GENERATING PARALLEL WORLDS OF A FLIGHT

Anonymous authors
Paper under double-blind review

ABSTRACT

We study parallel worlds of a flight: visually distinct videos that faithfully reproduce a prescribed trajectory's shape, orientation, and temporal progression. Pre-trained video models provide rich visual priors for these worlds, but can produce convincing camera motion that drifts from the prescribed flight. We present *WorldSlider*, a generative pipeline that addresses this control challenge. Existing camera controls require models to translate geometric instructions into visual motion, while source-video references also carry the original world's appearance and geometry. WorldSlider's FlightGrid representation bridges this gap by automatically rendering calibrated camera poses as an appearance-free control video of the prescribed trajectory and its surrounding flight corridor. We distill the generator to four inference steps and refine flight alignment through reinforcement learning with feedback from a camera-motion verifier. WorldSlider improves trajectory fidelity over existing methods on standard trajectories and diverse flights, with qualitative demonstrations of transfer between simulated and real inputs. Its task-consistent visual diversification provides training data for visuomotor policies, substantially improving task success in simulation. Code and synchronized supplementary videos are available on our anonymous project page.

> "What if you could slide into a thousand different worlds? . . . you're the same person, but everything else is different?"
>
> — *Quinn Mallory, Sliders*

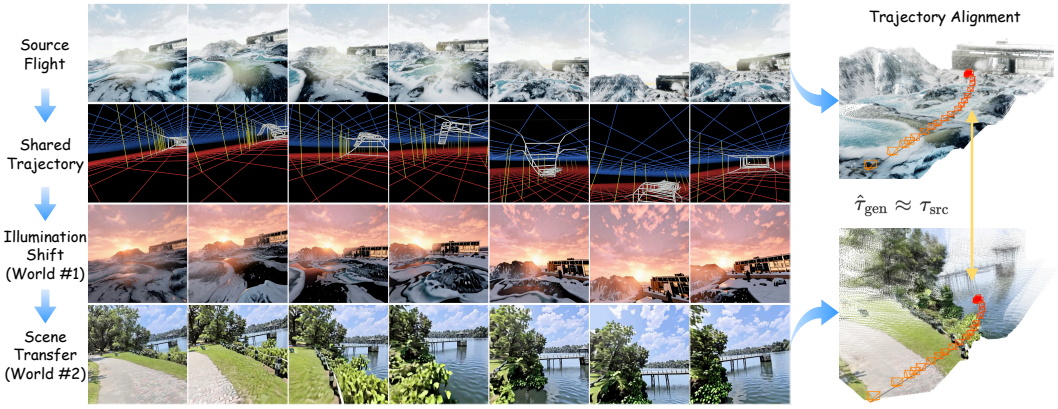

Figure 1: **Parallel worlds of a flight.** Trajectory shape and orientation are retained under new illumination (World #1) or scene (World #2). Rows: source, FlightGrid, and two worlds at matched times. Right: recovered trajectory alignment.

1 INTRODUCTION

Watching a moving-camera video invites a counterfactual: how would the same motion look in another world? In Figure 1, a flight through a polar research station unfolds beneath a rose-colored sky, above a tropical reservoir, or in a world that never existed. Geometry, illumination, and semantics

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.

may change while the camera advances, banks, and turns with the same rhythm. We study these *parallel worlds of a flight*: given a prescribed camera trajectory, environment prompt, and optional first frame, synthesize visually distinct videos matching its trajectory shape, orientation, and temporal progression.

Conventionally, producing such matched videos would require building each target world in a simulator and rendering it along the prescribed camera trajectory (Shah et al., 2018). This requires substantial environment-specific modeling, assets, and engineering, making it costly to scale to diverse worlds. Pre-trained generative models offer an alternative through text-guided video editing (Wu et al., 2023; Qi et al., 2023; Geyer et al., 2024), which can change illumination and semantics, such as the time of day or object categories, while retaining the source video's visible motion. However, these methods typically preserve much of the source video's spatial layout (Qi et al., 2023; Geyer et al., 2024), limiting how freely scene geometry can change. Moreover, their editing objectives do not explicitly enforce a prescribed camera trajectory (Wu et al., 2023; Qi et al., 2023; Geyer et al., 2024). We therefore seek to generate the world around a flight rather than transform the original video, using a pre-trained video model as a generative prior over possible worlds.

A natural starting point is camera-controlled image-to-video generation, which uses an initial frame and a sequence of camera poses to generate a video with the specified camera motion (Zheng et al., 2024; He et al., 2025b; Li et al., 2025b). Parallel worlds require faithful trajectory reproduction, not merely plausible camera motion. Existing models such as Wan2.1-Camera (aigc-apps, 2026; Team Wan et al., 2025) and RealCam-I2V (Li et al., 2025b) often capture only the coarse motion trend and drift from the prescribed poses when translation and rotation are coupled, whereas WorldSlider remains aligned (Figure 4). We hypothesize that a pose–video modality gap contributes to this difficulty: poses encode motion precisely, but translating them into coherent visual motion remains difficult. We introduce *FlightGrid* as a visual control interface that renders trajectory geometry and timing without source-scene appearance. A time-indexed trace and a metric spatial scaffold convert calibrated camera poses into frame-aligned motion and parallax. This exposes the prescribed flight in the model's native video modality while leaving the surrounding world's appearance and geometry to the generative conditions.

Explicit motion conditioning alone does not ensure trajectory fidelity: prior work shows that camera-conditioned generators can still deviate from their prescribed trajectories (Wang et al., 2026; Li et al., 2026). For parallel-world generation, sharing a trajectory condition therefore does not necessarily yield the same camera motion across worlds. We address residual errors through reinforcement learning (RL) for flight alignment: geometric feedback from trajectories recovered by Depth Anything 3 (DA3) (Lin et al., 2026) reweights the distribution-matching update of the four-step generator (Lu et al., 2026; Jiang et al., 2026). This makes trajectory fidelity an explicit training objective.

WorldSlider uses FlightGrid for trajectory control, few-step distillation for efficiency, and geometric feedback for alignment across visually distinct worlds. With only 26 hours of flight footage for adaptation, WorldSlider achieves accurate control on held-out complex simulated flights and generalizes to unseen canonical camera trajectories. On diverse flights, WorldSlider reduces rotation and translation errors by 67% and 73%, respectively, relative to the best baseline for each metric, using only four inference steps (Table 1). We also visualize cross-domain flight transfer and alternative flights branching from a shared observation. Beyond visual generation, we explore its use for simulated visuomotor policy learning by generating visually diverse observations with trajectory-aligned action supervision.

Our contributions are:

1. We propose FlightGrid, which renders trajectory geometry and timing as appearance-free visual control for pretrained video models, enabling the same flight to guide generation across different worlds.
2. We reduce residual trajectory errors across visually distinct worlds by refining a four-step flight generator through reinforcement learning with geometric feedback.
3. We introduce WorldSlider, which combines FlightGrid, few-step distillation, and geometric feedback to generate parallel-world videos in four inference steps. We demonstrate qualitative cross-domain flight transfer, trajectory-conditioned branching, and visual diversification for simulated visuomotor policy learning.

2 Related Work

Video editing. Video foundation models (Blattmann et al., 2023; Yang et al., 2025; Kong et al., 2024; Team Wan et al., 2025) have made flexible editing possible. Source-guided methods preserve structure through one-shot tuning and attention injection (Wu et al., 2023; Qi et al., 2023; Liu et al., 2024), or through tuning-free feature propagation (Geyer et al., 2024; Ku et al., 2024). Unified interfaces combine several creation and editing tasks (Jiang et al., 2025). Camera re-rendering likewise remains tied to the source scene, changing viewpoint through source-video conditioning or point-cloud renders (Bai et al., 2025; Yu et al., 2025; Yan et al., 2025). WorldSlider instead supports both appearance shifts and scene replacement under the same prescribed flight.

Controllable video generation. Camera-control methods encode motion through extrinsics, Plücker rays, epipolar geometry, or positional encodings (Wang et al., 2024; He et al., 2025a; Zheng et al., 2024; Zhang et al., 2026). Later designs extend these controls across backbones, motion ranges, and metric scales (Bahmani et al., 2025b;a; He et al., 2025b; Li et al., 2025b; aigc-apps, 2026; Team Wan et al., 2025; Sun et al., 2025). Such pose signals separate motion from content but require models to translate analytic geometry into pixel-space motion. Video-to-video models also accept frame-aligned depth, edges, segmentation, blurred RGB, or rendered proxy videos (Jiang et al., 2025; NVIDIA et al., 2025; Liu et al., 2026; Chen et al., 2026). These conditions expose structure in the video domain; those derived from source footage can also retain its layout or appearance. FlightGrid renders prescribed flight geometry through a time-indexed trace and metric spatial scaffold.

Video generation alignment. Generative-model alignment uses offline preference learning and online optimization. Direct preference methods learn from paired samples to improve visual quality, semantics, motion, and physical plausibility (Liu et al., 2025b;c; Chen et al., 2025), while Flow-GRPO (Liu et al., 2025a) applies online policy gradients to flow matching models. Multi-step video rollouts are costly, and exploration after distillation is limited to the compressed student's distribution. Reward-weighted distribution matching combines alignment with few-step distillation (Lu et al., 2026; Jiang et al., 2026). Feed-forward 3D models enable geometric rewards (Keetha et al., 2026; Lin et al., 2026), recently applied to camera control and temporal consistency (Li et al., 2026; Yin et al., 2026). WorldSlider uses recovered flight geometry to reweight distribution matching for a four-step flight generator.

3 Method

WorldSlider generates video from a calibrated camera trajectory, an environment caption, and an optional first frame. FlightGrid renders the trajectory as visual control (Section 3.1); training proceeds through control adaptation, distribution matching, and reinforcement learning for flight alignment (Sections 3.2–3.4).

3.1 Flight Representation without Scene Appearance

Camera poses specify motion geometrically, but generators can deviate from them under coupled 6-DoF motion and nonuniform timing. FlightGrid renders the trajectory's temporal and spatial structure directly in the video domain.

FlightGrid comprises two complementary structures: a time-indexed flight trace that preserves the temporal parameterization of the trajectory, and a metric spatial scaffold that provides a persistent world reference (Figure 2). Let $\tau = \{(\mathbf{p}_t, R_t)\}_{t=1}^{T}$ denote a camera trajectory whose translations use a calibrated spatial scale. We construct its trace $\mathcal{C}(\tau) = \{(\ell_j, t_j)\}_{j=1}^{N}$ by sweeping an attitude-aligned rectangular cross-section along the complete trajectory, where ℓ_j is a line segment and t_j is its traversal time. Cross-sectional rings are placed at uniform time intervals Δt; their spatial separation encodes flight speed, and their orientation encodes camera attitude. At frame t, each segment is rendered with opacity

$$\alpha(t, t_j) = \begin{cases} 1, & |t - t_j| \le T_{\text{full}}, \\ \dfrac{T_{\text{fade}} - |t - t_j|}{T_{\text{fade}} - T_{\text{full}}}, & T_{\text{full}} < |t - t_j| < T_{\text{fade}}, \\ 0, & |t - t_j| \ge T_{\text{fade}}. \end{cases} \tag{1}$$

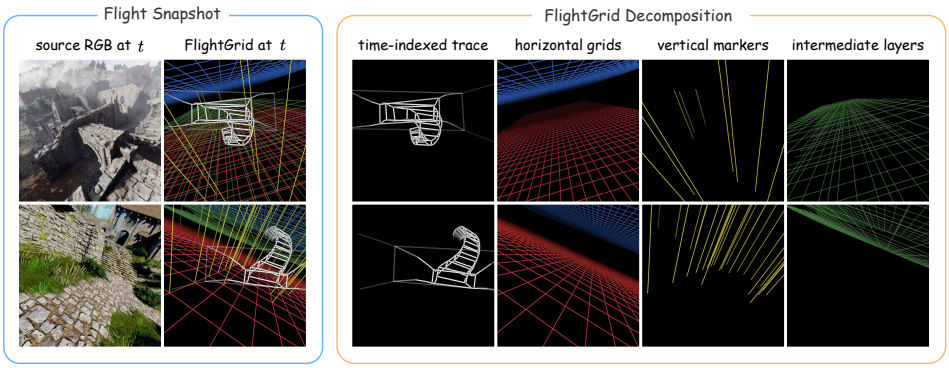

Figure 2: **Appearance-free FlightGrid representation.** Two vertical flights and their FlightGrids, decomposed into a white time-indexed trace, red/blue horizontal grids, yellow route markers, and green intermediate layers.

Rendering the time-stamped segments with these weights defines the frame-specific trace $\mathcal{C}_t(\tau)$. It provides frame-aligned cues for position, speed, attitude, and local corridor extent, while restricting visible geometry to a temporal neighborhood of the current frame. Because the underlying trace spans the complete flight, its construction does not assume that the viewing direction agrees with the direction of travel; the temporal restriction also prevents irrelevant segments from accumulating during revisits and loops.

The metric spatial scaffold $\mathcal{G}(\tau, \mathbf{g})$ complements the local trace with a world-fixed reference, where $\mathbf{g}$ denotes the gravity direction. It is constructed in gravity-aligned coordinates using metric distances: horizontal grids with one-meter spacing encode scale and gravity; route-side vertical markers provide nearby translational parallax; and intermediate horizontal layers are inserted according to the vertical span of the trajectory to retain local spatial references during ascent and descent. At each frame, the scaffold is combined with the time-indexed trace and projected from the prescribed camera pose,

$$F(\tau)_t = \Pi_K(\mathcal{G}(\tau, \mathbf{g}) \cup \mathcal{C}_t(\tau); \mathbf{p}_t, R_t)\,, \tag{2}$$

where Π_K denotes projection with camera intrinsics K. Both structures have fixed world coordinates, so their projected motion follows the prescribed camera motion; the temporal weights determine which trace segments remain visible. FlightGrid therefore depends only on camera poses, gravity, and camera intrinsics—not on source RGB pixels, scene depth, object boundaries, or semantic labels—leaving the surrounding world to be specified by the generative conditions.

3.2 Control Adaptation with FlightGrid

We inject FlightGrid through the pre-trained depth-control branch of Cosmos Transfer (NVIDIA et al., 2025). Given a target video $V^\star$ with camera trajectory τ, environment caption c, and optional first frame I_0, we render $F(\tau)$ and train the model under these conditions. Captions describe only the environment, leaving FlightGrid as the sole motion condition. Training uses 24.9 hours of TartanAir-V2 (Wang et al., 2020) and 1.1 hours of physical flight.

A frozen video VAE E maps the target video and FlightGrid to latents $z = E(V^\star)$ and $f = E(F(\tau))$. The ControlNet jointly processes f and noisy video tokens, injecting layer-wise residuals into the frozen video DiT. The ControlNet is optimized while the video DiT, VAE, and text encoder remain fixed. For $\epsilon \sim \mathcal{N}(0, I)$ and flow time $t \in [0, 1]$, we optimize the model with the standard conditional rectified-flow objective

$$z_t = (1-t)z + t\epsilon, \qquad u_t = \epsilon - z, \qquad \mathcal{L}_{\mathrm{SFT}}(\phi) = \mathbb{E}_{V^\star,\epsilon,t}\left[\left\|v_\phi(z_t, t \mid y) - u_t\right\|_2^2\right], \tag{3}$$

where $y = (f, c, I_0)$. The resulting multi-step generator follows FlightGrid while retaining the text- and image-conditioned visual prior.

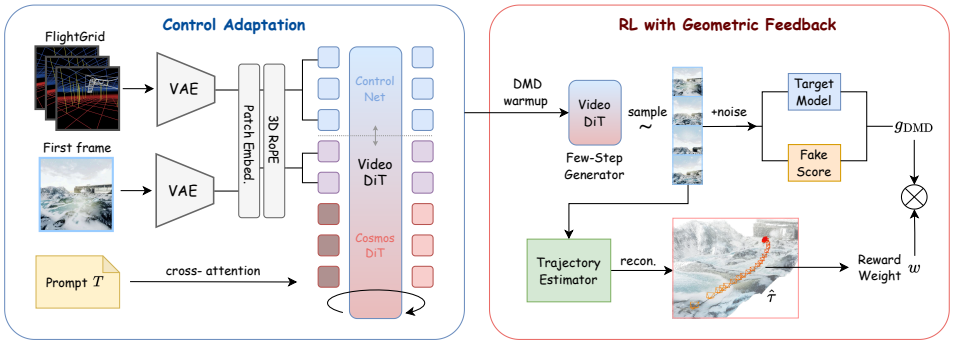

Figure 3: **WorldSlider training.** Control adaptation trains the control branch with the video DiT frozen; DMD distills the generator to four steps, then rewards from recovered trajectories refine flight alignment.

3.3 Distribution Matching for Efficient Sampling

Native multi-step sampling is costly for large-scale generation and rollout-based alignment. We therefore use DMD (Yin et al., 2024b) to distill the adapted model into a few-step generator. The teacher is frozen, while the student G_θ and fake-score model S_ψ are initialized from its weights. Following the regression-free multi-step scheme of DMD2 (Yin et al., 2024a) without its adversarial objective, we alternate student and fake-score updates using the inference-time sampler. For $\hat{z} = G_\theta(\epsilon; y)$, the TrigFlow parameterization (Lu & Song, 2025) gives

$$z_s = \cos(s)\hat{z} + \sin(s)\sigma_{\text{d}}\xi, \qquad s \in [0, \pi/2], \quad \xi \sim \mathcal{N}(0, I), \tag{4}$$

where σ_{d} is the latent data scale. This bounded parameterization interpolates from the student sample at $s = 0$ to scaled Gaussian noise at $s = \pi/2$. At a sampled s, the frozen teacher and S_ψ denoise the same z_s into $\hat{z}_{\text{T}}$ and $\hat{z}_\psi$, respectively. The fake-score model is trained with the standard conditional flow-matching objective on perturbed student samples, thereby estimating the current student distribution, while the frozen teacher provides the target distribution. Their normalized difference yields the DMD update

$$g_{\text{DMD}} = \text{sg}\left[\frac{\hat{z}_\psi - \hat{z}_{\text{T}}}{\delta(\hat{z}, \hat{z}_{\text{T}})}\right], \qquad \nabla_\theta \mathcal{L}_{\text{DMD}} = \mathbb{E}\left[g_{\text{DMD}}^{\text{T}} \frac{\partial \hat{z}}{\partial \theta}\right], \tag{5}$$

where sg denotes stop-gradient and δ is the standard per-sample gradient normalizer. Classifier-free guidance is applied only to the frozen teacher, so the student absorbs its effect without extra guidance evaluations at inference. This stage distills the adapted teacher into a four-step generator.

3.4 Reinforcement Learning for Flight Alignment

The distilled four-step generator can still deviate from the prescribed trajectory. Its reduced sampling cost makes optimization from generated rollouts practical. We treat the generator as a policy over videos conditioned on the prescribed flight. Each sampled video is a policy rollout whose recovered trajectory provides geometric feedback. Reward-weighted DMD updates give greater influence to better-aligned rollouts while preserving the original score target.

Geometry reward from trajectory recovery. We use the foundation depth model Depth Anything 3 (DA3) (Lin et al., 2026) as the trajectory estimator $\mathcal{A}$. Let $\hat{\tau} = \{(\hat{\mathbf{p}}_t, \hat{R}_t)\}_{t=1}^{T} = \mathcal{A}(\hat{V})$ be the camera trajectory recovered from a generated video. Over short and long temporal offsets $h \in \mathcal{H}$, we compare the prescribed and recovered segment translations in their local camera frames, $\mathbf{d}_{t,h} = R_t^{\text{T}}(\mathbf{p}_{t+h} - \mathbf{p}_t)$ and $\hat{\mathbf{d}}_{t,h} = \hat{R}_t^{\text{T}}(\hat{\mathbf{p}}_{t+h} - \hat{\mathbf{p}}_t)$. To absorb the estimator's coordinate convention, the gauge Q_τ is estimated once from the corresponding source flight and then held fixed. For each

generated video, we fit only a positive global scale:

$$s^\star = \arg\min_{s>0} \sum_{h\in\mathcal{H}} \sum_t \left\| sQ_\tau \hat{\mathbf{d}}_{t,h} - \mathbf{d}_{t,h} \right\|_2^2,$$
$$e_{\mathrm{seg}}(\hat{\tau}, \tau) = \frac{\sum_{h\in\mathcal{H}} \sum_t \left\| s^\star Q_\tau \hat{\mathbf{d}}_{t,h} - \mathbf{d}_{t,h} \right\|_2}{\sum_{h\in\mathcal{H}} \sum_t \left\| \mathbf{d}_{t,h} \right\|_2}. \tag{6}$$

Fixing Q_τ prevents per-video rotational alignment from concealing translation errors, while fitting $s^\star$ accounts for the unreliable global scale of monocular reconstruction. We define the geometry reward as

$$r_{\mathrm{geo}}(\hat{V}, \tau) = \exp\left(-\frac{[e_{\mathrm{seg}}(\hat{\tau}, \tau) - b_\tau]_+}{\kappa} \right) h_{\mathrm{dir}}\, h_{\mathrm{rot}}\, h_{\mathrm{mot}}, \tag{7}$$

where $[x]_+ = \max(x, 0)$ and b_τ is the error obtained by applying the same verifier to the source flight. This source-specific floor accounts for trajectory-dependent reconstruction noise. The bounded gates h_{dir}, h_{rot}, and h_{mot} suppress incorrect translation direction, relative rotation, and insufficient visual motion. If the source flight cannot be recovered reliably, the verifier abstains and the sample receives an unweighted update. Appendix C gives the gate definitions, abstention criteria, and additional DMD details.

Policy optimization with geometry rewards. Let μ_k and σ_k denote the running reward mean and standard deviation before iteration k. We use a clipped exponential weight and gradually introduce it through λ_k:

$$w_k(r) = (1 - \lambda_k) + \lambda_k \,\mathrm{clip}\left(\exp\left(\frac{r - \mu_k}{\beta \max(\sigma_k, \sigma_{\min})} \right), w_{\min}, w_{\max} \right),$$
$$\nabla_\theta \mathcal{L}_{\mathrm{RL}} = \mathbb{E}\left[w_k\Big(r_{\mathrm{geo}}(\hat{V}, \tau) \Big)\, g_{\mathrm{DMD}}^{\mathsf{T}} \frac{\partial \hat{z}}{\partial \theta} \right]. \tag{8}$$

The reward and weight are treated as stop-gradient quantities. Thus, geometric feedback changes each sample's relative influence without altering its DMD score difference target, while clipping and continuation limit sensitivity to verifier noise. Setting $\lambda_k = 0$, or assigning $w_k = 1$ to an abstained sample, exactly recovers the base DMD update.

4 Experiments

We evaluate trajectory fidelity on standard and diverse flights, demonstrate parallel-world generation and alternative flights, and test visual diversification for simulated flight-policy training. Ablations examine FlightGrid, reinforcement learning, and verifier capacity.

4.1 Flight Fidelity and Generalization

Evaluation protocol. WorldSlider is trained on 26.0 hours of flight clips: 24.9 hours from TartanAir-V2 (Wang et al., 2020) and 1.1 hours from our physical platform. We evaluate two fixed sets of 50 trajectories. Standard Trajectories are smooth, canonical RealEstate10K (Zhou et al., 2018) paths absent from training and therefore zero-shot. Diverse Flights use TartanAir-V2 paths with no overlap in environments, source flights, or clips; they stress large rotation and vertical displacement, direction changes, revisitation, and mismatched viewing and travel directions. Downstream application trajectories are also unseen.

Baselines and metrics. We compare five public camera-control models spanning 1.3B–5B parameters: MotionCtrl (Wang et al., 2024), CameraCtrl (He et al., 2025a), RealCam-I2V (Li et al., 2025b), Wan2.1-Camera (aigc-apps, 2026; Team Wan et al., 2025), and HY-WorldPlay (Sun et al., 2025). Table 1 benchmarks released systems in their native settings; their controls span pose tokens, Plücker rays, and PRoPE. We report geodesic rotation error, globally scale-aligned translation error, Q-Align (Wu et al., 2024), and the number of function evaluations (NFE). ViPE (Huang et al., 2025), an uncalibrated pose estimator for unconstrained videos, is used only for evaluation, whereas DA3

Table 1: **Flight fidelity on standard and diverse trajectories.** ViPE errors: mean ± 95% CI; rotation in degrees, translation scale-aligned and path-normalized. Q-Align averages both splits and seeds. Yellow marks best/second best.

			Standard trajectories		Diverse flights		
Method	Params.	NFE	Rot. ↓	Trans. ↓	Rot. ↓	Trans. ↓	Quality ↑
MotionCtrl	2B	25	11.28±2.07	0.234±0.019	27.04±1.47	0.441±0.019	3.20
CameraCtrl	2B	25	5.67±1.53	0.099±0.015	26.72±1.61	0.471±0.021	3.15
RealCam-I2V	1.4B	25	7.62±1.48	0.132±0.015	24.37±1.47	0.428±0.022	2.76
Wan2.1-Camera	1.3B	50	5.31±1.08	0.097±0.015	26.49±1.56	0.410±0.030	3.22
HY-WorldPlay	5B	4	4.11±0.61	0.152±0.013	19.93±1.12	0.378±0.024	3.53
WorldSlider	2B	4	2.52±0.65	0.079±0.017	6.65±0.63	0.101±0.008	3.45

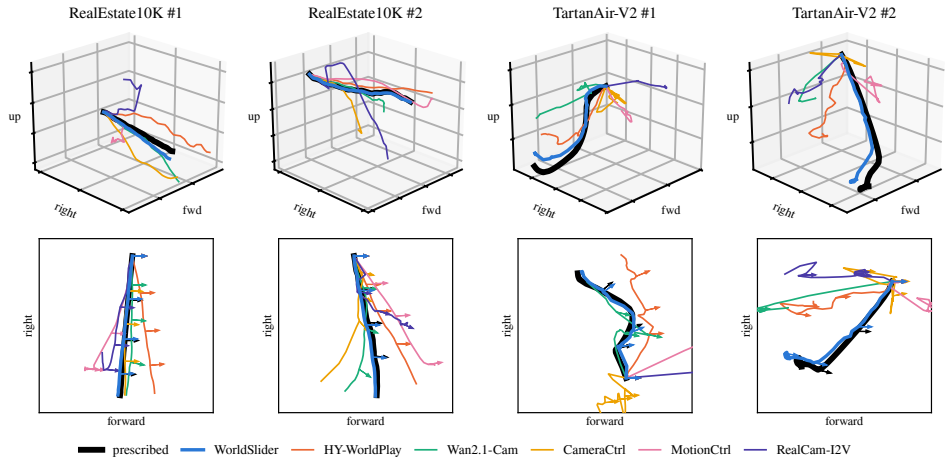

Figure 4: **Trajectory alignment on standard and diverse flights.** Prescribed trajectories (black) and ViPE recoveries from WorldSlider and five baselines. Top: 3D paths. Bottom: top views with orientation arrows. Closer overlap indicates higher flight fidelity.

supplies training feedback. Appendix C.5 details the protocol and native settings. Appendix C.6 compares Plücker and PRoPE with FlightGrid at the 35-NFE adaptation stage, then isolates plain DMD from geometric feedback at four NFEs with equal additional training. Unless stated otherwise, WorldSlider denotes the final four-step model.

Quantitative results. WorldSlider achieves the lowest error on all four trajectory measures. On zero-shot Standard Trajectories, it obtains $2.52°$ rotation error and 0.079 translation error. The best baseline differs by metric: HY-WorldPlay reaches $4.11°$ rotation error, while Wan2.1-Camera reaches 0.097 translation error; WorldSlider reduces these errors by 39% and 19%, respectively. On Diverse Flights, baseline errors rise to $19.93°$–$27.04°$ and 0.378–0.471. Relative to HY-WorldPlay, the best baseline on both metrics, WorldSlider reduces rotation error from $19.93°$ to $6.65°$ and translation error from 0.378 to 0.101, reductions of 67% and 73%, respectively. WorldSlider's Q-Align of 3.45 is second only to the 5B HY-WorldPlay at 3.53, despite using a 2B backbone with the same four network evaluations. The gains span complex flights and unseen canonical trajectories; matched controls and component ablations examine the visual interface (Section 4.3).

Qualitative results. The trajectory overlays in Figure 4 illustrate these differences on two standard and two diverse flights: accurate recovery follows the prescribed black curve in both position and orientation. The project page provides synchronized videos of these examples, with sampled frames in Appendix A.1. In these examples, RealCam-I2V drifts in position, HY-WorldPlay rotates with little translation, and Wan2.1-Camera attempts large motion but loses coherence on a diverse flight. WorldSlider follows both path and orientation through large altitude changes, consistent with the ViPE statistics.

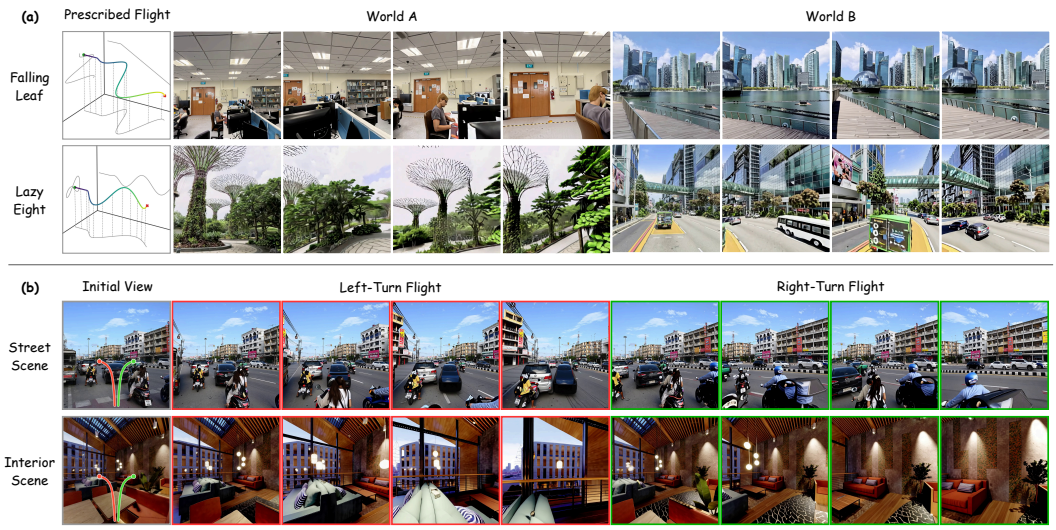

Figure 5: **Parallel worlds and alternative flights.** (a) Each prescribed flight in two visually distinct worlds. (b) Two alternative flights from one initial view. Columns show synchronized times.

4.2 Parallel Worlds and Applications

Parallel worlds and alternative flights. Figure 5(a) shows falling-leaf and lazy-eight flights in two visually distinct worlds. These out-of-distribution trajectories combine large spatial excursions and repeated direction reversals; generations remain coherent while following their shape, orientation, and timing. Across 12 trajectories and 20 worlds each, ViPE rotation and translation errors average 3.42° and 0.0762 (Table 5). Changing FlightGrid with the initial observation fixed produces the left- and right-turn futures in Figure 5(b), suggesting its use for alternative rollouts for world models.

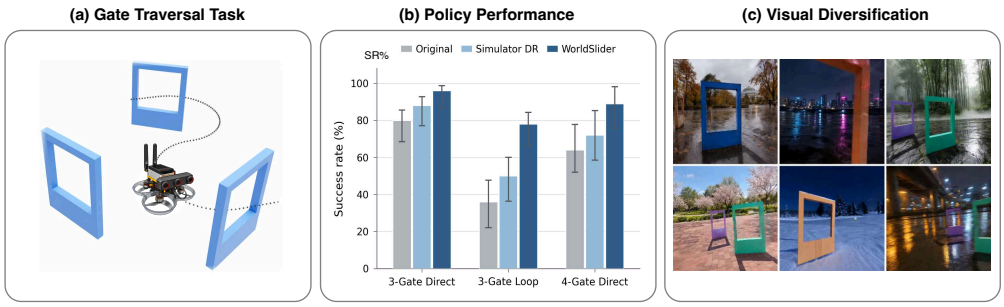

Figure 6: **Visual diversification for gate traversal.** (a) Three gate courses. (b) Success across five policies per condition and 20 rollouts per policy; error bars are 95% confidence intervals across policies. (c) Task-consistent training worlds from the same flight.

Visual diversification for flight policies. Simulator domain randomization varies illumination, gate appearance, and background (Loquercio et al., 2020). WorldSlider generates task-consistent environments around a fixed flight without manually constructing each world: *visual diversification*. All gates are visible in the initial frame; we edit its appearance, retain their layout, and pair it with the original FlightGrid. Appendix A.3 shows the original appearance and ten generated worlds across all three tasks.

We collect 20 expert trajectories per Isaac Sim course: three-gate direct, three-gate loop, and four-gate direct. An ACT policy (Zhao et al., 2023) predicts short waypoint sequences from egocentric RGB. Architecture, labels, and training budget are fixed. Repeating each demonstration ten times or providing ten simulator-randomized or WorldSlider variants yields 200 training sequences per course. Variants are used only for training. For each condition and course, five policies each undergo

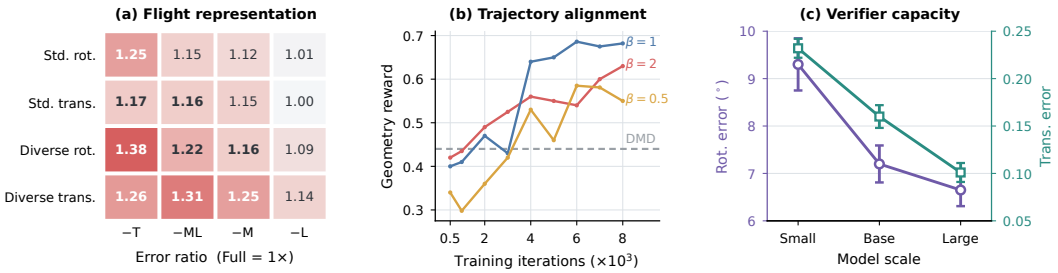

Figure 7: **Core ablations.** (a) Error ratios after component removal ($1\times$: full FlightGrid). (b) Validation reward across temperatures; first evaluation at 500 updates, dashed line: initial DMD reward. (c) Independent ViPE error across verifier capacities.

20 rollouts in the original simulator appearance, with only start poses randomized within the marked region. Success requires traversing every gate without collision. Gains over simulator randomization span all three courses and are largest on the loop (Figure 6), demonstrating start-pose generalization. Isaac Sim uses PX4 software-in-the-loop and our physical platform's ROS 2 interfaces and policy stack (Appendix C.4).

4.3 Ablation Study

Flight representation. Figure 7(a) evaluates FlightGrid at the SFT stage. We remove its time-indexed trace (T), route-side markers (M), and intermediate layers (L), while retaining the upper and lower horizontal grids in every variant. Removing the trace causes the largest rotation penalty, increasing error to $1.25\times$ on standard trajectories and $1.38\times$ on diverse flights. Removing the markers and intermediate layers together produces the largest diverse-flight translation penalty of $1.31\times$, compared with $1.25\times$ when only the markers are removed. Intermediate layers have little effect on standard trajectories with limited vertical excursion, but complement markers during the large altitude changes of diverse flights.

Control initialization. Initialized from pretrained base DiT weights, FlightGrid reaches 10.1° rotation error and 0.230 translation error on Diverse Flights, outperforming both matched analytic controls at 35 NFEs (Appendices C.3 and C.6). Depth-control initialization further improves these to 8.9° and 0.180. Together with the component ablations, this supports the visual trajectory interface: its advantage is not solely attributable to depth-control pretraining.

Reinforcement learning for flight alignment. Figure 7(b) compares RL temperatures from the same four-step DMD checkpoint, whose reward is 0.44 on a validation set excluded from RL. With DA3 fixed, $\beta = 1$ reaches approximately 0.70, the highest sustained reward. We adopt this setting: $\beta = 0.5$ gives sharper, less stable updates, while $\beta = 2$ improves more smoothly but slowly.

Validation with an independent trajectory estimator. Figure 7(c) evaluates models trained with DA3 rewards using ViPE, which is never used for optimization. Increasing DA3 capacity from Small to Large reduces rotation error from 9.30° to 6.65° and translation error from 0.232 to 0.101. Monotonic ViPE gains support the benefit of stronger geometric feedback beyond the training verifier's reward.

5 Conclusion

FlightGrid renders trajectory geometry and timing as appearance-free visual control for pretrained video models. Matched comparisons, initialization studies, and component ablations support this interface for complex camera motion. Adapted on 26 hours of flight footage, WorldSlider uses distillation and geometric feedback to follow held-out complex flights and unseen canonical trajectories in four inference steps. Cross-world evaluation quantifies trajectory fidelity, complemented by qualitative transfer between simulated and real inputs. Its visual diversification also preserves task structure for simulated flight-policy training.

AI USE STATEMENT

In this work, we used generative AI tools only for editing the text (grammar, spelling, and word choice). We did not use generative AI tools for research ideation, for writing or verifying code, for generating or analyzing experimental results, or for producing figures, and the remaining required-disclosure tasks are not applicable to this work. We have reviewed all AI-assisted edits and take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

To support reproduction, we release the FlightGrid renderer, training and inference code, and geometric reward implementation on our anonymous project page, with data construction, training configurations, and evaluation protocols detailed in Appendices B and C.

REFERENCES

aigc-apps. VideoX-Fun: A video generation pipeline for diffusion transformer. GitHub software repository, 2026. URL https://github.com/aigc-apps/VideoX-Fun.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22875–22889, 2025a.

Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, David B. Lindell, and Sergey Tulyakov. VD3D: Taming large video diffusion transformers for 3D camera control. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a6ef26da2f5b7a2d5d374dd13464c781-Abstract-Conference.html.

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*, pp. 14834–14844, 2025.

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

Harold Haodong Chen, Haojian Huang, Qifeng Chen, Harry Yang, and Ser-Nam Lim. Hierarchical fine-grained preference optimization for physically plausible video generation. In *Advances in Neural Information Processing Systems*, 2025.

Yiwen Chen, Guosheng Lin, and Chi Zhang. Code World Model: Coding agent as world brain. *arXiv preprint arXiv:2608.25927*, 2026. URL https://arxiv.org/abs/2608.25927.

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. TokenFlow: Consistent diffusion features for consistent video editing. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/069274129275ec1c6511abb5f6d96a31-Abstract-Conference.html.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for video diffusion models. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f98fd73d59d8494489ea970747b91fe4-Abstract-Conference.html.

Hao He, Ceyuan Yang, Shanchuan Lin, Yinghao Xu, Meng Wei, Liangke Gui, Qi Zhao, Gordon Wetzstein, Lu Jiang, and Hongsheng Li. CameraCtrl II: Dynamic scene exploration via camera-controlled video diffusion models. In *IEEE/CVF International Conference on Computer Vision*, pp. 13416–13426, 2025b.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixé, and Sanja Fidler. ViPE: Video pose engine for 3D geometric perception. *arXiv preprint arXiv:2508.10934*, 2025. URL https://arxiv.org/abs/2508.10934.

Dengyang Jiang, Dongyang Liu, Zanyi Wang, Qilong Wu, Liuzhuozheng Li, Hengzhuang Li, Xin Jin, Changsheng Lu, Zhen Li, Mengmeng Wang, Steven Hoi, Peng Gao, and Harry Yang. Distribution matching distillation meets reinforcement learning. In *European Conference on Computer Vision*, pp. 281–299, 2026. doi: 10.1007/978-3-032-37550-6_16.

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*, pp. 17191–17202, 2025.

Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011. doi: 10.1177/0278364911406761.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *International Conference on 3D Vision*, 2026. URL https://arxiv.org/abs/2509.13414.

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.

Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. AnyV2V: A tuning-free framework for any video-to-video editing tasks. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=RFrJCkw2oa.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *Advances in Neural Information Processing Systems*, 2025a. doi: 10.52202/085713-0540.

Teng Li, Guangcong Zheng, Rui Jiang, Shuigen Zhan, Tao Wu, Yehao Lu, Yining Lin, Chuanyun Deng, Yepan Xiong, Min Chen, Lin Cheng, and Xi Li. RealCam-I2V: Real-world image-to-video generation with interactive complex camera control. In *IEEE/CVF International Conference on Computer Vision*, pp. 28785–28796, 2025b.

Zizun Li, Haoyu Guo, Runzhe Teng, Chunhua Shen, and Tong He. Geo-Align: Video generation alignment via metric geometry reward. *arXiv preprint arXiv:2605.23903*, 2026.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/e4cd50120b6d7e8daff1749d6bbaa889-Abstract-Conference.html.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In *Advances in Neural Information Processing Systems*, 2025a. doi: 10.52202/085713-1362.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, Xiaohong Liu, Fei Yang, Pengfei Wan, Di Zhang, Kun Gai, Yujiu Yang, and Wanli Ouyang. Improving video generation with human feedback. In *Advances in Neural Information Processing Systems*, 2025b. doi: 10.52202/085713-2750.

Jiwen Liu, Shujuan Li, Zhixue Fang, Xiaohan Li, Yan Zhou, Zijie Meng, Zhimin Zhang, Yawen Luo, Guoxin Zhang, Yu-Shen Liu, and Pengfei Wan. OmniDirector: General multi-shot camera cloning without cross-paired data. *arXiv preprint arXiv:2606.13432*, 2026.

Runtao Liu, Haoyu Wu, Ziqiang Zheng, Chen Wei, Yingqing He, Renjie Pi, and Qifeng Chen. VideoDPO: Omni-preference alignment for video diffusion generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8009–8019, 2025c.

Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-P2P: Video editing with cross-attention control. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8599–8608, 2024.

Antonio Loquercio, Elia Kaufmann, René Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. Deep drone racing: From simulation to reality with domain randomization. *IEEE Transactions on Robotics*, 36(1):1–14, 2020. doi: 10.1109/TRO.2019.2942989.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/7e9c2053258b1bdd32ff2654802cd594-Abstract-Conference.html.

Yunhong Lu, Yanhong Zeng, Haobo Li, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jiapeng Zhu, Hengyuan Cao, Zhipeng Zhang, Xing Zhu, Yujun Shen, and Min Zhang. Reward forcing: Efficient streaming video generation with rewarded distribution matching distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34385–34397, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Lu_Reward_Forcing_Efficient_Streaming_Video_Generation_with_Rewarded_Distribution_Matching_CVPR_2026_paper.html.

NVIDIA, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World simulation with video foundation models for physical AI. *arXiv preprint arXiv:2511.00062*, 2025. URL https://arxiv.org/abs/2511.00062.

Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. FateZero: Fusing attentions for zero-shot text-based video editing. In *IEEE/CVF International Conference on Computer Vision*, pp. 15932–15942, 2023.

Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. AirSim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, pp. 621–635, 2018. doi: 10.1007/978-3-319-67361-5_40.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. WorldPlay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025. URL https://arxiv.org/abs/2512.14614.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang,

Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir: A dataset to push the limits of visual SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4909–4916, 2020. doi: 10.1109/IROS45743.2020.9341801.

Zhaoqing Wang, Xiaobo Xia, Zhuolin Bie, Jinlin Liu, Dongdong Yu, Jia-Wang Bian, and Changhu Wang. Taming camera-controlled video generation with verifiable geometry reward. In *European Conference on Computer Vision*, pp. 625–642, 2026. doi: 10.1007/978-3-032-37020-4_34.

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*, pp. 1–11, 2024. doi: 10.1145/3641519.3657518.

Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, Qiong Yan, Xiongkuo Min, Guangtao Zhai, and Weisi Lin. Q-Align: Teaching LMMs for visual scoring via discrete text-defined levels. In *International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/wu24ah.html.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-A-Video: One-shot tuning of image diffusion models for text-to-video generation. In *IEEE/CVF International Conference on Computer Vision*, pp. 7623–7633, 2023.

Yunzhi Yan, Zhen Xu, Haotong Lin, Haian Jin, Haoyu Guo, Yida Wang, Kun Zhan, Xianpeng Lang, Hujun Bao, Xiaowei Zhou, and Sida Peng. StreetCrafter: Street view synthesis with controllable video diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 822–832, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yan_StreetCrafter_Street_View_Synthesis_with_Controllable_Video_Diffusion_Models_CVPR_2025_paper.html.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ce31378e9f41d8907e97dab172b6c559-Abstract-Conference.html.

Tengjiao Yin, Jinglei Shi, Heng Guo, and Xi Wang. VIGOR: VIdeo geometry-oriented reward for temporal generative alignment. *arXiv preprint arXiv:2603.16271*, 2026.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024b.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. TrajectoryCrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *IEEE/CVF International Conference on Computer Vision*, pp. 100–111, 2025.

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*, pp. 38027–38037,

2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhang_Unified_Camera_Positional_Encoding_for_Controlled_Video_Generation_CVPR_2026_paper.html.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.016.

Guangcong Zheng, Teng Li, Rui Jiang, Yehao Lu, Tao Wu, and Xi Li. CamI2V: Camera-controlled image-to-video diffusion model. *arXiv preprint arXiv:2410.15957*, 2024.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 37(4):65:1–65:12, 2018. doi: 10.1145/3197517.3201323.

AI USE STATEMENT

We used generative AI tools to assist with language polishing and improving the clarity of the manuscript. All research ideas, methods, experiments, analyses, and conclusions were developed and verified by the authors. We reviewed every suggested edit and take responsibility for the final content of this work.

Supplementary Material

Contents

A Additional Results **15**
A.1 Additional Flight Comparisons 15
A.2 Parallel Worlds 18
A.3 Visual Diversification 19
B Flight Dataset **20**
B.1 Corpus Construction 20
B.2 Visual and Motion Diversity 21
C Implementation and Evaluation Details **23**
C.1 FlightGrid Parameterization 23
C.2 Training Details 24
C.3 Control Adaptation Analysis 25
C.4 Drone Platform 25
C.5 Evaluation Protocol 26
C.6 Matched Controls 27
C.7 Human Preference 28
D Limitations and Future Work **28**

A ADDITIONAL RESULTS

Full synchronized videos for the flight comparisons, parallel worlds, and visual diversification results in this section are available on our anonymous project page:

https://github.com/parallelworldsanon/WorldSlider

A.1 ADDITIONAL FLIGHT COMPARISONS

Figures 8 and 9 provide synchronized generations corresponding to the trajectory comparisons in Figure 4. In each panel, the left column compares the prescribed trajectory (black) with trajectories recovered by ViPE; arrows in the top view mark the viewing direction along each path. The right matrix shows synchronized frames at six normalized times for every method.

Across these cases, WorldSlider most consistently reproduces both the prescribed path evolution and orientation. The baselines often capture the coarse motion intent but subsequently drift from the prescribed poses, with substantially larger departures on diverse flights. In these examples, HY-WorldPlay exhibits a distinct failure mode: its viewing direction can remain well aligned while the camera center translates very little, yielding primarily in-place rotation rather than the prescribed flight. Its high Q-Align score with a 5B backbone (Table 1) does not imply accurate trajectory reproduction. These examples show that assessing trajectory fidelity requires translation accuracy alongside visual quality and orientation alignment.

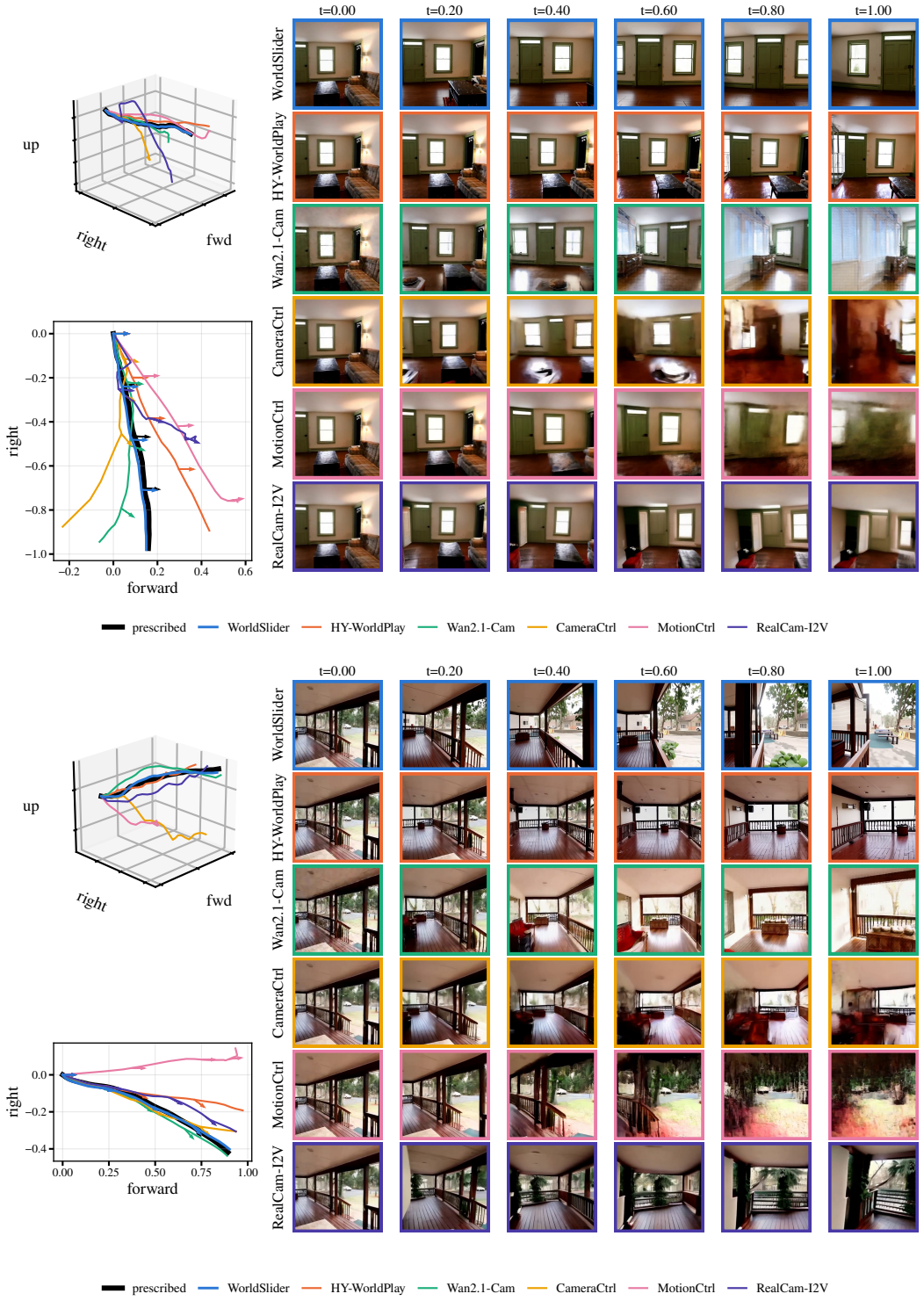

Figure 8: **Additional comparisons on standard trajectories.** Each panel presents one trajectory and the corresponding synchronized generations from all evaluated methods.

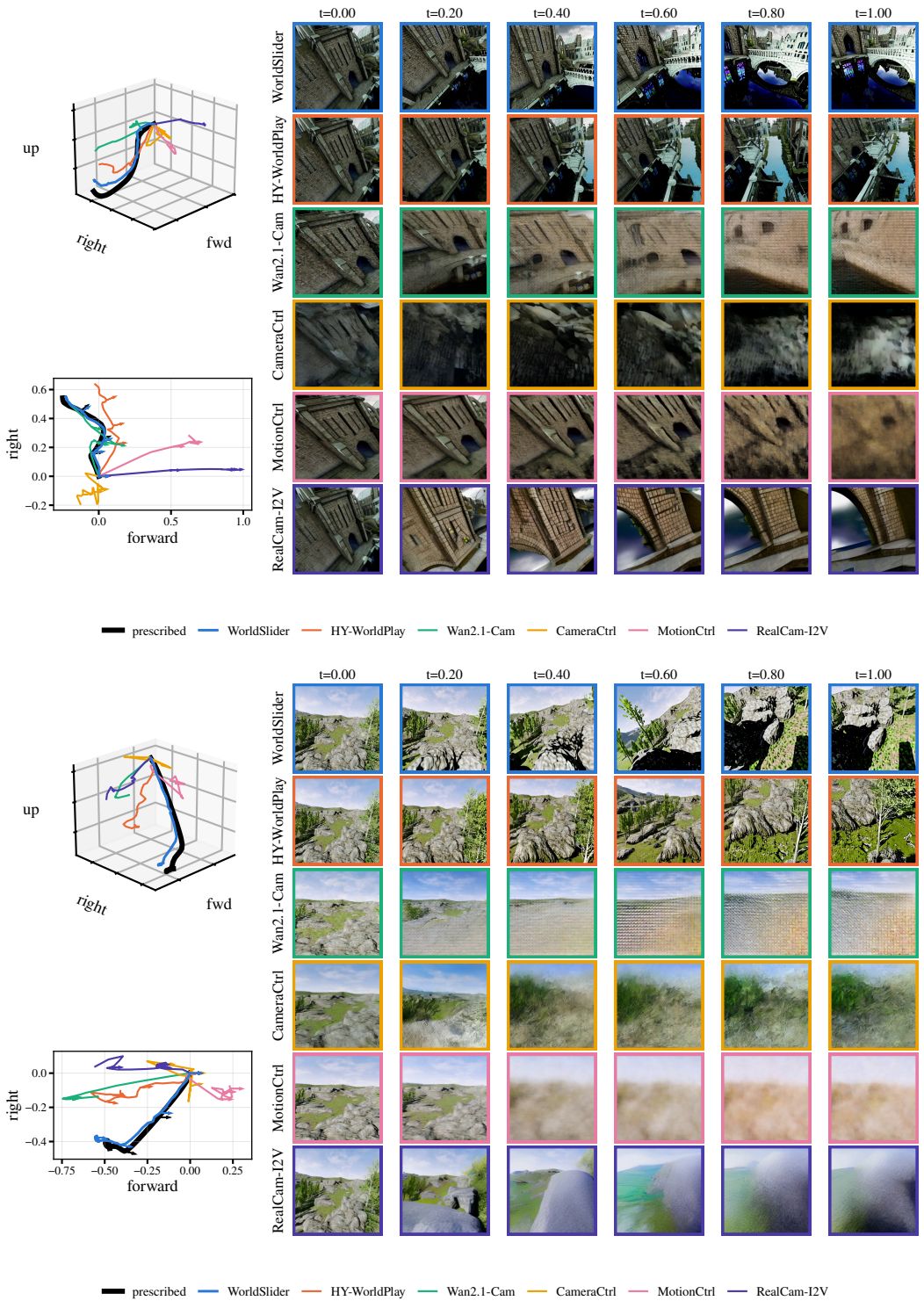

Figure 9: **Additional comparisons on diverse flights.**

A.2 PARALLEL WORLDS

Figure 10 provides qualitative examples of four flights with distinct translation, rotation, and altitude profiles. For each flight, three synchronized rollouts vary the indoor and outdoor geometry while preserving the direction and extent of motion prescribed by the trajectory at left. Table 5 reports cross-world trajectory fidelity.

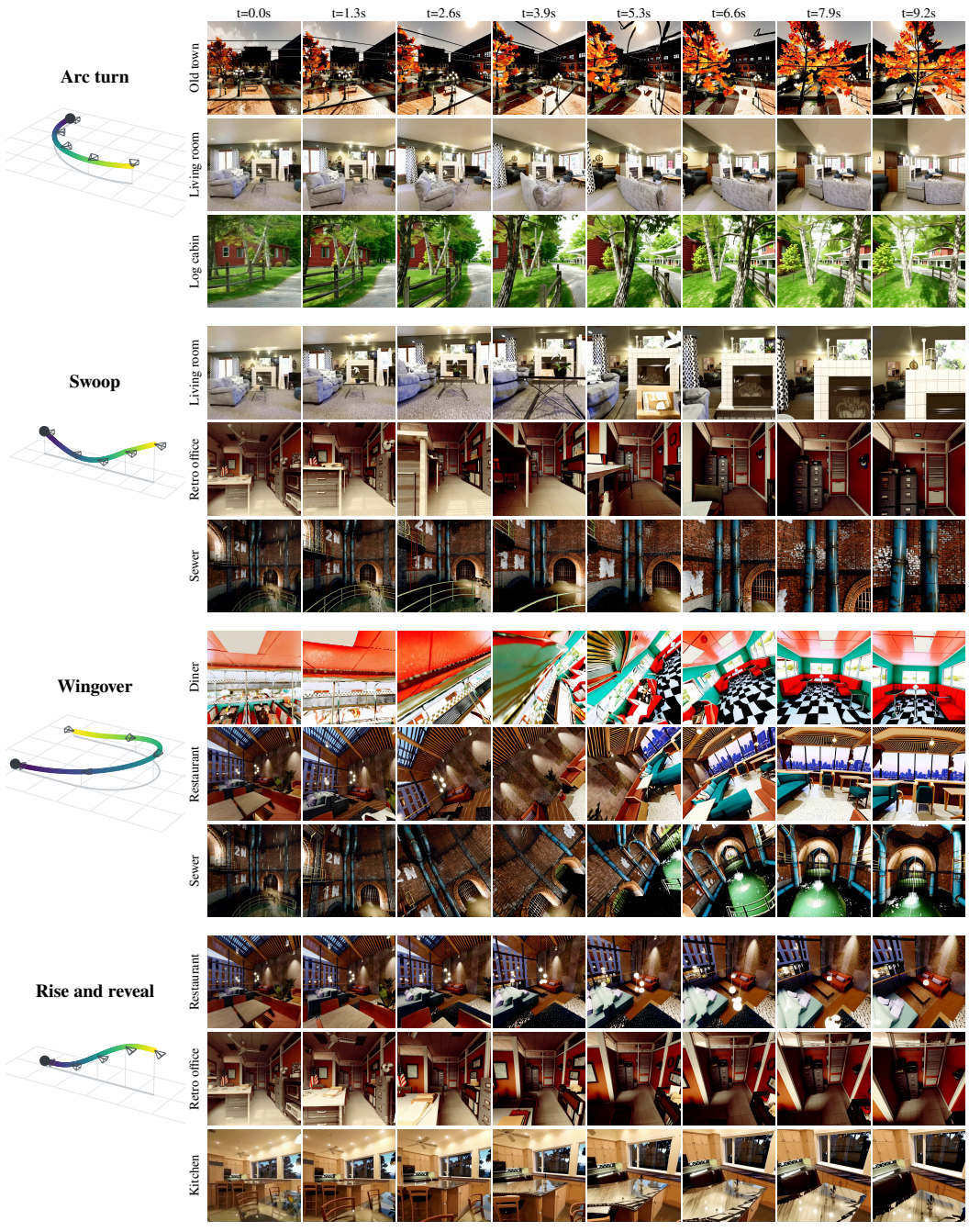

Figure 10: **Additional parallel worlds.** Each block fixes the prescribed flight shown at left and presents synchronized rollouts in three visually distinct worlds at right; columns report elapsed time.

A.3 VISUAL DIVERSIFICATION

Figure 11 shows three gate-traversal tasks and ten generated worlds spanning indoor, outdoor, seasonal, weather, and illumination variations. We edit initial frames with GPT Image 2 and use the original FlightGrid to generate coherent observations that retain the gate layout, flight progression, and waypoint labels. WorldSlider does not require a particular image-editing method.

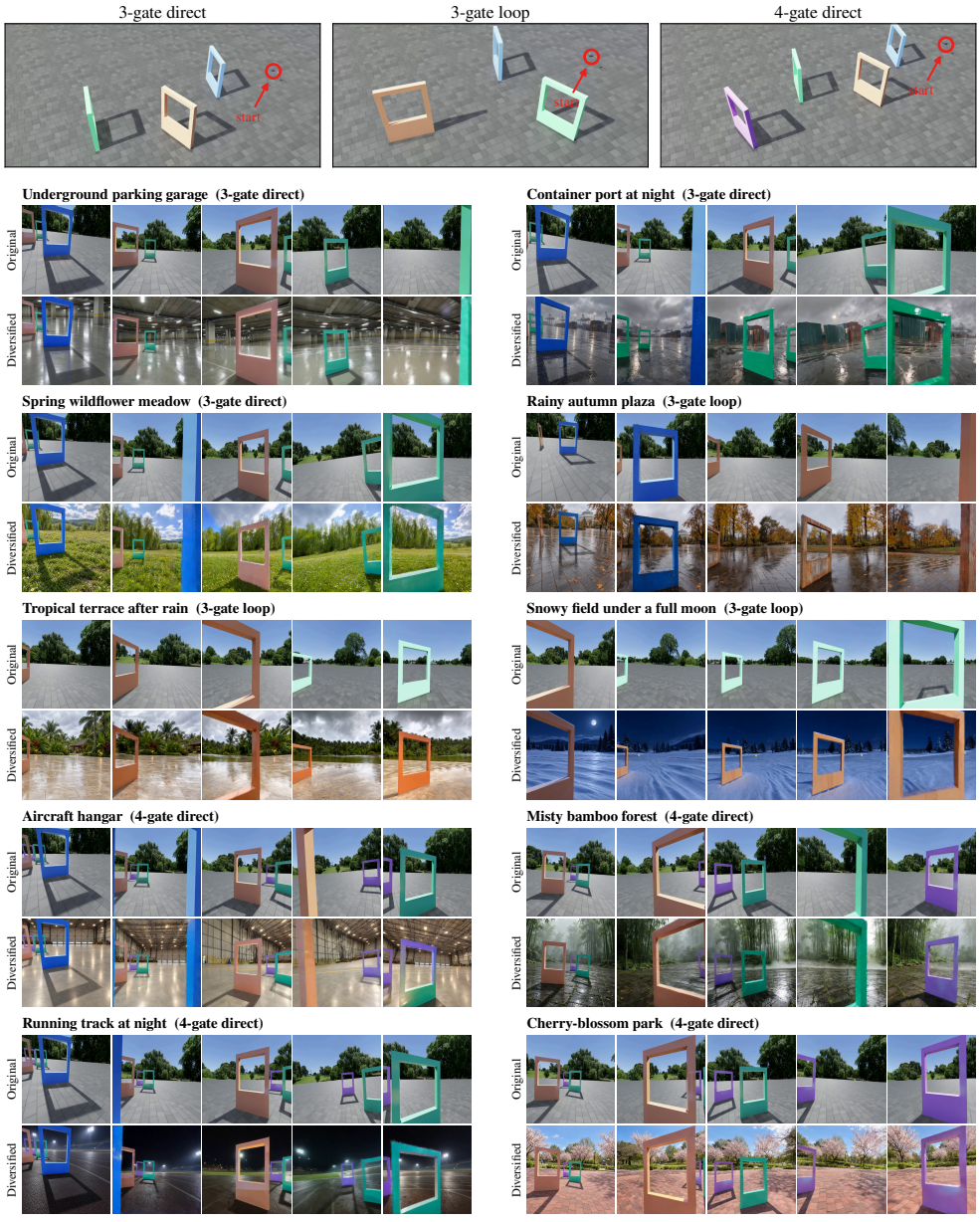

Figure 11: **Additional visual diversification results.** Top: the three gate-traversal tasks used for policy evaluation. Bottom: original simulator observations and their diversified counterparts across ten generated worlds, with the task structure and flight progression preserved.

B FLIGHT DATASET

B.1 CORPUS CONSTRUCTION

TartanAir-V2 (Wang et al., 2020) is a large-scale photorealistic simulation dataset developed for visual SLAM, odometry, and robot navigation. Its underlying acquisition pipeline maps each environment, samples nodes in free space, and connects them with RRT* (Karaman & Frazzoli, 2011) to form a graph of collision-free paths. Loop trajectories drawn from this graph are spline-smoothed, while incremental translation and rotation are randomized during collection. The resulting trajectories are collision-free and continuous, but navigation-driven rather than composed from canonical camera moves, producing the challenging motion patterns in Figure 12. We repurpose its synchronized RGB observations and metric camera poses as paired supervision for flight-conditioned video generation.

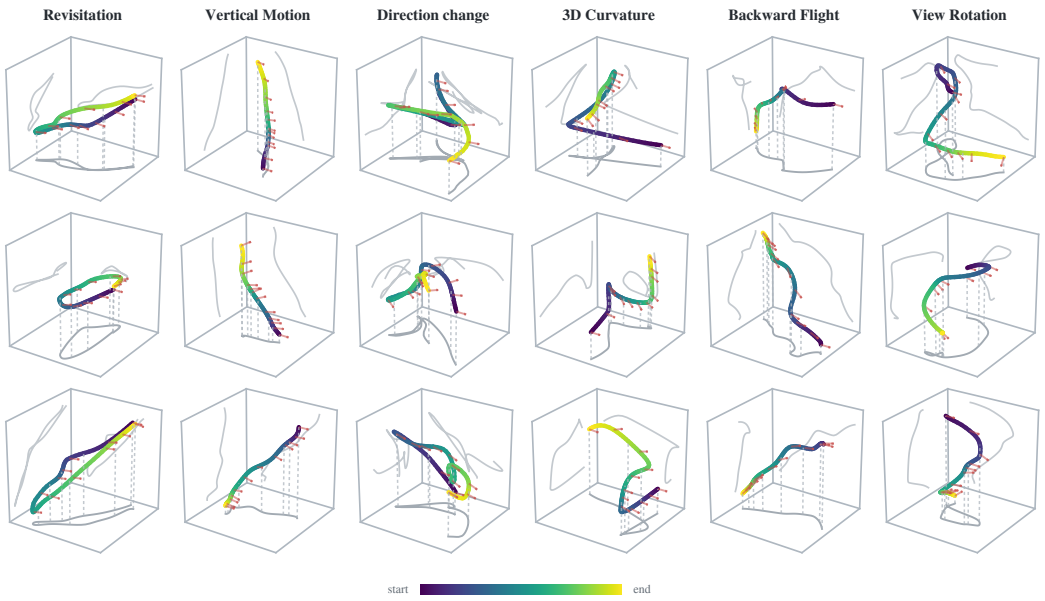

Figure 12: **Challenging training flights.** 93-frame windows grouped by six non-exclusive motion attributes. Color: time; red segments: viewing direction; gray curves: coordinate-plane projections. Positions use the first-camera frame and are normalized by path length.

The simulated corpus contains 24.9 hours of unique source footage from 558 TartanAir-V2 flights across 74 environments. We collect an additional 1.1 hours of physical flight using the platform in Appendix C.4. We synchronize its onboard RGB observations and metric poses and apply the same processing as for TartanAir-V2, yielding 26.0 hours of training footage. We divide each recording into overlapping 93-frame clips with a stride of 70 frames at 10 fps and 640×640 resolution; the simulated portion yields 12,369 such examples. Every example contains a target RGB video, its initial frame, the corresponding metric pose sequence, and a FlightGrid rendered from the same poses and timestamps. This construction preserves frame-level correspondence between the visual observation and its flight condition without estimating motion from RGB.

We generate a separate environment caption for every clip using Qwen3-VL-32B-Instruct[1]. The model observes eight frames sampled uniformly over the clip and describes only the visible world, including its setting, prominent objects, materials, colors, and illumination. The prompt excludes camera motion, flight direction, speed, and temporal narration. We screen the captions against a predefined list of motion, temporal, and medium-related terms, keeping text focused on the visual world and FlightGrid as the sole motion condition. Figure 13 summarizes how synchronized source streams provide the training conditions and RGB supervision.

[1] https://huggingface.co/Qwen/Qwen3-VL-32B-Instruct

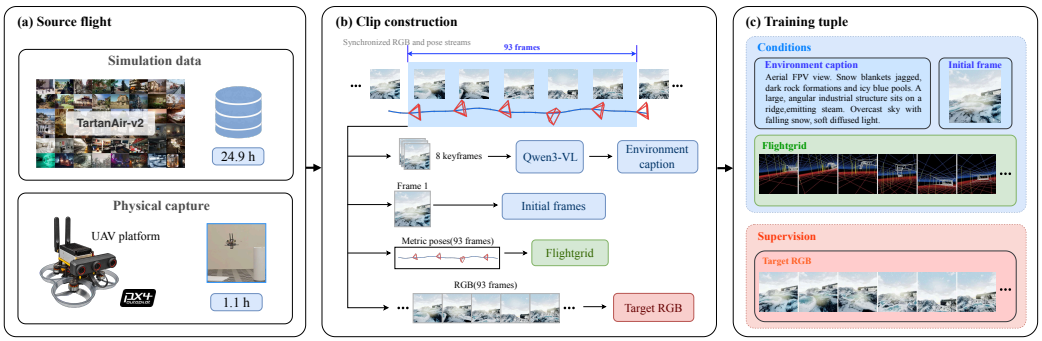

Figure 13: **Flight corpus construction.** (a) Simulated and physical capture. (b) Synchronized RGB and poses form 93-frame clips. (c) Captions, initial frames, and FlightGrid condition generation; RGB clips provide supervision.

Captioning prompt

Qwen3-VL-32B-Instruct | 8 uniformly sampled frames | $\leq$ 40 words

You are captioning aerial drone footage for a video generation dataset. Hard limit: at most 40 words total. Write two or three short sentences. Describe the visual world only: the environment type, the one or two most prominent objects and where they sit, the dominant materials or colors, and the lighting. Begin with "Aerial FPV view." Do not describe camera motion, flight direction, speed, or how the view changes over time. Do not mention "the video," "the footage," or "the camera." Present tense, declarative, concrete.

B.2 Visual and Motion Diversity

Figure 14 characterizes the TartanAir-V2 portion along the two dimensions of our conditioning design: the visual world described by text and the flight motion specified by FlightGrid. These corpus-level summaries cover the larger simulated portion only; physical recordings are not included.

Visual Diversity. The 74 simulated environments cover enclosed interiors, built environments, and open landscapes, with variations in season, weather, and illumination. Figure 14(A) summarizes this coverage through 44 environment concepts extracted from the clip captions. We merge synonymous terms and count each concept at most once per source flight, so overlapping clips do not inflate its frequency. Word size increases with source-flight prevalence, using a compressed scale for legibility; color groups related concepts. This vocabulary summarizes the range of visual settings paired with flight supervision while keeping motion out of the text condition.

Motion Diversity. Camera-control benchmarks commonly emphasize smooth, cinematographic viewpoint changes. Flight trajectories require position and attitude to evolve jointly along a timed path, including nonuniform speed and mismatched viewing and travel directions. Five descriptors characterize this regime: path indirectness, direction change, accumulated rotation, view–travel offset, and speed variability. They quantify path geometry, attitude variation, and temporal progression, complementing the flight patterns in Figure 12.

For completeness, let $\tau = \{(\mathbf{p}_t, R_t)\}_{t=0}^{N}$ be a uniformly sampled trajectory, where R_t maps camera coordinates to world coordinates and $\Delta t = 0.1\,\mathrm{s}$. Define the path length $L = \sum_{t=0}^{N-1} \|\mathbf{p}_{t+1} - \mathbf{p}_t\|_2$, local displacement $\mathbf{d}_t = \mathbf{p}_{t+h} - \mathbf{p}_t$, speed $v_t = \|\mathbf{d}_t\|_2/(h\Delta t)$, and viewing direction $\mathbf{f}_t = R_t(0,0,1)^{\mathsf{T}}$. We use $h = \min(5, \max(1, \lfloor N/2 \rfloor))$, giving a 0.5-second baseline whenever the window permits. Let $\angle(\cdot,\cdot)$ denote the angle between two vectors and $\angle_{\mathrm{SO}(3)}(\cdot,\cdot)$ the geodesic angle

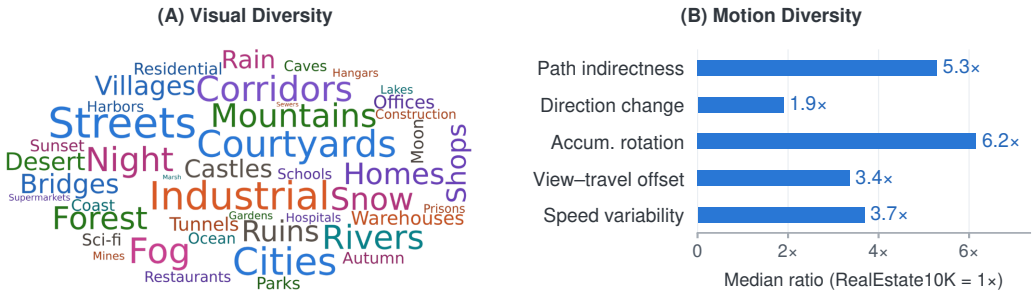

Figure 14: **Visual and motion diversity of the simulated corpus.** (A) Environment concepts, sized by source-flight prevalence. (B) Five motion descriptors under duration-matched sampling; bars show the TartanAir-V2 portion's median relative to RealEstate10K ($1\times$).

between two rotations, both in degrees. The five descriptors are

$$
\begin{aligned}
\eta(\tau) &= 1 - \frac{\|\mathbf{p}_N - \mathbf{p}_0\|_2}{L}, && \text{(path indirectness)},\\
\Phi(\tau) &= \frac{1}{h}\sum_{t=0}^{N-2h} \angle(\mathbf{d}_t, \mathbf{d}_{t+h}), && \text{(direction change)},\\
\Theta(\tau) &= \sum_{t=0}^{N-1} \angle_{\mathrm{SO}(3)}(R_t, R_{t+1}), && \text{(accumulated rotation)},\\
\Psi(\tau) &= \frac{\sum_{t=0}^{N-h} \|\mathbf{d}_t\|_2 \angle(\mathbf{f}_t, \mathbf{d}_t)}{\sum_{t=0}^{N-h} \|\mathbf{d}_t\|_2}, && \text{(view–travel offset)},\\
C_v(\tau) &= \frac{\mathrm{Std}_t(v_t)}{\mathrm{Mean}_t(v_t)}, && \text{(speed variability)}.
\end{aligned}
\tag{9}
$$

The local baseline reduces sensitivity to frame-to-frame pose jitter; the factor $1/h$ in Φ normalizes accumulated direction change for this h-frame baseline. The standard deviation in C_v is computed over all local speeds in the window. Zero-length trajectories are excluded; angular terms involving zero displacements and descriptors with zero denominators are assigned zero. All five descriptors are invariant to uniform rescaling of positions, allowing comparison with monocular RealEstate10K poses.

We use RealEstate10K (Zhou et al., 2018) as a representative reference for the smooth, canonical camera-trajectory regime common in prior camera-controlled video generation. RealEstate10K is used exclusively for evaluation: its 79,201 usable trajectories support the corpus-level statistics here, while a fixed set of 50 trajectories is used separately for zero-shot model evaluation in Section 4.1. No RealEstate10K sample contributes to WorldSlider training. Because its SfM poses have no absolute scale, we scale each of the 50 evaluation paths to a length of 9 m before rendering FlightGrid, matching the median TartanAir-V2 window. This choice affects only the control rendering scale; translation evaluation independently fits one global scale and normalizes by the prescribed path length.

To control for sequence duration, we resample trajectories at 10 Hz and take one centered RealEstate10K window per trajectory, capped at 9.2 seconds. We extract centered windows from the TartanAir-V2 portion with the same duration distribution and assign equal total weight to every source flight. Reference positions are linearly interpolated and orientations use spherical linear interpolation.

Figure 14(B) reports median ratios relative to this canonical-motion reference. Median path indirectness and accumulated rotation are $5.3\times$ and $6.2\times$ as large, respectively. Direction change, view–travel offset, and speed variability are $1.9\times$, $3.4\times$, and $3.7\times$ as large. These shifts place the simulated training corpus in a broader flight-motion regime across path geometry, attitude, and

Table 2: **FlightGrid rendering parameters.** All parameters are fixed across training and evaluation.

Parameter	Value
Input and projection	
Frames / frame rate / resolution	$93/10\,\text{fps}/640\times640$
Field of view	90°
Flight trace	
Cross-section	$0.6\times0.2\,\text{m}$
Ring interval	$0.3\,\text{s}$
Full / fade horizon	$3/5\,\text{s}$
Near-field fade	0.2–$0.8\,\text{m}$
Spatial scaffold	
Grid-cell width	$1\,\text{m}$
Band half-height	$h_0 = \max(1.5\,\text{m}, 6\widetilde{d})$
Intermediate-layer radius	$6\,\text{m}$
Marker offset from trace	1.5–$2.5\,\text{m}$
Attenuation range / minimum opacity	$30\,\text{m}/0.15$

timing than the canonical RealEstate10K reference. Reproducing these flights requires preserving the coupled progression of position and attitude beyond a plausible viewpoint change.

C Implementation and Evaluation Details

C.1 FlightGrid Parameterization

Design principle. FlightGrid combines a world-fixed spatial scaffold that conveys metric scale and gravity with an attitude-aware flight tube that encodes position, attitude, speed, and progression. Both are projected from the prescribed camera pose at every frame; the tube is additionally indexed by traversal time. The resulting visual control contains neither source appearance nor scene geometry. Projection uses calibrated camera intrinsics, with a 90° field of view for TartanAir-V2.

Metric spatial scaffold. We construct the scaffold in metric, gravity-aligned coordinates. Horizontal grids with one-meter cells establish scale and distinguish the two directions along gravity. Route-side vertical markers supply nearby translational parallax, including when the horizontal planes are viewed obliquely. For flights with substantial altitude variation, the scaffold is divided into multiple vertical bands so that a spatial reference remains locally visible throughout ascent and descent.

Let

$$h_0 = \max(1.5\,\text{m},\, 6\widetilde{d}), \qquad N_z = \max\left(1, \left\lceil \frac{z_{\max} - z_{\min} + 1\,\text{m}}{2h_0} \right\rceil\right), \tag{10}$$

where $\widetilde{d}$ is the median displacement between consecutive poses. The outer red and blue grids enclose N_z vertical bands of height $2h_0$. When $N_z > 1$, one green horizontal layer is inserted at every internal boundary, yielding $N_z - 1$ intermediate layers. Each intermediate layer is restricted to the region within $6\,\text{m}$ of the horizontal flight trace, avoiding unnecessary far-field geometry. Yellow vertical markers are placed at one-meter grid nodes whose horizontal distance from the trace lies in $(1.5, 2.5)\,\text{m}$. Scaffold opacity decreases with camera distance while retaining a nonzero far-field floor, keeping the metric reference visible without overwhelming the flight trace.

Flight trace over time. The agent trajectory is rendered as a narrow tube obtained by sweeping an attitude-aligned $0.6\,\text{m} \times 0.2\,\text{m}$ rectangular cross-section along the complete flight. Cross-sectional rings are placed every $0.3\,\text{s}$. Their orientation follows the camera attitude, while their spatial separation reflects the distance traveled during the corresponding time interval and therefore exposes changes in flight speed.

Every tube segment retains its traversal timestamp. At video time t, segments within $3\,\text{s}$ of t are rendered at full opacity and then fade linearly to zero at a temporal distance of $5\,\text{s}$. A short near-field opacity ramp prevents abrupt clipping as the tube approaches the camera plane. The representation

Table 3: **Training configuration.** The final stage uses reinforcement learning for flight alignment; the control branch initialization is compared in Section C.3.

Setting	Control adaptation	DMD	RL for flight alignment
Initialization	Depth control	SFT EMA	DMD
Trainable networks	ControlNet	G_θ, S_ψ	G_θ, S_ψ
Optimizer	AdamW	AdamW	AdamW
Control / student LR	8.63×10^{-5}	10^{-6}	10^{-6}
Fake-score learning rate	–	2×10^{-7}	2×10^{-7}
Weight decay	10^{-3}	10^{-2}	10^{-2}
Global batch (clips)	8	4	4
Student : fake-score updates	–	$1:4$	$1:4$
Sampling NFEs	35	4	4
Teacher CFG scale	–	3	3
Verifier interval	–	–	4 updates
Reward temperature β	–	–	1
Weight range $[w_{\min}, w_{\max}]$	–	–	$[0.9, 1.15]$
$\lambda_{\max}$ / ramp	–	–	$0.25/2{,}000$ updates

thus encodes the route together with each segment's traversal time and direction, which are particularly important for revisitation, backward flight, and self-intersecting trajectories.

C.2 Training Details

Training follows the three stages in Section 3: control adaptation, distribution matching for efficient sampling, and reinforcement learning for flight alignment. Each stage builds on the preceding model, allowing acceleration and geometric alignment to refine the learned control. All stages are trained on four NVIDIA H200 GPUs.

Control adaptation. We instantiate WorldSlider with the 2B Cosmos Transfer 2.5 model and initialize its control branch from the released depth-control checkpoint. We feed FlightGrid through the existing visual-control input without changing the architecture and optimize the control branch with the video DiT, VAE, and text encoder frozen. The adapted model's exponential-moving-average weights form the frozen multi-step teacher for distillation.

Distribution matching for efficient sampling. The student and fake-score model are initialized from the adapted teacher. With $\sigma_{\mathrm{d}} = 1$, the four TrigFlow evaluations correspond to rectified-flow times $t = 1, 15/16, 5/6, 5/8$, followed by the endpoint $t = 0$. We alternate one student update with four fake-score updates. Classifier-free guidance is applied only to the teacher; the student therefore requires four network evaluations at inference without an additional guidance branch. The normalizer and fake-score objective referenced in Equation 5 are

$$\begin{aligned} \delta(\hat{z}, \hat{z}_{\mathrm{T}}) &= \frac{1}{|\Omega|} \sum_{i \in \Omega} |\hat{z}_i - \hat{z}_{\mathrm{T},i}|, \\ \mathcal{L}_{\mathrm{fake}}(\psi) &= \mathbb{E}_{\hat{z},s,\xi}\left[\frac{\|\hat{z}_\psi(z_s, s \mid y) - \hat{z}\|_2^2}{\sin^2 s} \right], \end{aligned} \tag{11}$$

where Ω excludes latent entries occupied by conditional frames and $\hat{z}$ is treated as a fixed target when optimizing S_ψ.

Reinforcement learning for flight alignment. Reinforcement learning continues from the distilled checkpoint while retaining the same student, fake-score model, optimizer, and sampling schedule. Every fourth student update, the generated latent is decoded and evaluated by DA3-Large on 24 uniformly sampled frames. Geometry-reward weighting is introduced gradually over 2,000 updates and omitted whenever the verifier abstains. Table 3 collects the settings that differ across stages. We use $\mathcal{H} = \{2, 8\}$ on the 24-frame subsampled sequence, corresponding to temporal offsets of 0.8 and $3.2\,$s (8 and 32 source-video frames), and set $\kappa = 0.15$. For each source flight, we fit Q_τ

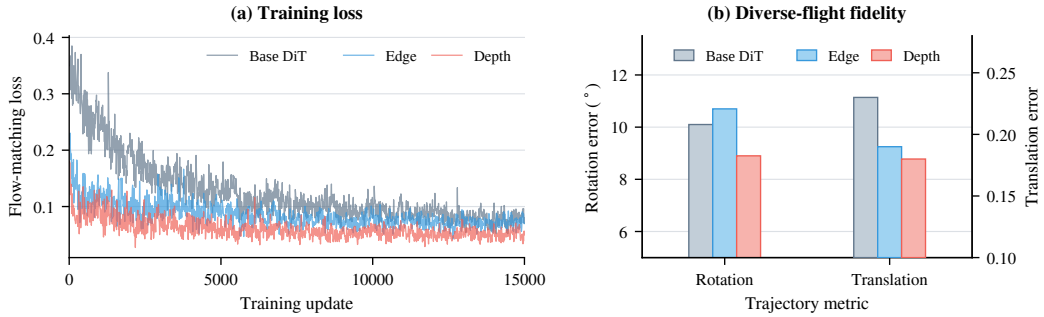

Figure 15: **Control initialization.** (a) Flow-matching loss over 15k matched updates. (b) ViPE rotation and scale-aligned translation errors on Diverse Flights after adaptation; lower is better.

once by pooling the segment displacements from both offsets and solving a rotation-only orthogonal Procrustes problem between the DA3 recovery and the prescribed trajectory. The SVD solution includes a determinant correction to exclude reflections. This rotation is frozen for generated videos, for which only the positive scale $s^\star$ is fitted. For $\rho(x; a, b) = \mathrm{clip}((x-a)/(b-a), 0, 1)$, the three bounded gates in Equation 7 are

$$\begin{aligned} h_{\mathrm{dir}} &= \rho(c; 0.10, c_{\mathrm{hi}}), & c_{\mathrm{hi}} &= \min(0.6, \max(0.8c_\tau, 0.15)), \\ h_{\mathrm{rot}} &= 1 - \rho([\theta - \theta_\tau]_+; 0.25, 0.90), & h_{\mathrm{mot}} &= \rho(m/m_\tau; 0, 0.6). \end{aligned} \tag{12}$$

Here c is the length-weighted translation-direction cosine, θ is the relative-rotation error in radians, and m is the mean optical-flow magnitude; the corresponding τ-subscripted quantities are measured on the source flight. The verifier abstains if pose recovery fails or if the source flight itself has $e_{\mathrm{seg}} \geq 0.6$.

Inference efficiency. The four-step student also absorbs the classifier-free guidance of its multi-step teacher, eliminating an additional guidance evaluation at inference. Both student and teacher are benchmarked on a single NVIDIA H200. The student achieves a generation throughput of 10 fps and a measured $18.5\times$ end-to-end speedup over the teacher.

C.3 Control Adaptation Analysis

The control branch can learn FlightGrid directly from pretrained base DiT weights. An existing dense-control checkpoint provides a useful visual interface for adaptation. To isolate this initialization choice, we train three otherwise identical branches from base DiT weights, an edge-control checkpoint, and a depth-control checkpoint, using the same data and optimization schedule.

Both dense-control warm starts achieve lower flow-matching loss at matched training updates than base DiT initialization (Figure 15(a)). The depth branch maintains the lowest loss throughout training, followed by the edge branch; the base DiT branch converges more slowly and to a higher loss. Prior dense visual control thus aids initialization without prescribing FlightGrid's semantics.

We assess flight fidelity separately from training loss, using the main benchmark's independent ViPE estimator to evaluate the adapted checkpoints on Diverse Flights. Figure 15(b) reports rotation errors of 10.1°, 10.7°, and 8.9°, and translation errors of 0.230, 0.190, and 0.180, for initialization from the base DiT, edge-control, and depth-control checkpoints, respectively. Edge initialization improves translation but not rotation over the base DiT, whereas depth initialization gives the lowest error on both measures. We consequently use the depth-control checkpoint for all remaining experiments.

C.4 Drone Platform

Our gate-traversal experiments use a shared simulation and physical deployment stack (Figure 16). The physical platform carries onboard cameras and an IMU connected to a companion computer, together with a PX4-based flight controller. ROS 2 connects perception, state estimation, high-level

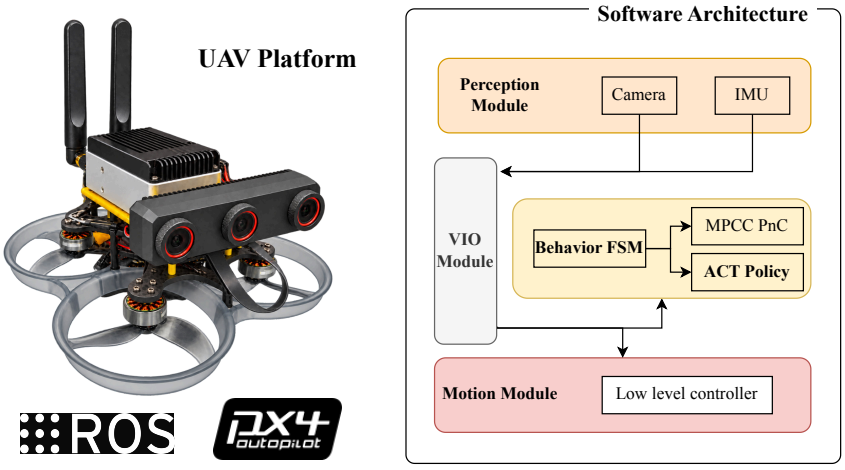

Figure 16: **Drone platform and shared flight stack.** Left: physical platform. Right: ROS 2/PX4 sensing, VIO, and control, with a state machine selecting MPCC or ACT. Isaac Sim shares these interfaces.

policy execution, and low-level control. The Isaac Sim setup mirrors the same sensor and control interfaces through PX4 software-in-the-loop, so the policy and flight stack are unchanged between simulated and physical execution.

The platform supports both data collection and policy execution. An MPCC controller with collision-avoidance constraints collects 1.1 hours of synchronized physical flight footage and poses for WorldSlider training. The ACT gate-traversal policy predicts short-horizon waypoints from egocentric RGB observations and uses the same stack. In both cases, the motion module converts the commanded trajectory into attitude and body-rate targets for the PX4 low-level controller, giving data collection and learned-policy deployment a common execution path.

C.5 Evaluation Protocol

Evaluation sets and shared inputs. We evaluate two fixed sets of 50 trajectories. Standard Trajectories comprise smooth, canonical RealEstate10K paths used only for zero-shot evaluation. Diverse Flights are selected for large rotation and vertical displacement, direction changes, path revisitation, and mismatched viewing and travel directions; their environments, source flights, and clips are excluded from training. Downstream application trajectories are also unseen. All methods are evaluated using their official released checkpoints and author-recommended native inference configurations. For each comparison, we canonicalize the prescribed trajectory at its first pose and resample the same complete curve to the corresponding camera convention. Evaluation uses normalized trajectory time, and all methods receive the same first frame, prompt, and three matched random seeds.

Camera conditioning. Table 4 summarizes how each evaluated method represents and injects the prescribed camera condition, following its released implementation.

Trajectory evaluation. We apply the released ViPE (Huang et al., 2025) pipeline to every generated video with the same configuration, without method-specific calibration or tuning. We canonicalize the recovered and prescribed trajectories at their first poses and resample both to 32 normalized timestamps. Rotation error is the mean geodesic angle between paired orientations. FlightGrid retains the metric structure of the input poses, but monocular video in a new world has no shared absolute scale: jointly scaling the scene and camera path preserves its projection. We therefore match one global scale to the prescribed path length and report mean pointwise distance normalized by that length. Throughout our evaluation, trajectory fidelity denotes orientation, normalized path geometry, and relative temporal progression; it does not measure absolute metric displacement. DA3 supplies training feedback but is never used for this evaluation. The monotonic ViPE gains with increasing

Table 4: **Camera-conditioning mechanisms.** Existing methods inject analytic camera encodings, whereas WorldSlider renders the prescribed flight as video control.

Method	Camera representation	Injection mechanism
MotionCtrl	Per-frame extrinsics	Pose tokens in temporal attention
CameraCtrl	Plücker ray field	Side encoder into attention
RealCam-I2V	Plücker ray field	Camera-control branch
Wan2.1-Camera	Plücker ray field	Adapter at patch embedding
HY-WorldPlay	Rays, poses, and intrinsics	Projective RoPE (PRoPE)
WorldSlider	FlightGrid video	Video-control branch

DA3 capacity in Figure 7(c) provide a cross-estimator check. For each method and prescribed trajectory, we first average each metric over the valid outputs from three matched random seeds. We then report the split mean and 95% confidence interval as 1.96 times the standard error across the 50 trajectory-level means.

Pose recovery failures. We count a recovery as valid when ViPE returns finite camera poses that can be resampled to all 32 evaluation timestamps. ViPE returns valid trajectories for 1792 of the 1800 generated videos. We omit only the eight failed seed-level outputs before forming the corresponding trajectory-level averages; all other generated outputs are retained.

Evaluation of visual quality. We evaluate every generated video with Q-Align (Wu et al., 2024), which predicts perceptual quality through text-defined ordinal levels. The main table reports the mean score over both evaluation sets and all three seeds. This measure is used only for visual quality; trajectory fidelity is evaluated separately with ViPE.

Cross-world trajectory fidelity. We evaluate 12 prescribed trajectories in 20 target worlds each, keeping the frame-by-frame FlightGrid identical across worlds for a given trajectory. This single-seed evaluation contains 240 videos, all successfully recovered by ViPE. We use the trajectory metrics above, first averaging errors over the 20 worlds for each trajectory, then reporting the mean and 95% confidence interval across the 12 trajectory-level means. Table 5 quantifies fidelity under changes in the target world.

Table 5: **Cross-world trajectory fidelity.** Mean $\pm$ 95% CI across trajectories after averaging over worlds. Rotation is in degrees; translation is scale-aligned and path-normalized.

Trajectories	Worlds/trajectory	Videos	Rot. $\downarrow$	Trans. $\downarrow$	Recovery
12	20	240	3.42 ± 1.28	0.0762 ± 0.037	$240/240$

C.6 Matched Controls

Table 6 compares control mechanisms using the same Cosmos Transfer 2.5 backbone, 26.0-hour corpus, 93-frame output, and optimization schedule. Both analytic-control variants adapt the shared video backbone with LoRA. Following Wan2.1-Camera, the Plücker variant injects per-pixel rays at the patch embedding; following HY-WorldPlay, PRoPE (Li et al., 2025a) encodes relative camera geometry through projective positional embeddings. We compare these variants with the FlightGrid teacher at the 35-NFE control-adaptation stage, without subsequent distillation of the analytic-control variants. At this stage, FlightGrid achieves lower trajectory errors on both held-out diverse flights and unseen canonical trajectories, together with higher Q-Align scores. The base-DiT-initialized FlightGrid variant in Section C.3 uses the same training budget, 35-NFE inference, and evaluation protocol. Without dense-control pretraining, it reaches 10.1° rotation error and 0.230 translation error on Diverse Flights, compared with $18.90^\circ/0.345$ for Plücker and $19.55^\circ/0.315$ for PRoPE. This comparison shows that the advantage on complex flights persists without a pretrained depth-control branch. For FlightGrid at four NFEs, geometric feedback reduces residual trajectory errors on both splits relative to plain DMD under the same additional training budget.

Table 6: **Matched variant analysis.** Shaded rows show absolute degradation from same-stage WorldSlider: the 35-NFE teacher or final four-step model. ViPE errors are mean ± 95% CI; Q-Align averages both splits.

			Standard trajectories		Diverse flights		
Variant	Stage	NFE	Rot. ↓	Trans. ↓	Rot. ↓	Trans. ↓	Quality ↑
Plücker rays	Control adaptation	35	9.20±1.10	0.134±0.018	18.90±1.20	0.345±0.020	3.17
Degradation vs. same-stage WorldSlider			Δ 6.15	Δ 0.037	Δ 10.00	Δ 0.165	Δ 0.31
PRoPE	Control adaptation	35	10.58±1.05	0.163±0.017	19.55±1.10	0.315±0.019	3.09
Degradation vs. same-stage WorldSlider			Δ 7.53	Δ 0.066	Δ 10.65	Δ 0.135	Δ 0.39
FlightGrid	Plain DMD	4	3.83±0.72	0.108±0.016	10.40±0.85	0.184±0.012	3.34
Degradation vs. same-stage WorldSlider			Δ 1.31	Δ 0.029	Δ 3.75	Δ 0.083	Δ 0.11

C.7 Human Preference

Q-Align measures perceptual video quality but is agnostic to whether the generated motion follows the prescribed flight. We therefore conduct two independent, blinded pairwise studies over all six methods. Cohort A judges visual quality alone, without seeing the prescribed flight. Cohort B sees the synchronized reference flight and judges which result better realizes its camera motion while remaining visually coherent. Each cohort contains 50 participants; every participant evaluates all 15 unordered method pairs once, with trajectories, seeds, order, and left–right placement balanced. We fit and normalize a separate Bradley–Terry model per cohort; 95% intervals use 10,000 two-way cluster-bootstrap replicates over participants and trajectories. All participants provided informed consent before taking part in the study.

Table 7: **Human preference scores.** Scores are normalized Bradley–Terry strengths; brackets show 95% confidence intervals. Cohort A rates visual quality; Cohort B jointly rates flight fidelity and visual coherence. Each cohort has 50 participants. Higher is better.

Method	Visual quality (%)	Flight-aware overall (%)
MotionCtrl	5.9 [4.4, 7.7]	4.8 [3.4, 6.6]
CameraCtrl	12.2 [9.5, 15.3]	6.9 [4.9, 9.4]
RealCam-I2V	9.7 [7.5, 12.3]	5.4 [3.8, 7.4]
Wan2.1-Camera	15.8 [12.5, 19.6]	8.2 [5.9, 11.1]
HY-WorldPlay	**27.6 [22.3, 33.3]**	16.2 [11.9, 21.4]
WorldSlider	**28.8 [23.3, 34.7]**	**58.5 [48.8, 67.3]**

In the visual-quality cohort, WorldSlider and HY-WorldPlay have similar estimated strengths (28.8% versus 27.6%), consistent with their close Q-Align scores. In the independent cohort shown the reference flight, WorldSlider is clearly preferred when fidelity and coherence are judged together: the two methods have normalized strengths of 58.5% and 16.2%, respectively.

D Limitations and Future Work

WorldSlider assumes calibrated input trajectories. Evaluation recovers generated motion monocularly and aligns one global scale, so it does not measure absolute metric displacement. Low-texture or corrupted videos can cause recovery failure. Scaling to uncalibrated drone, navigation, and Internet videos requires reliable scale recovery, temporal consistency, and confidence filtering of reconstructed geometry. Future work may support larger backbones and output resolutions, extend to more aggressive maneuvers, and evaluate visual diversification in closed-loop physical flight. The latter requires a separate embodied study with new data collection, system integration, and safety-controlled experiments.