CAMERA-CONTROLLED VIDEO GENERATION AT SCALE

Anonymous authors
Paper under double-blind review

ABSTRACT

Camera-controlled video generation has become a promising interface for cinematic content creation, but scaling up 3D-camera-pose-conditioned training remains challenging. In large video corpora, camera translation is typically not aligned to a metric scale, making global motion control ambiguous across examples. Metric-alignment approaches address this issue with a preprocessing pipeline that combines structure-from-motion and metric depth estimation to approximate a shared scale. Applying this pipeline to every training video makes metric alignment a computational bottleneck as the corpus grows, while depth-estimation errors can also affect the aligned scale. To remove this bottleneck, we propose a scale-decoupled formulation for camera conditioning that treats trajectory magnitude as an explicit control variable rather than a quantity that must be metrically calibrated in advance. We normalize translation by its total path length, retaining zero translation for stationary cameras. This preserves direction and relative frame-to-frame magnitude while discarding unreliable global scale. Our goal is scalable training without metric-scale alignment, rather than resolving monocular scale ambiguity. We then inject camera-pose $\Pi(\mathbf{R}, \hat{\mathbf{c}}/a)$ and scale $\phi(\log a)$, where a controls the overall strength of camera translation, ϕ denotes a sinusoidal-style embedding, and Π denotes Plücker embedding. The two branches are encoded separately and fed to the diffusion transformer. In addition, to avoid requiring human-annotated trajectory preferences, we introduce trajectory preference optimization (TPO), a preference-learning scheme that keeps the video fixed and constructs negative examples by changing the conditioning trajectory, thereby isolating trajectory fidelity from video quality. After supervised fine-tuning (SFT), our 2B and 5B models reduce translation error by 49.4% and 36.8%, rotation error by 38.2% and 38.9%, FVD by 40.7% and 43.3%, and FID by 15.9% and 12.0%, respectively, relative to size-matched AC3D baselines. At the same total optimization-step count, SFT followed by TPO reduces the 2B model's translation and rotation errors by 4.4% and 4.0%, respectively, relative to SFT alone, while keeping FVD, FID, and CLIP score broadly stable.

1 INTRODUCTION

Recent diffusion-based image and video generation models have achieved remarkable visual realism (Ho et al., 2020; 2022b). Foundational diffusion formulations (Ho et al., 2020; Song et al., 2020; Ho et al., 2022b), latent-space modeling (Rombach et al., 2022), transformer-based denoising architectures (Peebles & Xie, 2023), accelerated sampling (Song et al., 2021; Zhao et al., 2023b; Lu et al., 2022a;b), and large-scale video generation models (Ho et al., 2022a; He et al., 2022; Blattmann et al., 2023; Hong et al., 2022; Wan et al., 2025; Kong et al., 2024; Yang et al., 2024c) together form the modern backbone of video synthesis. As visual quality improves, controllability becomes increasingly important. Among different control signals, camera motion is central because it determines viewpoint, scene exploration, and cinematic composition. Recent works study camera-controlled video generation by separating camera and object motion (Wang et al., 2024), using Plücker embedding as a camera condition (He et al., 2024), and improving text-to-video or image-to-video camera control (Bahmani et al., 2025a; Cheng et al., 2026; Xu et al., 2024; Li et al., 2025a). However, a fundamental obstacle remains: camera translations estimated from large-scale videos are subject to arbitrary scene scale. For example, one unit in translation may mean 1 meter or

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

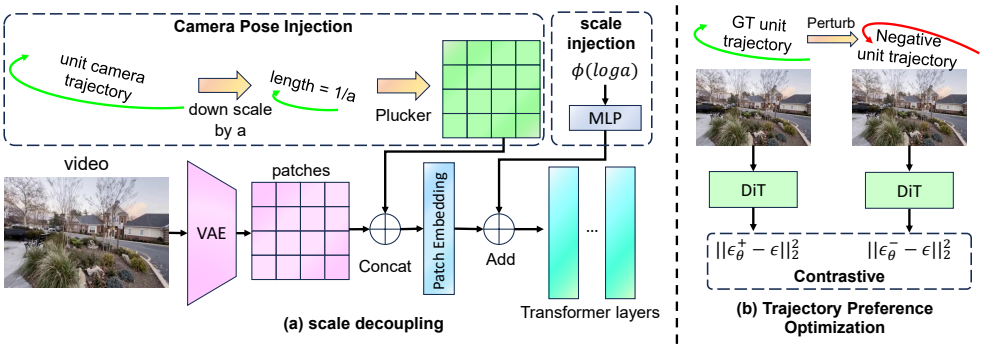


Figure 1: (a) **Scale decoupling**. We normalize nonzero translation paths to unit length and retain zero translation otherwise. During training, the unit trajectory is scaled by $1/a$ in the trajectory branch, while a is encoded as a separate magnitude condition. During inference, the trajectory branch receives the unit trajectory, and a serves as an explicit control variable for camera-motion strength. (b) **Trajectory Preference Optimization (TPO)**. We construct negative trajectories by perturbing the ground-truth unit trajectory in translation and/or rotation while keeping the video fixed. A contrastive preference loss is computed between the original and perturbed trajectory conditions for the same video.

ten meters depending on scenes, creating inconsistency between training trajectories and real-world camera motion (Li et al., 2025a). RealCam-I2V (Li et al., 2025a) addresses this issue by aligning trajectories to metric scale using COLMAP triangulation (Schonberger & Frahm, 2016) and a metric depth estimator (Yang et al., 2024b).

However, repeating this metric-alignment pipeline across a large video corpus adds substantial preprocessing computation, creating a bottleneck for scaling camera-conditioned training. Errors in estimated metric depth can also propagate into the aligned scale. This dependence on physical scale further complicates user control: unlike sketches, poses, or masks in controllable image generation (Zhang et al., 2023; Zhao et al., 2023a; Li et al., 2024; Lyu et al., 2025), distances in meters may not directly express the camera movement a user has in mind. This is especially true for cartoon or comic-style scenes, where absolute metric scale may have no clear interpretation and relative camera-motion strength can provide a more intuitive interface.

To remove this preprocessing bottleneck while retaining intuitive control over motion strength, we propose to decouple trajectory shape from trajectory magnitude, as shown in Fig. 1(a). We normalize each nonzero translation path to unit length, which preserves motion direction, temporal ordering, and frame-wise relative translation magnitude while removing unreliable global scale. We then introduce a scalar control variable a to represent sequence-level motion strength. During training, the trajectory branch receives the normalized trajectory scaled by $1/a$, while a separate branch encodes a using a Fourier embedding of $\log a$. At inference time, the input trajectory remains unit-normalized, and a acts as an explicit knob for controlling camera-motion strength. This formulation supports scalable training on scale-ambiguous pose annotations without an additional metric-alignment stage, while letting users adjust relative translation strength through the dimensionless parameter a without specifying distances in meters.

This representation also provides a natural basis for preference-based post-training. Existing camera-controlled models are typically adapted from text-to-video models through supervised fine-tuning with ControlNet or LoRA (Bahmani et al., 2025a; Cheng et al., 2026; Zhang et al., 2023; Hu et al., 2022). While preference optimization methods such as DiffusionDPO (Wallace et al., 2024) can further refine generation behavior, applying them to camera control is challenging because preference pairs must reflect trajectory consistency, not only visual quality. Instead of relying on externally generated preference datasets, we introduce **Trajectory Preference Optimization** (TPO), an on-the-fly negative-trajectory construction method that uses unit trajectories to make translation perturbations comparable across videos. Since translation is represented by direction and frame-wise relative magnitude rather than metric scale, TPO can perturb the unit trajectory directly while keeping the video fixed. Specifically, we sample a subset of temporal anchor frames and perturb the frame-relative translation direction, relative translation magnitude, and/or frame-relative rotation at these anchors. This produces structured trajectory negatives: because camera trajectories are reconstructed by accumulating frame-relative translations and composing frame-relative rotations, errors at a few anchors can propagate to subsequent poses and make the entire trajectory inconsistent with the observed video. As a result, TPO yields preference pairs that directly target camera-trajectory fidelity while controlling for appearance, content, and video quality, as illustrated in Fig. 1(b).

Static control is also important when the scene moves but the camera viewpoint should remain fixed. This cannot be achieved by simply removing the camera condition, because removing the condition means "no constraint" rather than "no movement." Static control is also naturally handled by TPO: for static-camera videos, we construct negative trajectories by injecting small artificial translation and rotation perturbations, thereby teaching the model to distinguish static from non-static camera behavior. To support scalable training and broader evaluation, we collect a large-scale camera-annotated video dataset from diverse publicly available sources, including OpenVid-HD (Nan et al., 2025), SpatialVID-HQ (Wang et al., 2025), Sekai real-drone (Li et al., 2025b), and RealCam-Vid (Zheng et al., 2025), which contains videos from DL3DV (Ling et al., 2024), RealEstate10K (Zhou et al., 2018), and MiraData9K (Ju et al., 2024). The main contributions are summarized as follows:

- We introduce a scale-decoupled camera representation for scalable training without expensive metric-scale alignment, normalizing per-video translation length while preserving trajectory direction and frame-wise relative motion.
- We propose Trajectory Preference Optimization, a preference-based post-training method that constructs structured trajectory negatives during training by perturbing sparse anchor motions while keeping the video fixed.
- We collect a large-scale camera-annotated video dataset from diverse sources to scale up training and broaden the evaluation of camera-controlled video generation, including static-camera control.
- On our 2,000-video test set, our size-matched 2B and 5B models, trained for 20,000 and 10,000 SFT steps, respectively, reduce translation error by 49.4% and 36.8%, rotation error by 38.2% and 38.9%, FVD by 40.7% and 43.3%, and FID by 15.9% and 12.0%, respectively, relative to the corresponding AC3D baselines (Bahmani et al., 2025a), while maintaining comparable or slightly higher CLIP scores.

2 Related Work

Diffusion Models. DDPM (Ho et al., 2020) introduces diffusion models for high-quality image generation through iterative denoising, but its sampling process is inefficient because it typically requires a large number of denoising steps. To improve sampling efficiency, subsequent works propose advanced samplers that reduce the number of sampling steps while preserving generation quality (Song et al., 2021; Lu et al., 2022b;a; Zhao et al., 2023b). Latent Diffusion Models (LDM) introduce autoencoders (Rombach et al., 2022) to perform diffusion in a compressed latent space, substantially improving the efficiency of high-resolution training and inference. Together with cross-attention mechanisms (Vaswani et al., 2017), LDM also enables flexible text-conditioned generation. Building on these foundations, diffusion models have been applied to a broad range of downstream tasks (Li et al., 2023; Lyu et al., 2024; Lyu & Chen, 2025). ControlNet and subsequent works (Zhang et al., 2023; Li et al., 2024; Lyu et al., 2025; Zhao et al., 2023a) further improve controllability by incorporating additional conditions such as Canny edges. Diffusion Transformer (Peebles & Xie, 2023) provides an architecture for scaling diffusion models with transformer backbones, while flow matching has recently emerged as another scalable formulation for image and video generation (Lipman et al., 2023). These advances have been extended to video generation: early spatio-temporal diffusion models show that coherent short-video synthesis can be learned directly in pixel space (Ho et al., 2022b), while methods such as Imagen Video (Ho et al., 2022a) and Make-A-Video (Singer et al., 2022) improve quality, resolution, and temporal consistency through cascaded generation. Latent video diffusion further improves efficiency and generation quality (He et al., 2022; Blattmann et al., 2023). More recent works scale text-to-video and image-to-video generation with Diffusion Transformer backbones (Yang et al., 2024c; Kong et al., 2024; Wan et al., 2025). Recent audio-video foundation models further extend generative modeling to joint audio-video synthesis (HaCohen et al., 2026).

Camera-Controlled Video Generation. Recent methods augment video generators with explicit camera trajectories to improve controllability. MotionCtrl separates camera motion from object motion for controllable video generation (Wang et al., 2024). CameraCtrl studies plug-and-play camera control for text-to-video generation using Plücker embedding as the camera condition (He et al., 2024). CamCo also employs Plücker embedding and further introduces epipolar attention for camera-controlled image-to-video generation (Xu et al., 2024). VD3D (Bahmani et al., 2025b)

adopts a Diffusion Transformer backbone with ControlNet-like camera conditioning. RealCam-I2V (Li et al., 2025a) identifies metric-scale inconsistency as an important issue in camera-controlled video generation and proposes metric-scale alignment for image-to-video generation. AC3D further analyzes conditioning schedules and pose parameterization in video diffusion transformers (Bahmani et al., 2025a). MoCa (Cheng et al., 2026) observes that camera conditioning can degrade appearance consistency and uses a reference text-to-video generation model as an additional condition to improve appearance preservation. UCPE (Zhang et al., 2026) utilizes a flow-based geometry recovery method and combines relative ray and absolute orientation encodings to control camera poses, intrinsics, and lens distortion through lightweight spatial attention adapters.

Direct Preference Optimization. Direct preference optimization (DPO) is a reinforcement-learning-from-human-feedback (RLHF) method for post-training large language models, providing a simple way to optimize models from pairwise preference data without explicitly training a reward model (Rafailov et al., 2023). DiffusionDPO (Wallace et al., 2024) adapts DPO to image generation by rewriting the preference objective in terms of an ELBO-derived likelihood surrogate over diffusion paths. D3PO proposes another reward-model-free formulation for tuning diffusion models from human feedback (Yang et al., 2024a). Other alignment methods instead formulate denoising as a reinforcement-learning problem and optimize scalar rewards (Black et al., 2024; Fan et al., 2023). CPO (Lyu et al., 2025) exploits the imperfect controllability of diffusion models to construct negative conditions rather than negative images. CPO constructs negative conditions through image generation and condition extraction. Extending this procedure to video adds generation and annotation costs. In contrast, our TPO formulation is motivated by our scale-decoupled camera representation and perturbs camera translation and rotation during training, independent of the absolute translation scale.

3 METHOD

3.1 PRELIMINARIES

Diffusion Models. Diffusion models (Ho et al., 2020) define a forward diffusion process $q(\mathbf{x}_t \mid \mathbf{x}_0)$ that gradually corrupts a clean sample $\mathbf{x}_0$ with Gaussian noise, and a reverse sampling process $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ that progressively denoises the sample. The marginal distribution of the forward process is given by

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\big(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\big)\,, \quad \bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s), \tag{1}$$

where $\{\beta_t\}_{t=1}^{T}$ is a predefined noise schedule. For $t > 1$, the learned reverse process is parameterized as

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\big(\mathbf{x}_{t-1} \mid \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I}\big)\,, \tag{2}$$

where the noise-prediction parameterization is

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right). \tag{3}$$

Here, $\boldsymbol{\epsilon}_\theta$ predicts the Gaussian noise added in the forward process, and σ_t^2 is a prescribed reverse variance. A standard choice is $\sigma_t^2 = \tilde{\beta}_t = \beta_t(1-\bar{\alpha}_{t-1})/(1-\bar{\alpha}_t)$ for $t > 1$. The final step reconstructs $\mathbf{x}_0$ from $\mathbf{x}_1$.

Diffusion-DPO. Given preference pairs $\{\mathbf{x}_0^+, \mathbf{x}_0^-\}$, where $\mathbf{x}_0^+$ denotes the preferred sample and $\mathbf{x}_0^-$ denotes the less preferred sample, the goal is to fine-tune a pretrained diffusion model $\boldsymbol{\epsilon}_\theta$ such that it assigns higher preference to $\mathbf{x}_0^+$ than to $\mathbf{x}_0^-$. Based on the Bradley–Terry model (Bradley & Terry, 1952), Diffusion-DPO (Wallace et al., 2024) defines the preference objective as

$$\mathcal{L}_{\mathrm{DPO}}^{\mathrm{marg}} = -\mathbb{E}_{y,\mathbf{x}_0^+,\mathbf{x}_0^-} \log\sigma\left(\beta\left[\log\frac{p_\theta(\mathbf{x}_0^+ \mid y)}{p_{\mathrm{ref}}(\mathbf{x}_0^+ \mid y)} - \log\frac{p_\theta(\mathbf{x}_0^- \mid y)}{p_{\mathrm{ref}}(\mathbf{x}_0^- \mid y)}\right]\right). \tag{4}$$

Here, $\beta > 0$ controls the preference-logit scale and differs from the diffusion noise schedule $\{\beta_t\}$. Diffusion-DPO (Wallace et al., 2024) uses a tractable denoising surrogate:

$$\begin{aligned}\mathcal{L}_{\mathrm{DPO}}^{\mathrm{den}} := -\mathbb{E}_{(\mathbf{x}_0^+,\mathbf{x}_0^-),\,t,\,\boldsymbol{\epsilon}}\Big[\log\sigma\Big(&-\beta T\omega(\lambda_t)\big(\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta(\mathbf{x}_t^+,t)\|_2^2 - \|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_{\mathrm{ref}}(\mathbf{x}_t^+,t)\|_2^2 \\ &- \big(\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta(\mathbf{x}_t^-,t)\|_2^2 - \|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_{\mathrm{ref}}(\mathbf{x}_t^-,t)\|_2^2\big)\big)\Big)\Big].\end{aligned} \tag{5}$$

Here, t is uniform on $\{1, \ldots, T\}$, $\lambda_t = \bar{\alpha}_t/(1-\bar{\alpha}_t)$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. The practical objective uses a constant weight $\omega(\lambda_t) = 0.5$, rather than the exact variational weight. The noisy samples $\mathbf{x}_t^+$ and $\mathbf{x}_t^-$ are obtained from Eq. (1). The policy model $\boldsymbol{\epsilon}_\theta$ and the reference model $\boldsymbol{\epsilon}_{\text{ref}}$ are initialized from the same pretrained model, while only $\boldsymbol{\epsilon}_\theta$ is updated during training. The text condition y is also provided to both models, but is omitted from Eq. (5) for notational simplicity.

3.2 Trajectory-Scale Decoupling

Problem Setup for Camera-Controlled Video Generation. We consider a video generator conditioned on a text prompt y and a camera trajectory $\tau = \{(\mathbf{R}_s, \mathbf{c}_s)\}_{s=1}^S$, where $\mathbf{R}_s \in SO(3)$ and $\mathbf{c}_s \in \mathbb{R}^3$ denote the camera rotation matrix and camera center at frame s, respectively, and S is the number of video frames. We use s to index video frames and reserve t for diffusion timesteps.

A challenge for scalable training is that $\mathbf{c}_s$ is typically not metrically scaled when camera poses are estimated from monocular videos. In monocular structure-from-motion pipelines, camera trajectories are recovered only up to an unknown global scale. As a result, semantically similar camera motions may have inconsistent numerical magnitudes across different videos. Directly using these raw translations for training weakens the correspondence between the camera condition and the observed image motion, making global motion control ambiguous.

Scale-Decoupled Camera Trajectory. To avoid calibrating global scale for every training video, we remove the absolute magnitude of translation from the pose condition. Given a camera trajectory τ, we define its total translation path length as $L(\tau) = \sum_{s=2}^S \|\mathbf{c}_s - \mathbf{c}_{s-1}\|_2$. We then normalize the camera centers by this path length:

$$\hat{\mathbf{c}}_s = \frac{\mathbf{c}_s - \mathbf{c}_1}{\max(L(\tau), \epsilon)}, \tag{6}$$

where $\epsilon > 0$ is a small constant for numerical stability. The normalized path has unit length when $L(\tau) \geq \epsilon$. Zero-translation trajectories remain zero, including pure-rotation and static-camera trajectories.

This transformation preserves the trajectory properties that are most useful for camera control: the overall direction of motion, the temporal ordering of frames, and the relative translation magnitude across frames. In contrast, it discards the unknown global scale, which is precisely the quantity that is difficult to estimate consistently across large video corpora. Intuitively, Eq. (6) converts the translation sequence into a canonical *unit trajectory* that describes *how* the camera moves, while removing *how far* it moves in absolute metric units.

To recover controllable global motion strength, we introduce a scalar variable $a > 0$ that represents relative sequence-level translation strength. Instead of encoding translation with a single branch, we use two complementary branches during training:

$$\mathbf{e}_s^c = \Pi(\mathbf{R}_s, \hat{\mathbf{c}}_s/a)\,, \quad \mathbf{e}^a = \phi(\log a). \tag{7}$$

Here, $\mathbf{e}_s^c$ denotes the camera pose embedding at frame s, and $\mathbf{e}^a$ denotes the global magnitude embedding. The function $\Pi(\mathbf{R}_s, \hat{\mathbf{c}}_s/a)$ constructs a dense Plücker-ray encoding from the camera-to-world rotation matrix relative to the first frame and the scaled unit translation. Details on the formulation of ϕ are in Appendix B.

During inference, we keep the input camera trajectory at unit length and use a as an explicit control variable for motion magnitude: $\mathbf{e}_s^c = \Pi(\mathbf{R}_s, \hat{\mathbf{c}}_s)\,, \quad \mathbf{e}^a = \phi(\log a)$.

During training, we sample a while keeping the target video fixed. The translation input and magnitude variable form a reciprocal pair: $a(\hat{\mathbf{c}}_s/a) = \hat{\mathbf{c}}_s$. This construction encourages the model to associate a with the translation scale of the pose branch. At inference, we hold the pose branch at the unit trajectory $\hat{\mathbf{c}}_s$ and vary a, testing whether the learned relationship yields stronger generated translation for larger a. This response is learned rather than guaranteed or metrically calibrated. We evaluate it empirically in Sec. 5.2.

Paired videos with identical trajectory shape and rotations but different path lengths are impractical to collect. We therefore learn translation-strength control without paired magnitude supervision.

Appendix B details this unsupervised scale-control mechanism, whose effectiveness is demonstrated in Sec. 5.2. In a CogVideoX-based camera-control branch (Yang et al., 2024c; Bahmani et al., 2025a), we inject the camera pose embedding $\mathbf{e}^c$ and the magnitude embedding $\mathbf{e}^a$ into the diffusion transformer together with the noisy video latent $\mathbf{x}_t$ at diffusion timestep t:

$$\mathbf{h}_t = \mathrm{PatchEmb}(\mathrm{cat}\,([\mathbf{x}_t, \mathbf{e}^c])) + f_\theta(\mathbf{e}^a). \tag{8}$$

Here, $\mathrm{cat}(\cdot)$ denotes channel-wise concatenation, and f_θ is a two-layer MLP whose final layer is zero-initialized. The magnitude embedding is broadcast to match the video-token representation before addition. The resulting hidden representation $\mathbf{h}_t$ is fed into the transformer layers of the camera-control branch.

3.3 Trajectory Preference Optimization

To construct trajectory preference pairs, we keep the video fixed and perturb only the camera condition. For each training video $\mathbf{x}_0$ with camera trajectory τ and text prompt y, we omit y for notational simplicity and define the positive normalized trajectory as $\tau^+ = \{(\mathbf{R}_s, \hat{\mathbf{c}}_s)\}_{s=1}^S$. We then construct a negative trajectory by applying a trajectory perturbation operator: $\tau^- = \mathcal{J}(\tau^+)$, where $\mathcal{J}(\cdot)$ perturbs frame-relative translations and rotations, such as changing the translation direction, modifying the frame-wise relative translation magnitude, or perturbing yaw, pitch, and roll. The original pair $(\mathbf{x}_0, \tau^+)$ is treated as the preferred condition, while $(\mathbf{x}_0, \tau^-)$ is treated as the rejected condition. Since the video $\mathbf{x}_0$ is shared by both pairs, the preference signal is focused on trajectory fidelity rather than visual appearance or video quality.

Inspired by Diffusion-DPO, we define Trajectory Preference Optimization (TPO) to rank camera conditions for the same video using policy-to-reference likelihood ratios:

$$\mathcal{L}_{\mathrm{TPO}}^{\mathrm{marg}} = -\mathbb{E}_{\tau^+,\tau^-,\mathbf{x}_0} \log \sigma\left(\beta\left[\log \frac{p_\theta(\mathbf{x}_0 \mid \tau^+)}{p_{\mathrm{ref}}(\mathbf{x}_0 \mid \tau^+)} - \log \frac{p_\theta(\mathbf{x}_0 \mid \tau^-)}{p_{\mathrm{ref}}(\mathbf{x}_0 \mid \tau^-)}\right]\right). \tag{9}$$

Here, p_θ denotes the trainable policy model, and p_{ref} denotes the frozen reference model initialized from the same SFT checkpoint as the policy. In our TPO run, both start from the 16,000-step 2B SFT checkpoint. Intuitively, this objective encourages the policy model to assign a higher relative likelihood to the video when conditioned on the original trajectory than when conditioned on the perturbed trajectory.

We optimize a denoising surrogate in the form of Eq. (5). Appendix C explains its connection to the variational objective and distinguishes it from the marginal-likelihood loss:

$$\begin{aligned}\mathcal{L}_{\mathrm{TPO}}^{\mathrm{den}} := -\mathbb{E}_{(\tau^+,\tau^-,\mathbf{x}_0),\,t,\,\boldsymbol{\epsilon}}\Big[\log \sigma\Big(-\beta T \omega(\lambda_t)\Big(\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \tau^+, t)\|_2^2 - \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\mathrm{ref}}(\mathbf{x}_t, \tau^+, t)\|_2^2 \\ - \Big(\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \tau^-, t)\|_2^2 - \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\mathrm{ref}}(\mathbf{x}_t, \tau^-, t)\|_2^2\Big)\Big)\Big)\Big].\end{aligned} \tag{10}$$

Both branches use the same clean video $\mathbf{x}_0$, timestep t, and noise sample $\boldsymbol{\epsilon}$. The only difference is the camera trajectory condition. Therefore, TPO directly optimizes trajectory-conditioned denoising preference while controlling for the underlying video content.

Negative Trajectory Generation. We construct τ^- by perturbing frame-relative translations and rotations at sparse anchor frames of the positive unit trajectory. Translation-only, rotation-only, and joint perturbations are sampled with probabilities 20%, 20%, and 60%, respectively. Accumulating the perturbed relative motions produces negative trajectories with coupled changes across frames. For static-camera videos, we inject small additive motions. The complete procedure and perturbation parameters are provided in Appendix D.

4 Dataset Collection

Our dataset contains approximately 532K videos collected from five source subsets. We include 100K static-camera videos from OpenVid-HD (Nan et al., 2025), selected from the highest-aesthetic subset, and approximately 76K non-static OpenVid-HD videos with aesthetic score at least 5.5. We further include 235K motion-rich videos from SpatialVID-HQ (Wang et al., 2025), 24K real drone videos from Sekai real-drone (Li et al., 2025b), and 97K videos from RealCam-Vid (Zheng et al.,

2025). RealCam-Vid contains videos from RE10K (Zhou et al., 2018), DL3DV (Ling et al., 2024), and MiraData9K (Ju et al., 2024). All videos are required to contain at least 49 frames. We store videos at 8 fps by default. If a video has fewer than 49 frames at 8 fps, we increase the sampling frame rate up to the original frame rate. Beyond these source-specific selections, we apply only the frame-count constraint.

Our scale-decoupled representation allows these sources to be combined without additional metric-scale alignment, while camera-pose annotations remain necessary for supervision. The dynamic OpenVid-HD subset is annotated with MegaSAM (Li et al., 2025c), following the preprocessing recipe of SpatialVID-HQ, due to its favorable efficiency-accuracy trade-off. For static-camera videos, we set the camera extrinsics to zero translation and identity rotation. The normalized camera intrinsics are set to $[0.8333, 1.25, 0.5, 0.5]$ (see Appendix E for details). This mixture combines high-quality static scenes, open-domain videos, indoor videos, and motion-rich clips, providing diverse supervision for camera-controlled video generation. The dataset composition is summarized in Table 2 in Appendix A.

To construct an evaluation set, we randomly select 250 videos from OpenVid-HD static, 250 videos each from SpatialVID-HQ with motion tags 'forward', 'left forward', and 'right forward', as they are the most dominant tags. We also randomly select 250 videos from Sekai real-drone (Li et al., 2025b) and 250 videos each from the three subsets (RE10K (Zhou et al., 2018), DL3DV (Ling et al., 2024), MiraData9K (Ju et al., 2024)) in RealCam-Vid (Zheng et al., 2025). These sources provide varied scene content and camera motion, including open-domain videos from MiraData9K. The resulting 2,000 evaluation clips are held out from all training stages. The summary is shown in Table 3 in Appendix A.

5 Experiments

Baselines and Evaluation Metrics. We compare our method with three open-sourced camera-controlled text-to-video generation models, CameraCtrl (He et al., 2024), AC3D (Bahmani et al., 2025a), and UCPE (Zhang et al., 2026). Following AC3D (Bahmani et al., 2025a), we report translation error (Trans.↓), rotation error (Rot.↓), Fréchet Video Distance (FVD↓) (Unterthiner et al., 2019), Fréchet Inception Distance computed with the clean-FID implementation (FID↓) (Parmar et al., 2022), and CLIP score (CLIP↑) (Radford et al., 2021). Our models are trained and evaluated with 49 frames following (Yang et al., 2024c). FID is computed with uniformly sampled 8 frames per video to reduce the computation burden. We adapt the rotation and translation distances from CameraCtrl (He et al., 2024), using our path-length normalization for translation:

$$\text{Rot} = \sum_{i=1}^{S} \arccos\left(\frac{\text{tr}\left(\mathbf{R}^i_{\text{gen}}\mathbf{R}^{i\top}_{\text{gt}}\right) - 1}{2}\right), \quad \text{Trans} = \sum_{i=1}^{S} \left\|\hat{\mathbf{c}}^i_{\text{gt}} - \hat{\mathbf{c}}^i_{\text{gen}}\right\|_2^2. \tag{11}$$

Here, $\hat{\mathbf{c}}$ denotes the normalized translation defined in Eq. 6, and $\mathbf{R}$ denotes the camera-to-world rotation matrix relative to the first frame. Because ground-truth and generated translations are normalized independently, Trans. measures trajectory shape and is invariant to global translation magnitude. We evaluate the magnitude response separately in Fig. 4. For the main quantitative comparison, we estimate camera poses from generated videos with MegaSAM (Li et al., 2025c). Pose estimation succeeded for all generated videos, so no samples were excluded from Trans. or Rot. due to estimation failure.

Training Setup. Following AC3D (Bahmani et al., 2025a), we train camera-control branches with CogVideoX-2B and CogVideoX-5B (Yang et al., 2024c) as the base video generation models. The SFT-only 2B and 5B models are trained for 20,000 and 10,000 steps, respectively. For Ours-2B+TPO, we initialize from a 16,000-step SFT checkpoint of the 2B model and perform 4,000 TPO steps with a global batch size of 64, giving 20,000 optimization steps in total. Thus, Ours-2B and Ours-2B+TPO have the same total number of optimization steps, allocated as 20,000 SFT steps versus 16,000 SFT plus 4,000 TPO steps. Additional details are provided in Appendix F.

5.1 Evaluation Results

Quantitative Results. Table 1 reports four generated videos per test sample. All three variants outperform CameraCtrl and AC3D on translation error, rotation error, FVD, and FID. Against size-

Table 1: Quantitative results on the held-out 2,000-video test set. Ours-2B uses 20k SFT steps, Ours-5B uses 10k SFT steps, and Ours-2B+TPO uses 16k SFT plus 4k TPO steps. Arrows indicate preferred metric directions. Best and second-best results are **boldfaced** and underlined.

Method	Trans.↓	Rot.↓	FVD↓	FID↓	CLIP Score↑
CameraCtrl (He et al., 2024)	0.0459	0.0799	362.48	34.90	0.3089
AC3D-2B (Bahmani et al., 2025a)	0.0271	0.0942	256.09	19.48	0.3244
AC3D-5B	0.0223	0.0868	265.83	18.52	0.3265
UCPE (Zhang et al., 2026)	**0.0104**	0.0684	368.10	20.65	0.3232
Ours-2B	0.0137	0.0582	151.74	16.38	0.3248
Ours-2B+TPO	0.0131	0.0559	**150.12**	16.70	0.3249
Ours-5B	0.0141	**0.0530**	150.60	**16.30**	**0.3268**

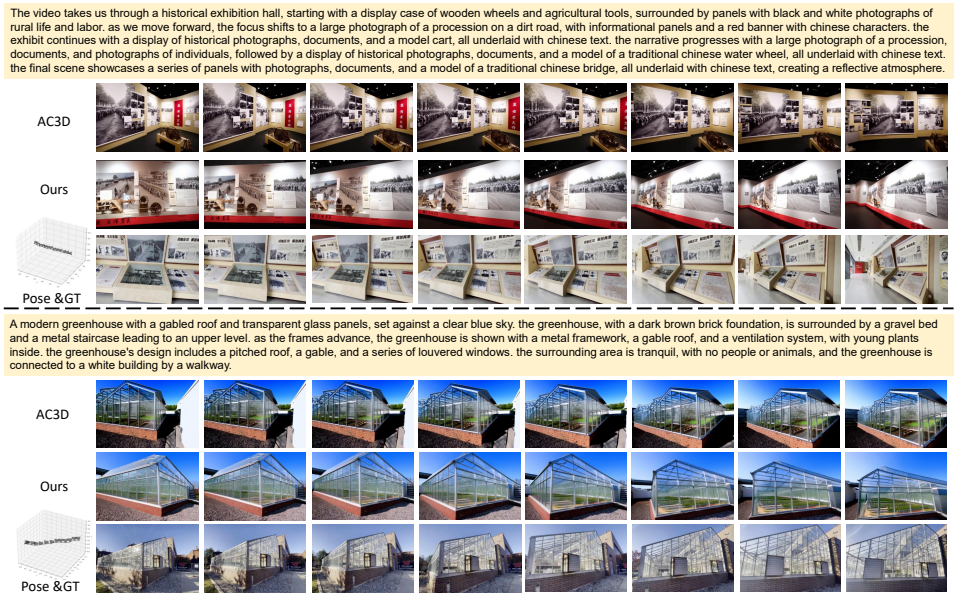


Figure 2: Qualitative comparison with AC3D (Bahmani et al., 2025a) and ground truth. Each row shows eight uniformly sampled frames from a 49-frame video.

matched AC3D baselines, our 2B and 5B models reduce translation error by 49.4% and 36.8%, rotation error by 38.2% and 38.9%, FVD by 40.7% and 43.3%, and FID by 15.9% and 12.0%, respectively. Ours-5B obtains the highest CLIP score. The 5B model is trained for 10,000 steps while the 2B model is trained for 20,000 steps. On the held-out RealEstate10K subset of our test split, our 2B model also improves FVD and trajectory errors over the released AC3D-2B checkpoint (Table 4).

UCPE uses Wan2.1-T2V-1.3B, a stronger base video generator than CogVideoX-5B (Wan et al., 2025), yet our CogVideoX models outperform it on all metrics except translation error. Our Wan2.1 version is still training, and its results will be updated.

At 20,000 total steps, allocating 16,000 to SFT and 4,000 to TPO reduces translation and rotation errors by 4.4% and 4.0%, respectively, compared with the SFT-only 2B baseline. FVD, FID, and CLIP score remain broadly stable, consistent with TPO's objective of improving camera control while preserving visual quality and text alignment. Since each preference pair shares the same video, improved trajectory accuracy without gains in these other metrics is the intended outcome.

Qualitative Results. We include qualitative comparisons between our method and AC3D (Bahmani et al., 2025a) in Fig. 2, along with GT video as reference. Our models achieve a more similar camera motion to the ground truth video. We include additional comparisons in Appendix H together with static controls.

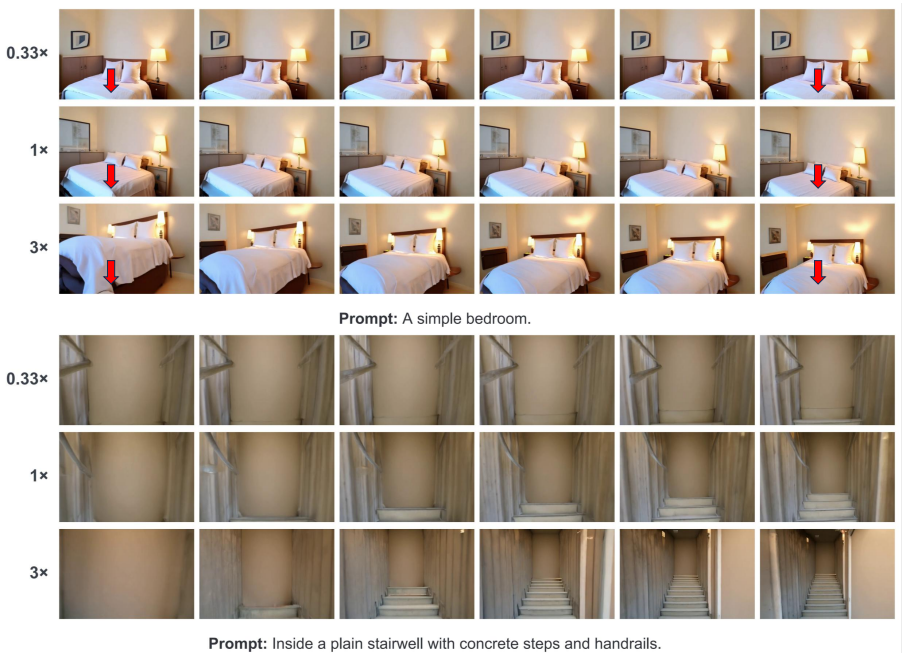


Figure 3: Qualitative scale ablation with scale parameters 0.33, 1, and 3, shown from top to bottom for each prompt. Each row contains six frames uniformly sampled from the generated video.

5.2 Ablation Study

We evaluate the effect of the scale parameter a on camera-motion magnitude. Figure 3 presents qualitative comparisons at $a \in \{0.33, 1, 3\}$ for bedroom and stairwell scenes. In these examples, larger scale parameters produce stronger viewpoint changes, illustrating translation-strength control learned without paired magnitude supervision.

We evaluate our 20k-step 2B checkpoint on 2,000 translation-only test poses. For each sample, we fix the prompt, unit trajectory, and generation seed across $a \in \{1, 5, 10\}$. We estimate generated camera trajectories with MegaSAM and apply per-video metric-scale alignment adapted from AC3D (Bahmani et al., 2025a). The estimated scale $L(a)$ is the sum of Euclidean distances between consecutive aligned camera centers. For each a, we average the 2,000 per-sample ratios $L(a)/L(1)$. Despite training without metric-scale alignment, our model achieves an average relative scale response comparable to metric-aligned AC3D-2B over the tested range (Fig. 4).

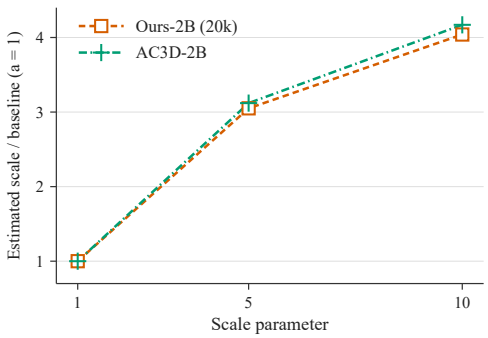


Figure 4: Scale response of Ours-2B (20k steps) and AC3D-2B. Our curve averages 2,000 per-sample scale ratios relative to the same checkpoint's $a = 1$ baseline.

6 Conclusion

We introduce a scale-decoupled camera representation that separates normalized trajectory shape from global motion magnitude, enabling scalable training without metric-scale calibration. Trajectory Preference Optimization complements this representation by perturbing camera conditions while keeping the video fixed. On the held-out 2,000-video test set, our 2B and 5B models outperform the corresponding AC3D baselines after 20,000 and 10,000 SFT steps, respectively. Allocating the 2B model's 20,000 steps to 16,000 SFT steps and 4,000 TPO steps further improves trajectory accuracy. The scale ablation shows a nonlinear average translation response learned without paired magnitude supervision on the evaluation set.

AI USE STATEMENT

We used generative AI to produce an initial draft of the paper, which we manually rewrote, and subsequently to polish the writing. OpenAI Codex also assisted with manuscript editing and LaTeX formatting. The authors take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Improved camera control can support filmmaking, education, simulation, and creative prototyping, but can also facilitate deceptive or manipulative synthetic footage. We discuss these potential impacts and the need for deployment safeguards in Appendix K.

REPRODUCIBILITY STATEMENT

The paper describes the data sources and held-out evaluation split, camera representation, training schedules, and evaluation metrics. Appendix B details the scale embedding, Appendix C derives the training surrogate, and Appendices D, E, and F describe trajectory perturbations, default intrinsics, and additional training settings. Appendix J discusses limitations.

REFERENCES

Sherwin Bahmani, Ivan Skorokhodov, Guocheng Qian, Aliaksandr Siarohin, Willi Menapace, Andrea Tagliasacchi, David B Lindell, and Sergey Tulyakov. Ac3d: Analyzing and improving 3d camera control in video diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025a.

Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, David B. Lindell, and Sergey Tulyakov. Vd3d: Taming large video diffusion transformers for 3d camera control. *Proc. ICLR*, 2025b.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 1952.

Chenjie Cao, Jingkai Zhou, Shikai Li, Jingyun Liang, Chaohui Yu, Fan Wang, Xiangyang Xue, and Yanwei Fu. Uni3C: Unifying precisely 3D-enhanced camera and human motion controls for video generation. *arXiv preprint arXiv:2504.14899*, 2025.

Zhijing Cheng, Xuancheng Zhang, Donglin Di, Chen Wei, Hao Li, and Xun Yang. Moca: Modeling object consistency for 3d camera control in video generation. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 2023.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, Eitan Richardson, Guy Shiran, Itay Chachy, Jonathan Chetboun, Michael Finkelson, Michael Kupchick, Nir Zabari, Nitzan Bitton Guetta, N. G. Kotler, Ofir Bibi, Ori Gordon, Poriya Panet, Roi Benita, Shahar Armon, V. M. Kulikov, Yaron Inger, Yonatan Shiftan, Zeev Melumian, and Zeev Farbman. Ltx-2: Efficient joint audio-visual foundation model. *ArXiv*, 2026.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 2022b.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 2022.

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. *Advances in Neural Information Processing Systems*, 2024.

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

Bo Li, Kaitao Xue, Bin Liu, and Yu-Kun Lai. Bbdm: Image-to-image translation with brownian bridge diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern Recognition*, 2023.

Ming Li, Taojiannan Yang, Huafeng Kuang, Jie Wu, Zhaoning Wang, Xuefeng Xiao, and Chen Chen. Controlnet++: Improving conditional controls with efficient consistency feedback. In *European Conference on Computer Vision*, 2024.

Teng Li, Guangcong Zheng, Rui Jiang, Shuigen Zhan, Tao Wu, Yehao Lu, Yining Lin, Chuanyun Deng, Yepan Xiong, Min Chen, et al. Realcam-i2v: Real-world image-to-video generation with interactive complex camera control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025a.

Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, Zizhen Li, Fanrui Zhang, Jiaxin Ai, Zhixiang Wang, Yuwei Wu, Tong He, Jiangmiao Pang, Yu Qiao, Yunde Jia, and Kaipeng Zhang. Sekai: A video dataset towards world exploration. *arXiv preprint arXiv:2506.15675*, 2025b.

Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. MegaSaM: Accurate, Fast and Robust Structure and Motion from Casual Dynamic Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025c.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022a.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022b.

Zonglin Lyu and Chen Chen. Tlb-vfi: Temporal-aware latent brownian bridge diffusion for video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2025.

Zonglin Lyu, Ming Li, Jianbo Jiao, and Chen Chen. Frame interpolation with consecutive brownian bridge diffusion. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024.

Zonglin Lyu, Ming Li, Xinxin Liu, and Chen Chen. CPO: Condition preference optimization for controllable image generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 2021.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 2023.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. GEN3C: 3D-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6121–6132, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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.

Wan-AI. Wan2.2-I2V-A14B. Model card, 2025. URL `https://huggingface.co/Wan-AI/Wan2.2-I2V-A14B`.

Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. Spatialvid: A large-scale video dataset with spatial annotations. *arXiv preprint arXiv:2509.09676*, 2025.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, 2024.

Dejia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. Camco: Camera-controllable 3d-consistent image-to-video generation. *arXiv preprint arXiv:2406.02509*, 2024.

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024a.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 2024b.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024c.

Cheng Zhang, Boying Li, Meng Wei, Yan-Pei Cao, Camilo Gambardella, Dinh Phung, and Jianfei Cai. Unified camera positional encoding for controlled video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38027–38037, 2026.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 2023a.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 2023b.

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.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

Appendix

The appendix is structured as follows:

- Section A: dataset composition and held-out evaluation split.
- Section B: details of formulation of ϕ.
- Section C: derivation of the TPO denoising surrogate.
- Section D: negative trajectory generation.
- Section E: details of selecting default intrinsics.
- Section F: additional training details.
- Section G: adaptation to metric-scale control.
- Section H: additional qualitative results.
- Section I: comparison with GEN3C and Uni3C.
- Section J: limitations and future work.
- Section K: broader impact.

A Dataset Composition and Evaluation Split

Tables 2 and 3 summarize the dataset composition and held-out evaluation split described in the main paper.

Table 2: Composition of our dataset.

Source	Videos	Notes
OpenVid-HD static	100K	High-aesthetic static-camera videos
OpenVid-HD dynamic	∼76K	Aesthetic score ≥ 5.5, MegaSAM annotations
SpatialVID-HQ	∼235K	Motion-rich videos
Sekai real-drone	∼24K	Real drone videos
RealCam-Vid	∼97K	RE10K, DL3DV, and MiraData
Total	∼532K	Training corpus

Table 3: Held-out test split used for the main comparisons.

Source	Category	Clips
OpenVid-HD	Static	250
SpatialVID-HQ	Forward	250
SpatialVID-HQ	Left-forward	250
SpatialVID-HQ	Right-forward	250
Sekai	Drone	250
RealCam-Vid	RE10K	250
RealCam-Vid	DL3DV	250
RealCam-Vid	MiraData	250
Total	–	2,000

Held-out RealEstate10K subset. To assess performance on held-out samples from a source included in our training corpus, we separately evaluate the 250 RealEstate10K (RE10K) clips in our 2,000-video test split. Table 4 compares the released AC3D-2B checkpoint with Ours-2B after 20,000 SFT steps, before any metric-aligned fine-tuning. Both models use the main-paper evaluation protocol. These clips are from our test split, not AC3D's RealEstate10K test split used in Appendix G. Ours achieves lower FVD, rotation error, and path-normalized translation error on this held-out subset.

Table 4: Results on the 250 held-out RealEstate10K clips in our main test split. We compare the released AC3D-2B checkpoint with our 20k-step SFT CogVideoX-2B model before metric-aligned fine-tuning, using the main-paper evaluation protocol. Lower is better.

Method	Trans.↓	Rot.↓	FVD↓
AC3D-2B	0.01502	0.05990	458.18
Ours-2B	**0.00837**	**0.03968**	**395.57**

B Scale Embedding

The scale parameter is intended to strengthen or weaken camera translation along a fixed trajectory shape and rotation sequence. Paired videos with the same relative translation pattern and rotations but different path lengths are unavailable and impractical to collect at scale. We therefore design scale control as an unsupervised mechanism, using Eq. (7) while keeping the video target fixed. Here, unsupervised refers to the absence of paired magnitude supervision. The generator retains video–camera supervision, and Sec. 5.2 demonstrates the effectiveness of scale control.

The scalar a is encoded separately to represent the global motion magnitude. We embed $\log a$ using a sinusoidal-style embedding:

$$\phi(z) = [z,\, \cos(\pi\omega_1 z),\, \sin(\pi\omega_1 z),\, \ldots,\, \cos(\pi\omega_K z),\, \sin(\pi\omega_K z)]\,, \tag{12}$$

where the frequencies are logarithmically spaced:

$$\omega_k = \exp\left(\log\omega_{\min} + \frac{k-1}{K-1}\left(\log\omega_{\max} - \log\omega_{\min}\right)\right), \qquad k = 1, \ldots, K. \tag{13}$$

We use $K = 6$ logarithmically spaced frequencies with $\omega_{\min} = 1$ and $\omega_{\max} = 64$, giving a 13-dimensional embedding.

C Derivation of the TPO Denoising Surrogate

TPO ranks two camera conditions for the same video using the policy-to-reference likelihood ratios in Eq. (9). The clean video and text prompt are fixed within each pair. We omit the text prompt below and write $q_t = q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$. The forward process is independent of the camera condition.

For $t > 1$, define the difference of conditional denoising KL terms as

$$\begin{aligned} G_t(\tau) = {} & D_{\mathrm{KL}}(q_t \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \tau)) \\ & - D_{\mathrm{KL}}(q_t \,\|\, p_{\mathrm{ref}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \tau))\,. \end{aligned} \tag{14}$$

Assuming the policy and reference share the same prescribed reverse variance, Gaussian KL terms give

$$G_t(\tau) = w_t\, \delta_\theta(\tau, t, \boldsymbol{\epsilon}), \qquad w_t = \frac{\beta_t^2}{2\sigma_t^2(1-\beta_t)(1-\bar{\alpha}_t)}, \tag{15}$$

where

$$\begin{aligned} \delta_\theta(\tau, t, \boldsymbol{\epsilon}) = {} & \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \tau, t)\|_2^2 \\ & - \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\mathrm{ref}}(\mathbf{x}_t, \tau, t)\|_2^2, \end{aligned} \tag{16}$$

and $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. This is the standard Gaussian denoising identity (Ho et al., 2020). The exact variational decomposition also contains a reconstruction term at $t = 1$. A constant denoising weight is a practical reweighting of this decomposition, rather than an exact equality to the marginal log likelihood.

To state the practical objective precisely, define a denoising score

$$\widetilde{r}_\theta(\mathbf{x}_0, \tau) = -T\, \mathbb{E}_{t,\boldsymbol{\epsilon}}\left[\omega(\lambda_t)\, \delta_\theta(\tau, t, \boldsymbol{\epsilon})\right], \tag{17}$$

where t is uniform on $\{1, \ldots, T\}$ and $\omega(\lambda_t)$ is the weight in Eq. (10). Replacing marginal log-likelihood ratios by these denoising scores gives the score-based objective

$$\mathcal{L}_{\mathrm{TPO}}^{\mathrm{score}} = -\mathbb{E}_{\tau^+, \tau^-, \mathbf{x}_0} \log\sigma\big(\beta\left[\widetilde{r}_\theta(\mathbf{x}_0, \tau^+) - \widetilde{r}_\theta(\mathbf{x}_0, \tau^-)\right]\big)\,. \tag{18}$$

Writing $D_\theta(t, \boldsymbol{\epsilon}) = \delta_\theta(\tau^+, t, \boldsymbol{\epsilon}) - \delta_\theta(\tau^-, t, \boldsymbol{\epsilon})$, the convexity of $-\log\sigma$ and Jensen's inequality give

$$\begin{aligned}\mathcal{L}_{\text{TPO}}^{\text{score}} \leq & -\mathbb{E}_{\tau^+,\tau^-,\mathbf{x}_0,t,\boldsymbol{\epsilon}} \log\sigma(-\beta T\omega(\lambda_t) D_\theta(t,\boldsymbol{\epsilon})) \\ = & \mathcal{L}_{\text{TPO}}^{\text{den}}.\end{aligned} \tag{19}$$

Expanding D_θ recovers Eq. (10). Both conditions share the same clean video, timestep, and noise sample. This inequality bounds the score-based objective. The denoising score is a surrogate for the marginal likelihood ratio, so the inequality does not establish a bound on Eq. (9) itself. This distinction follows the use of a variational approximation and denoising reweighting in diffusion preference optimization (Wallace et al., 2024).

D Negative Trajectory Generation

To generate the negative trajectory τ^-, we perturb the positive unit trajectory $\tau^+ = \{(\mathbf{R}_s, \hat{\mathbf{c}}_s)\}_{s=1}^S$. With probability 20%, we perturb only translation. With probability 20%, we perturb only rotation, and with probability 60%, we perturb both. For clips with translation-only camera motion, we override this sampling rule and perturb only translation, leaving the rotation sequence unchanged. For each perturbation, we select k distinct anchor frames from indices $2, \ldots, S$, with $3 \leq k \leq S-1$ ($S = 49$), and define $\gamma = \sqrt{(S-1)/k}$. Fewer anchors lead to stronger perturbations. We perturb sparse anchors rather than independently jittering every frame because camera trajectories are cumulative: translation is reconstructed from frame-relative translations, and rotation is reconstructed from frame-relative rotations. Errors at a few anchors can therefore propagate to subsequent poses through cumulative reconstruction.

For moving-camera videos, translation perturbation is performed as follows:

1. We select k anchor frames and compute their frame-relative translations $\Delta\hat{\mathbf{c}}_s = \hat{\mathbf{c}}_s - \hat{\mathbf{c}}_{s-1}$ at the selected anchors.
2. We choose to perturb translation direction, relative magnitude, or both with equal probability. For direction perturbation, we normalize each relative translation to unit length, add Gaussian noise $\boldsymbol{\eta}_s \sim \mathcal{N}(0, (0.2\gamma)^2\mathbf{I})$, and re-normalize it to the length of $\Delta\hat{\mathbf{c}}_s$.
3. For magnitude perturbation, we compute the relative translation length $\ell_s = \|\Delta\hat{\mathbf{c}}_s\|_2$, sample $\rho_s \sim \text{LogUniform}(0.8\gamma, 1.25\gamma)$, and set $\ell_s^- = \rho_s\ell_s$. The perturbed relative translation is $\Delta\hat{\mathbf{c}}_s^- = \ell_s^-\mathbf{d}_s^-$, where $\mathbf{d}_s^-$ is the resulting unit direction for a nonzero step and remains unchanged for magnitude-only perturbations. We reconstruct the translation trajectory and apply the normalization in Eq. (6).

For moving-camera videos, rotation perturbation is performed as follows:

1. We select k anchor frames following the same strategy as translation, compute the frame-relative rotations $\Delta\mathbf{R}_s = \mathbf{R}_s\mathbf{R}_{s-1}^\top$ at the selected anchors, and convert them to yaw, pitch, and roll angles.
2. **Jitter.** For each anchor frame, we perturb yaw, pitch, and roll independently. For example, $\theta_s^{\text{yaw},-} = \theta_s^{\text{yaw}} + \delta_{\text{yaw}}$, where $\delta_{\text{yaw}} = \xi A$. $\xi \in \{-1, +1\}$ is sampled with equal probability, and $A \sim \text{Uniform}(l, h)$ with $l = 0.2\gamma$ and $h = 2.2\gamma$. The same operation is applied to pitch and roll. All angles are measured in degrees.
3. **Pattern.** We define four temporal patterns: increase, decrease, increase-then-decrease, and decrease-then-increase. We first sample $A \sim \text{Uniform}(l, h)$ as in jitter and convert it to radians as $\alpha = \pi A/180$ to avoid overly large exponents. For the increase pattern, we set $\theta_s^- = \theta_s \exp(\alpha)$. For the decrease pattern, we set $\theta_s^- = \theta_s \exp(-\alpha)$. For the increase-then-decrease and decrease-then-increase patterns, we randomly select a switching anchor and flip the sign of α after that anchor.

We choose jitter or one of the four patterns with equal probability (20%). Here, increase and decrease denote amplification and attenuation of relative angles, not necessarily monotonic absolute camera rotation. The perturbed angles are converted back to relative rotations. Consistent with $\Delta\mathbf{R}_s = \mathbf{R}_s\mathbf{R}_{s-1}^\top$, we reconstruct $\mathbf{R}_s^- = \Delta\mathbf{R}_s^-\mathbf{R}_{s-1}^-$ from $\mathbf{R}_1^- = \mathbf{I}$. The factor $\gamma = \sqrt{48/k}$

increases the perturbation strength when fewer anchors are selected, which is useful because camera errors can accumulate over time.

For static-camera videos, the relative translations are zero and the relative rotations are identity matrices, so multiplicative motion perturbations have no effect. Therefore, for translation, we add small random frame-relative translations to the selected anchor frames, with random directions and magnitudes sampled from $\mathrm{Uniform}(0.0015\gamma, 0.008\gamma)$. For rotation, we only apply additive jitter to yaw, pitch, and roll, using $l = 0.005\gamma$ and $h = 0.5\gamma$. This produces nonzero motion that violates the static-camera condition.

E Default Intrinsics

The normalized camera intrinsics are represented as $\mathbf{k} = [f_x/W, f_y/H, c_x/W, c_y/H]$, where $f_x = W/(2\tan(\mathrm{FOV}_x/2))$ and $f_y = H/(2\tan(\mathrm{FOV}_y/2))$. We use the default intrinsic values $[0.8333, 1.25, 0.5, 0.5]$ by assuming the optical center is at the image center and using a horizontal field of view of approximately 61.9° and a vertical field of view of approximately 43.6°.

F Additional Training Details

Our models are trained with 16 NVIDIA H100 GPUs under the FSDP strategy. The SFT-only runs use 20,000 steps for CogVideoX-2B and 10,000 steps for CogVideoX-5B. The TPO run starts from the 16,000-step 2B SFT checkpoint and adds 4,000 TPO steps, for 20,000 steps in total. For SFT, the learning rate is set to 10^{-4} with 2,000 steps of warm-up followed by cosine decay. We sample the magnitude variable a in Eq. 7 from a log-uniform distribution over $[0.05, 20]$. For TPO, we follow the learning-rate and preference-strength settings of DiffusionDPO (Wallace et al., 2024). The warm-up lasts 1,000 steps and is followed by cosine decay, and we fix $a = 1$ (equivalently, $\log a = 0$) during TPO.

Variation across generated videos. For each of the 2,000 held-out trajectories, we generate four videos with Ours-2B+TPO. We form four evaluation rounds, each containing one video per trajectory, and compute the full-test-set mean Trans. and Rot. errors for each round. Across the four round-level means, the standard deviations are 0.00027 for Trans. and 0.00086 for Rot. These values measure generation variability for a fixed checkpoint and test split. They do not quantify uncertainty in the difference between Ours-2B and Ours-2B+TPO.

G Adaptation with Metric-Aligned Trajectories

Our core objective is to scale camera-conditioned training without expensive metric-scale alignment, whose accuracy also depends on estimated depth. When aligned trajectories are available, the resulting checkpoint can be adapted to metric-aligned conditioning. Starting from our CogVideoX-2B checkpoint after 20,000 SFT steps without metric alignment, we fine-tune for 2,000 additional steps on metric-aligned RealEstate10K (Zhou et al., 2018) trajectories, following AC3D's metric-aligned data preparation and evaluation protocol (Bahmani et al., 2025a). Under AC3D's normalized translation metric, Table 5 shows lower TransErr for the adapted model (0.198 versus 0.374), together with lower rotation error, FVD, and FID and a higher CLIP score. This supports adaptation to metric-aligned camera conditioning and improved trajectory following under AC3D's evaluation protocol. Because TransErr is normalized, this table does not measure absolute translation-distance accuracy. Fig. 4 separately evaluates the original model's average relative translation-strength response.

Table 5: Adaptation with metric-aligned RealEstate10K trajectories under AC3D's evaluation protocol. Trans. is the normalized translation error and does not measure absolute displacement. Best results are **boldfaced**.

Method	Trans.↓	Rot.↓	FVD↓	FID↓	CLIP Score↑
AC3D	0.374	0.039	38.20	1.27	0.2862
Ours (metric-aligned fine-tuning)	**0.198**	**0.025**	**31.47**	**1.09**	**0.2912**



Figure 5: Additional stationary-camera comparisons with AC3D (Bahmani et al., 2025a). Red prompt text highlights content missing from the AC3D example, and red arrows mark apparent camera drift. Each row shows eight uniformly sampled frames from a 49-frame video. Our method preserves more stable camera framing in these examples.

H Additional Qualitative Evaluation

Figure 5 compares stationary-camera generation for four prompts. Red arrows highlight changes in image framing in the AC3D examples. The first example shows the car shifting relative to the image boundary. In the second, the indicated tree moves partly out of view. In the third, the mirror boundary becomes visible near the end. The fourth shows an apparent move toward the children, with the milk cup serving as a reference. Our method preserves more stable camera framing in these examples.

Figure 6 presents additional moving-camera comparisons. The generated camera motion is more similar to the ground-truth motion with our method in these examples.

I Additional Related Work: GEN3C and Uni3C

GEN3C (Ren et al., 2025) builds a 3D cache from depth estimates of input images or video frames and conditions video diffusion on views rendered along target camera trajectories. Uni3C (Cao et al., 2025) derives point clouds from a reference image and combines their rendered views with Plücker-ray embeddings through PCDController. It also aligns scene geometry with SMPL-X human representations for joint camera and human-motion control. Both methods emphasize explicit scene priors for consistent generation, particularly in I2V settings. Uni3C also supports a T2V backbone while retaining scene-derived guidance.



Figure 6: Additional moving-camera comparisons with AC3D (Bahmani et al., 2025a) and ground truth. Each row shows eight uniformly sampled frames from a 49-frame video. Our method follows the ground-truth camera motion more closely in these examples.

Our method differs substantially in its conditioning requirements and optimization target. It normalizes camera translation by total path length, encodes motion scale separately, and uses TPO to improve trajectory fidelity by perturbing camera conditions while keeping the video fixed. These components support scalable training without metric-scale alignment and improve camera adherence without requiring reference-image point clouds or rendered scene geometry as generation conditions. In this sense, our representation and training objective provide a more general formulation that could also complement scene priors. Our experiments cover T2V with CogVideoX-2B and CogVideoX-5B. The I2V extension is still training, and matched comparisons with GEN3C and Uni3C remain future work.

Model scale and computational constraints. The released GEN3C-Cosmos model uses a 7B backbone (Ren et al., 2025), while Uni3C's primary configuration uses a frozen 14B Wan2.1-I2V backbone with a 0.95B trainable PCDController (Cao et al., 2025). Adapting and training our method on these larger backbones exceeds our available compute, which currently limits our I2V extension to CogVideoX-5B-I2V.

J Limitations and Future Work

At the time of submission, our image-to-video (I2V) extension is still under training. Limited computational resources restrict us to CogVideoX-5B-I2V (Yang et al., 2024c), with an estimated training duration of at least one month on our available hardware. Our current budget does not support 10B+ I2V backbones, such as Wan2.2-I2V-A14B (Wan-AI, 2025) or the 14B backbone used by Uni3C (Cao et al., 2025). Scaling to larger I2V backbones and evaluating against GEN3C (Ren et al., 2025) and Uni3C remain future work.

The proposed decoupling removes the need for metric calibration during large-scale training, but it does not resolve absolute scene scale. The aggregate scale response is monotonic over the reported settings but not linearly calibrated. We report point estimates without confidence intervals or significance tests, so small differences between training schedules should be interpreted cautiously. Relative translation-strength control is appropriate for open-domain generation. When metric-scale control is required, fine-tuning on metric-scaled trajectories provides an adaptation path, as demonstrated in Appendix G. Future work may investigate weak scale supervision to further improve

calibration. Joint modeling of camera magnitude and scene depth, or combining TPO with human preference signals that capture both trajectory fidelity and video quality, are also attractive.

K Broader Impact

Improved camera control can make video generation systems more useful for filmmaking, education, simulation, and creative prototyping. At the same time, more controllable video models can also make synthetic footage easier to tailor for deceptive or manipulative content. Future deployment of video generation is encouraged to include appropriate safeguards for misuse.

L Per-Sample Monotonicity of Translation-Strength Control

We evaluate the CogVideoX-2B checkpoint after 20,000 SFT steps on the same 2,000 translation-only test poses used in Fig. 4. For each sample, we fix the prompt, unit trajectory, and generation seed across $a \in \{1, 5, 10\}$. We use the same MegaSAM trajectory estimation and per-video metric-scale alignment as in Fig. 4, with alignment performed only for diagnostic evaluation. Let $L_i(a)$ denote the total path length of the aligned generated trajectory for sample i at scale parameter a. We count a sample as strictly monotonic if $L_i(1) < L_i(5) < L_i(10)$. Under this criterion, 92% of the test samples exhibit strictly increasing estimated translation magnitude. This result supports per-sample relative translation-strength control over the evaluated settings. It does not imply a linear response or calibrated absolute displacement.