intervals (Appendix J).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments run on a single NVIDIA B200 GPU (Appendix B). A default 10K-step training run (frozen 1.3B-parameter backbone, global batch size 8, 81-frame 480×832 clips, gradient checkpointing) takes approximately 3–7 days of wall-clock time; the reduced-budget ablations use 1K steps on the same hardware. Preliminary and discarded runs increased the total compute over the reported runs by an estimated factor of 2–3$\times$.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available datasets and models under their terms of use, involves no human subjects, and discusses potential misuse in Appendix L.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix L discusses positive uses (content creation, simulation and synthetic data for wide-angle and fisheye perception) and negative uses (more convincing misleading synthetic video), together with mitigation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We release only lightweight camera-control adapter weights (the CRePE camera-control and CRePE-Mix adapters) for the publicly available Wan2.1-T2V-1.3B backbone; they add camera, lens, and radial-map control but no content-generation capability beyond that backbone, and no scraped datasets are released. The weights are distributed with a model card describing the model, its training data, and its limitations, and users must also comply with the backbone's license and usage terms. Appendix L discusses misuse risks and recommends content filtering and provenance marking for deployments.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All models, datasets, and evaluation tools used (Wan2.1, UniK3D, UniDepthV2, ViPE, GeoCalib, VBench, Segment Any Motion, PanShot, KITTI-360, RealCam-Vid, DAVIS-2016, and the compared methods) are cited, and are used for non-commercial academic research in accordance with their released licenses and terms of use (e.g., Apache-2.0 for Wan2.1 and CC BY-NC-SA 3.0 for KITTI-360).

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code and the adapter weights (CRePE camera-control and CRePE-Mix) released at `https://github.com/jinotter3/CRePE` are new assets; the code is documented in its repository README (installation and usage) and the weights in a model card (training data, settings, limitations), both released with a license.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not a component of the proposed method; LLM use was limited to writing assistance, auxiliary data processing, and result visualization, which does not affect the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.