PAIMON: AN AERIAL WORLD TRACKING MODEL BRIDGING SIMULATION TO REALITY

Anonymous authors
Paper under double-blind review

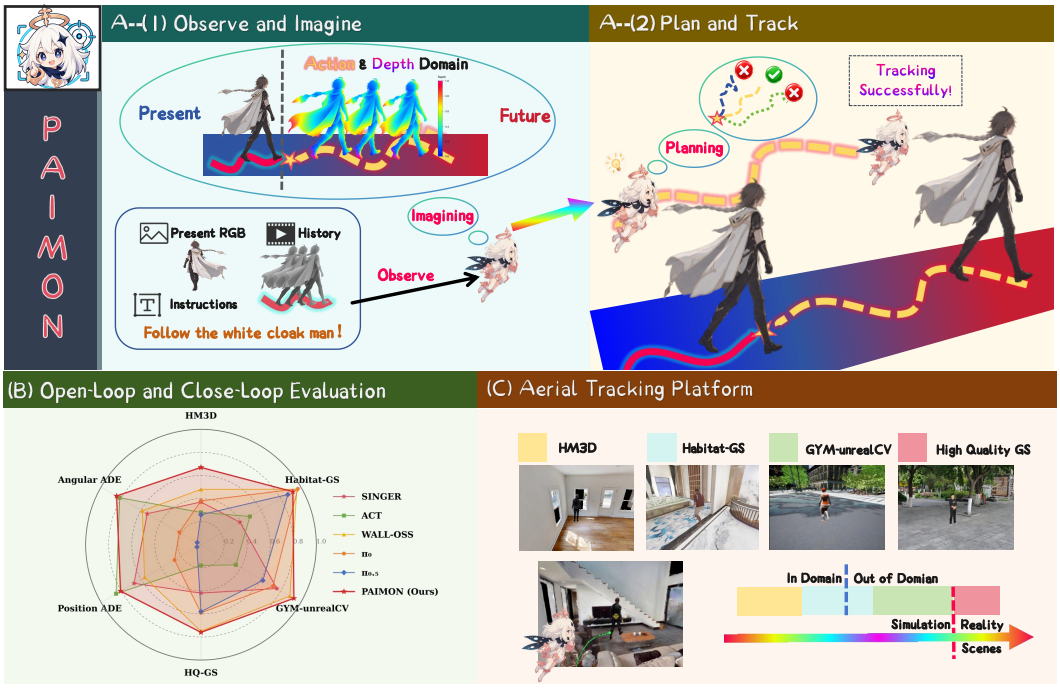

Figure 1: **PAIMON and ATP.** (A) PAIMON separates appearance transfer from action-relevant geometric learning. Future depth supervises camera-motion planning during training and is removed at deployment. (B) Open- and closed-loop results across ATP environments. Mean position and angular ADE are mapped to $(2.5 - 3\mathrm{ADE}_{\mathrm{pos}})$ and $(2.5 - 0.23\mathrm{ADE}_{\mathrm{ang}})$, respectively; ADE is measured in meters and degrees, and larger radar values indicate better performance. (C) ATP combines tracking expert, aligned simulation data, and four closed-loop evaluation environments.

ABSTRACT

Scaling high-quality supervision for aerial target tracking is challenging, as real-world flights rarely provide synchronized geometric and action supervision. Although simulation offers such supervision at scale, the appearance gap hinders sim-to-real transfer. We present **PAIMON**, a dual-expert MoT world action model that decouples action-relevant geometric learning from appearance adaptation. PAIMON translates simulated RGB observations toward real-world appearance with a structure-preserving image editor, while action-conditioned future-depth prediction provides geometric supervision during training. At deployment, PAIMON directly predicts actions from current observations, visual and ego-motion histories, and target instructions, without explicit video rollout. We further introduce the Aerial Tracking Platform (ATP) for rule-based expert data collection and closed-loop evaluation with high-fidelity 3D Gaussian-splatting scenes. PAIMON achieves the highest average closed-loop success and tracking rates among the evaluated baselines. Real-world flight tests further demonstrate transfer of simulation-learned tracking behavior to RGB-based closed-loop tracking under varying flight conditions. Code and data will be released upon publication.

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.

1 INTRODUCTION

Visual target tracking is a foundational capability of embodied perception, especially in situations where the embodied agent must interact with a moving referent, as in following a person through a crowd or handing an object to a collaborator. This capability matters even more for aerial agents, whose camera moves freely rather than along a ground-constrained path. Such mobility, combined with occlusion and limited onboard sensing, makes the target easy to lose and aerial target tracking a particularly challenging task (Sun et al., 2024; Dionigi et al., 2024; Sun et al., 2025).

Such a task calls for closed-loop control, in which every camera action decides what will be observed next. Vision-language-action (VLA) policies learn this control from demonstrations but leave the action–observation coupling implicit (Zhang et al., 2026). World-action models (WAMs) instead make this coupling explicit by predicting how observations evolve under actions, learning scene dynamics that the VLA paradigm cannot capture (Huang et al., 2026). This advantage comes at a price, since a WAM tracker needs whole videos aligned with expert actions and scene geometry, which real flights cannot provide at scale and simulation provides only in synthetic imagery (Dionigi et al., 2024). This leads to a central question: *How can an aerial WAM tracker learn **real-world** tracking from **simulation-only supervision**?*

Our answer builds on the observation that the deficiency of simulation lies in imagery, not in supervision. Generative editing can render simulated frames realistic while preserving their structure and hence their labels, but it operates per frame and cannot yield the temporally consistent video needed for WAM training. This calls for a prediction target invariant to the sim-to-real shift, and depth is exactly such a representation.

We therefore propose **PAIMON**, a Mixture-of-Transformers (MoT) world model that takes edited simulation images as input and comprises two well-aligned experts. The action expert refines data-derived motion anchors into 5-DoF camera trajectories via truncated diffusion, conditioned on the current frame, grayscale history, ego-motion, and a language instruction. The video expert predicts action-conditioned future depth videos with a latent diffusion transformer. The two experts interact through joint attention under a unidirectional mask, in which depth prediction is conditioned on the planned trajectory while planning remains independent of depth.

To support systematic closed-loop evaluation, we introduce the Aerial Tracking Platform (ATP), which combines three established simulation environments with a rule-based expert for collecting aerial-tracking camera trajectories. ATP also includes HQ-GS, two high-quality 3D Gaussian Splatting scenes reconstructed from 4K panoramic-camera and LiDAR captures. Their renderings, illustrated in Figure 3(B), provide a high-fidelity testbed for evaluating generalization to real-world appearance and geometry.

Our contributions are summarized as follows:

- We develop a sim-to-real training strategy that decouples action-relevant geometric supervision from visual appearance adaptation, using simulator-aligned depth and camera motion for geometry learning and structure-preserving editing for appearance transfer.
- We propose PAIMON, a dual-expert MoT world model that uses future depth as training-only geometric supervision via asymmetric joint attention and performs anchor-centered diffusion planning without test-time depth prediction or video rollout. Real-world flight tests further demonstrate transfer of simulation-learned tracking behavior to RGB-based closed-loop control on a physical quadrotor.
- We introduce ATP, which unifies rule-based expert-data collection and closed-loop evaluation across four simulation environments, including scenes reconstructed from real captures. On ATP, PAIMON achieves the highest average success and tracking rates among five baselines.

2 RELATED WORK

Active aerial tracking. Aerial target tracking couples target localization with camera control, because each action changes the next observation. Prerecorded benchmarks such as UAV123, UAVDT,

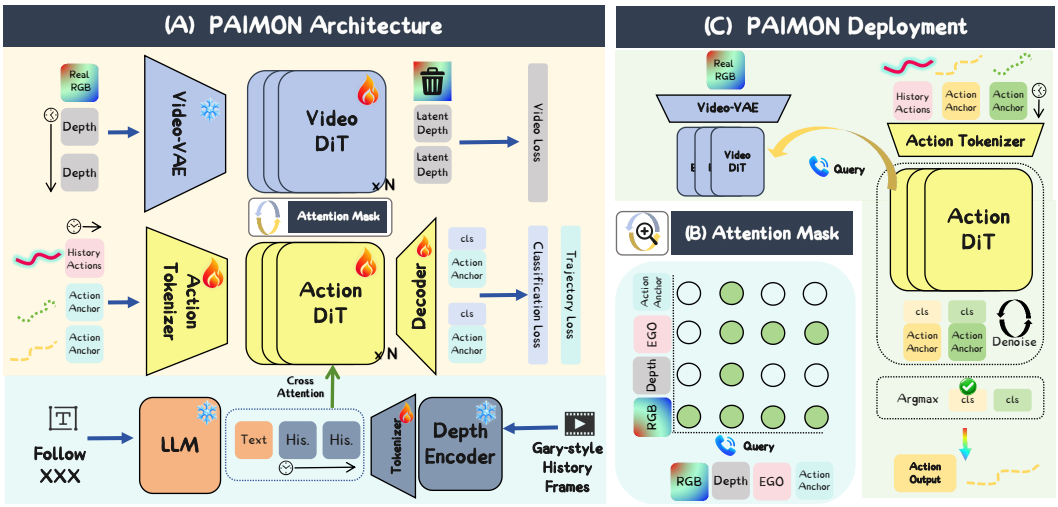

Figure 2: **PAIMON architecture.** (A) Training uses two experts: a video expert predicts action-conditioned future depth, and an action expert predicts camera trajectories. (B) An asymmetric attention mask lets the depth branch read action tokens while keeping future depth out of the action branch. (C) At deployment, PAIMON generates camera actions from the current RGB frame, visual and ego-motion history, and target instruction, without future-depth prediction or video rollout.

and VTUAV mainly measure image-space localization under camera motion, scale changes, and occlusion (Müller et al., 2016; Du et al., 2018; Zhang et al., 2022); AutoTrack and Aba-ViTrack improve localization and background modeling within this setting (Li et al., 2020; 2023). Active trackers instead learn or optimize observation-to-action policies (Luo et al., 2018; 2020; Sadeghi et al., 2018). RSPT uses reconstructed geometry and target prediction, D-VAT learns an end-to-end RGB policy with privileged state only for its critic, and depth-based visual servoing directly uses depth for camera motion (Zhong et al., 2023; Dionigi et al., 2024; Teuliere & Marchand, 2014). YOPO and YOPOv2-Tracker further incorporate depth/state planning and target-aware control (Lu et al., 2024; 2025). These methods establish the closed-loop formulation, but do not jointly address simulation-trained geometric supervision and RGB-only deployment under an appearance shift.

VLA and WAM in aerial robotics. VLA and WAM methods add target specification and predictive representations to embodied control. RT-2 and OpenVLA connect vision-language features to robot actions, while TrackVLA applies target recognition and anchor-based planning to embodied tracking (Zitkovich et al., 2023; Kim et al., 2024; Wang et al., 2025). Aerial variants differ in their interfaces: SINGER uses semantic images and onboard state, AutoFly adds pseudo-depth features, VLA-AN targets efficient onboard navigation, and UAV-Track VLA combines temporal compression, target-relative pose supervision, and flow-matching actions (Adang et al., 2025; Sun et al., 2026; Wu et al., 2025; Zhang et al., 2026). WAMs also differ at inference: NavDreamer generates and scores future videos, whereas Fast-WAM removes test-time future generation while retaining video-action co-training (Huang et al., 2026; Yuan et al., 2026). PAIMON follows this training–inference separation but predicts action-conditioned metric future depth, using it to teach camera-relative geometry during training while deploying from the current edited frame, motion history, ego motion, and language instruction only.

Simulation-to-real appearance transfer. Sim-to-real methods address rendering, dynamics, or feature mismatch through domain randomization, recurrent visual servoing, and distribution alignment such as CyCADA (Tobin et al., 2017; Peng et al., 2018; Sadeghi et al., 2018; Hoffman et al., 2018); SINGER also uses photorealistic Gaussian-splatting scenes for aerial data generation (Adang et al., 2025). Image editing offers a complementary appearance-level route: SDEdit preserves input structure, while InstructPix2Pix follows language-specified edits (Meng et al., 2022; Brooks et al., 2023). For tracking, however, edits that move the target or alter viewpoint break correspondence with metric depth and expert camera trajectories. PAIMON therefore edits appearance and instruction-specified clothing while preserving layout and pose, and applies the same editor to train-

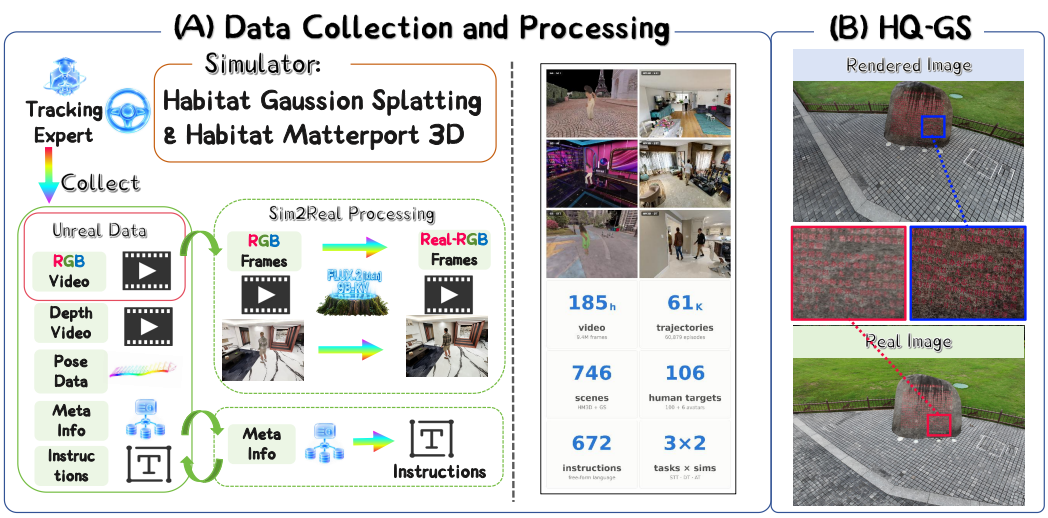

Figure 3: **ATP data collection and HQ-GS scenes.** (A) Expert collection and structure-preserving appearance transfer using HM3D and Habitat-GS. (B) HQ-GS rendering and corresponding real RGB capture with enlarged regions, showing close correspondence in scene layout, target placement, and fine-scale appearance.

ing images and simulator rollouts. This closes the visual gap without discarding the geometric supervision needed for action learning.

3 Data Collection and Sim-to-Real Transfer

3.1 Data Collection

Real-world aerial tracking data rarely provide dense depth, camera poses, expert camera actions, target boxes, and target instructions in one aligned episode. We therefore collect ATP's training data in simulation, where these signals are available together and can be varied systematically. As shown by the DataPipe in Figure 3, each episode aligns RGB/depth, camera pose, target box, instruction, and, when available, future camera actions. A rule-based expert generates camera trajectories subject to target motion, navmesh reachability, visibility, and collision/clearance constraints.

The training corpus contains 60,879 episodes (185.12 hours): 41,638 HM3D (Savva et al., 2019) and 19,241 Habitat-GS (Xia et al., 2026) episodes. The fixed open-loop validation set contains 2,000 HM3D episodes and 200 Habitat-GS episodes. Both sources use Single-Target Tracking (STT), Distractor Tracking (DT), and Ambiguous / Appearance Tracking (AT). STT tracks one target without distractors. DT adds pedestrians and uses appearance-specific target instructions. AT includes distractors and pairs a generalized identity instruction with a target avatar randomized per episode.

Within each source, the training and validation scenes are disjoint. HM3D follows TrackVLA's official partition (Wang et al., 2025) with 703 training and 101 nominal validation scenes. Habitat-GS uses 110 training and 19 nominal validation scenes, including synthetic environments and InteriorGS reconstructions. HM3D uses reconstructed indoor meshes. The retained validation episodes cover 80 HM3D and 18 Habitat-GS scenes (Table 6). For HM3D STT/DT episodes, 70 of the 100 humanoid identities are used for training and the other 30 for validation. Distractors may use the full pool, and AT randomizes the target avatar per episode. Habitat-GS uses six Gaussian avatars shared across scene splits. Appendix A details collection, filtering, and annotation.

3.2 Sim-to-Real Transfer

Simulation provides aligned labels, but its rendered RGB differs from real imagery in materials, illumination, textures, and avatar appearance. To reduce this gap, we apply the structure-preserving editing step in Figure 3: a frozen FLUX.2 Klein-9B editor uses metadata-conditioned SDEdit (Meng

Table 1: Point-estimate FID gains, where larger positive values indicate greater reductions in FID. Mean is the unweighted average across the five reference domains.

Simulator source	ImageNet	Landscape	COCO	Pascal VOC	NYU Depth V2	Mean ↑
Habitat-GS	+1.412	+6.205	+0.427	+0.942	+0.267	+1.851
HM3D	+6.839	+1.608	+5.809	+6.044	+22.647	+8.589

et al., 2022) to adapt selected simulator frames toward real-image appearance while preserving scene layout, viewpoint, target position, and person count. Depth, target annotations, camera poses, action labels, and source-provided language metadata remain paired and unchanged. Appendix A details the editor and sampling protocol.

We measure the appearance shift with Fréchet Inception Distance (FID) against five held-out real-image reference domains using 2,000 complete, matched source/edited frame pairs per simulator source. The reference domains are ImageNet (Deng et al., 2009), Landscape, COCO (Lin et al., 2014), Pascal VOC (Everingham et al., 2010), and NYU Depth V2 (Silberman et al., 2012). The four open-source datasets are evaluated on their respective public validation splits. Landscape is a web-crawled dataset of high-quality landscape images selected for composition quality. All images are resized directly to 640×480. Table 1 reports the point-estimate gains; Appendix A gives the metric and feature-extraction details.

The positive gain for every source–reference pair shows that appearance transfer reduces the FID between both simulator sources and all five evaluated real-image domains, aligning simulator RGB toward the evaluated real-image distribution. Habitat-GS has a smaller mean gain than HM3D, consistent with its InteriorGS-reconstructed scenes already being closer to real-image references. The FID audit measures appearance alignment; preserved paired labels and downstream tracking results evaluate the corresponding geometric and task-level behavior.

4 Method

4.1 Task Formulation

We formulate aerial target tracking as receding-horizon action prediction. At step t, the tracker receives the present RGB observation $\tilde{\mathbf{I}}_t$, an observation history $\mathcal{H}_t$, a target instruction s, and recent camera motion, and predicts a normalized present-ego action sequence

$$\bar{\boldsymbol{Y}}_t = (\bar{\boldsymbol{y}}_{t,1}, \ldots, \bar{\boldsymbol{y}}_{t,T}), \qquad \bar{\boldsymbol{y}}_{t,k} = [\Delta x, \Delta y, \Delta z, \Delta\text{pitch}, \Delta\text{yaw}]_{t,k} \in \mathbb{R}^{d_u}, \tag{1}$$

where $d_u = 5$ under the zero-roll convention. The increments are normalized in the current camera frame. The controller executes only a leading prefix and replans after the next observation, which provides closed-loop correction without committing to a full open-loop trajectory. Coordinate transforms, angle wrapping, and normalization are given in Appendix B.

4.2 Dual-Expert PAIMON Architecture

Figure 2 summarizes PAIMON, a dual-expert diffusion model for action generation with geometric supervision. The action expert maintains M anchor-centered motion hypotheses, while the video expert models the present RGB frame and future depth latents. The experts have separate parameters and communicate through one masked joint-attention layer per block. The mask is asymmetric: depth tokens can read action hypotheses, but action tokens cannot read future-depth tokens. This design lets future depth supervise action representations during training while remaining absent from deployment.

Action Expert: Anchor-Centered Diffusion Planning We represent multimodal camera motion with normalized step increments and M data-derived anchors $\bar{\boldsymbol{A}}^{(m)} \in \mathbb{R}^{T \times d_u}$. The anchors are cluster means computed offline from the training motion pool (Wang et al., 2025; Xia et al., 2026); they provide valid motion prototypes without imposing hand-written templates. At diffusion index

τ, each anchor is perturbed independently:

$$\boldsymbol{X}_\tau^{(m)} = \sqrt{\bar{\alpha}_\tau}\,\bar{\boldsymbol{A}}^{(m)} + \sqrt{1-\bar{\alpha}_\tau}\,\epsilon^{(m)}, \qquad \epsilon^{(m)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{2}$$

where the expert trajectory is not used to construct $\boldsymbol{X}_\tau^{(m)}$. Thus, the noisy input preserves mode diversity without revealing the ground-truth mode. Given the present visual latent $\mathbf{R}_t$, ego token $\boldsymbol{e}_t$, instruction tokens $\boldsymbol{L}_t$, history tokens $\boldsymbol{H}_t$, and the noisy anchors, the action expert predicts one clean trajectory and one classification logit $\ell^{(m)}$ per hypothesis:

$$\left\{\hat{\bar{\boldsymbol{Y}}}_0^{(m)}, \ell^{(m)}\right\}_{m=0}^{M-1} = f_{\boldsymbol{\theta}}\left(\mathbf{R}_t, \boldsymbol{e}_t, \boldsymbol{L}_t, \boldsymbol{H}_t, \{\boldsymbol{X}_\tau^{(m)}\}_{m=0}^{M-1}, \tau\right). \tag{3}$$

Action prediction uses separate instruction and history conditioning paths. The instruction is encoded offline by frozen umT5-XXL (Chung et al., 2023) and mapped to the action width by a trainable projection. During training, to mitigate the limited generalization of rendered RGB history frames to real-world appearances, each of the preceding eight frames is converted to grayscale and independently encoded by the frozen DA3 ViT-L backbone (Lin et al., 2025). A trainable history tokenizer then compresses these features into tokens with temporal and spatial positional embeddings:

$$\boldsymbol{L}_t = g_{\text{lang}}(\text{umT5}(s)), \qquad \boldsymbol{H}_t = g_{\text{hist}}(\text{DA3}(\mathcal{H}_t^{\text{gray}})). \tag{4}$$

At block r, two residual cross-attention modules update the complete action sequence, including mode and ego tokens:

$$\boldsymbol{A}_{\text{ctx}}^{(r)} = \mathcal{C}_{\text{lang}}^{(r)}\left(\mathcal{C}_{\text{hist}}^{(r)}(\boldsymbol{A}^{(r)}, \boldsymbol{H}_t), \boldsymbol{L}_t\right). \tag{5}$$

Language and history tokens therefore condition only the action expert and do not alter the joint video–action token interface.

Video Expert: Action-Conditioned Future Depth The video expert supplies geometric supervision through future depth rather than through a test-time visual rollout. The present RGB frame and the depth clip are both converted to the input format of the same frozen Wan VAE (Wan et al., 2025); depth is clipped to the valid range, mapped to three channels, and encoded by this RGB VAE without a separate depth tokenizer. Let $\mathbf{R}_t$ be the present RGB latent and $\mathbf{D}_0, \ldots, \mathbf{D}_{J-1}$ the depth latents. During training, only the depth latents are noised:

$$\mathbf{R}_t = \mathcal{E}_{\text{VAE}}(\tilde{\mathbf{I}}_t), \qquad \mathbf{D}_j = \mathcal{E}_{\text{VAE}}\left(\phi(\tilde{\mathbf{D}}_{t+j})\right), \qquad \mathbf{Z}_0 = [\mathbf{R}_t \mid \mathbf{D}_0, \ldots, \mathbf{D}_{J-1}], \tag{6}$$

where ϕ performs the fixed depth-to-three-channel mapping. Continuous flow matching constructs the training input and target field as

$$\mathbf{Z}_\sigma = (1-\sigma)\mathbf{Z}_0 + \sigma\epsilon, \qquad \mathbf{V}^* = \epsilon - \mathbf{Z}_0, \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{7}$$

The video expert predicts the flow field conditioned on the present RGB latent, action tokens, and ego token. The loss is evaluated only on depth positions, so the branch learns geometry while the clean present-RGB latent remains available as a visual condition.

4.3 Geometry-Supervised Joint Training and Rollout-Free Inference

The central design question is how to use future depth for learning without making it an inference dependency. We concatenate the present RGB tokens, depth tokens, action mode tokens, and ego token, and apply the following visibility mask. Let $\mathcal{D} = \{\mathbf{D}_0, \ldots, \mathbf{D}_{J-1}\}$ and $\mathcal{A} = \{\boldsymbol{a}_0, \ldots, \boldsymbol{a}_{M-1}\}$ denote the depth and action token sets, respectively; $\mathcal{K}(q)$ denotes the keys visible to query q:

$$\begin{aligned} &\mathcal{K}(\mathbf{R}_t) = \{\mathbf{R}_t\}; && \mathcal{K}(\mathbf{D}_j) = \{\mathbf{R}_t\} \cup \mathcal{D} \cup \mathcal{A} \cup \{\boldsymbol{e}_t\} && j = 0, \ldots, J-1; \\ &\mathcal{K}(\boldsymbol{e}_t) = \{\mathbf{R}_t, \boldsymbol{e}_t\}; && \mathcal{K}(\boldsymbol{a}_m) = \{\mathbf{R}_t, \boldsymbol{a}_m, \boldsymbol{e}_t\} && m = 0, \ldots, M-1. \end{aligned} \tag{8}$$

Thus depth queries can read candidate actions and transmit geometric gradients to the action expert, whereas action queries cannot read future depth. Modes remain isolated from one another, so removing depth tokens at inference leaves the action-visible subgraph unchanged.

For a normalized expert trajectory $\bar{\boldsymbol{Y}}$, we assign its mode in cumulative-waypoint space:

$$P_k(\bar{\boldsymbol{Y}}) = \sum_{j=1}^{k} \bar{\boldsymbol{y}}_j, \qquad m^* = \underset{m \in \{0,\ldots,M-1\}}{\arg\min} \frac{1}{T d_u} \sum_{k=1}^{T} \left\| P_k(\bar{\boldsymbol{Y}}) - P_k(\bar{\boldsymbol{A}}^{(m)}) \right\|_2^2. \tag{9}$$

This assigns labels in the integrated path space. The selected mode receives winner-takes-all regression supervision. For classification, the action expert produces one logit $\ell_{b,m}$ for each anchor. Let $y_{b,m} = \mathbf{1}[m = m_b^*]$ and $p_{b,m} = \sigma(\ell_{b,m})$ denote the binary target and sigmoid probability, respectively. We use $p_{b,m}^{\mathrm{t}} = p_{b,m}$ for a positive anchor and $p_{b,m}^{\mathrm{t}} = 1 - p_{b,m}$ otherwise, and apply an independent per-anchor sigmoid focal BCE loss:

$$\begin{aligned} \mathcal{L}_{\mathrm{reg}} &= \frac{1}{B}\sum_{b=1}^{B}\frac{1}{Td_u}\sum_{k=1}^{T}\left\| P_k\left(\hat{\boldsymbol{Y}}_{0,b}^{(m_b^*)}\right) - P_k(\bar{\boldsymbol{Y}}_b)\right\|_2^2, \\ \mathcal{L}_{\mathrm{cls}} &= -\frac{1}{B}\sum_{b=1}^{B}\sum_{m=0}^{M-1}(1-p_{b,m}^{\mathrm{t}})^{\gamma}\left[y_{b,m}\log p_{b,m} + (1-y_{b,m})\log(1-p_{b,m})\right]. \end{aligned} \tag{10}$$

We set the focal exponent to $\gamma = 2$. We jointly optimize the video flow-matching, trajectory regression, and mode classification losses using the weighted objective in Eq. 15. At inference, a deterministic DDIM scheduler denoises the anchor-centered hypotheses. Its schedule and update rule are detailed in Appendix B (Eq. 14). The final sigmoid scores select one mode, whose normalized increments are denormalized, accumulated into waypoints, and executed as a leading prefix before replanning. Future depth provides training supervision, while inference requires neither depth prediction nor video rollout.

5 Experiments

We evaluate open- and closed-loop performance on ATP. Open-loop evaluation uses the HM3D and Habitat-GS validation sets; closed-loop tracking covers HM3D (Wang et al., 2025), Habitat-GS (Xia et al., 2026), GYM-unrealCV (Zhong et al., 2025), and HQ-GS. HQ-GS reconstructs scenes from real captures; Figure 3(B) shows close visual correspondence between an HQ-GS rendering and the corresponding real RGB capture, supporting its use as a realistic simulation proxy for closed-loop evaluation. We compare PAIMON with five baselines: the VLA policies π_0 (Black et al., 2024), $\pi_{0.5}$ (Intelligence et al., 2025), and SINGER (Adang et al., 2025), together with ACT (Zhao et al., 2023) and WALL-OSS (Zhai et al., 2025). All methods use the same training data and evaluation configuration. PAIMON initializes its video expert from the WAN2.1-1.3B checkpoint (Wan et al., 2025), and each model predicts a 12-step camera trajectory at 0.2 s intervals as five-dimensional ego-relative increments. We measure closed-loop success and tracking rates (SR/TR), and open-loop horizon-average and final-step errors (ADE/FDE) for position, pitch, and yaw. The closed-loop metrics are defined in Appendix C.3.

5.1 Closed-loop Tracking Results

As shown in Table 2, PAIMON achieves state-of-the-art closed-loop tracking on HQ-GS, whose scenes are reconstructed from real captures. This result tests transfer to real-capture-derived appearance and geometry under the 2 Hz evaluation protocol. For all other environments, we apply the same real-time FLUX-based appearance transfer to every model's closed-loop observations except PAIMON-Raw, matching the training inputs and keeping the comparison protocol consistent. Overall, PAIMON achieves the highest average SR and TR in Table 2, with the largest gains over the strongest baseline on HM3D. On Habitat-GS, PAIMON achieves the highest TR, whereas π_0 achieves the highest SR. Reducing PAIMON's replanning frequency from 2 Hz to 1 Hz lowers the performance of most models. The corresponding 1 Hz comparisons and detailed scene-level result are reported in Appendix C.4. Representative baseline trajectories are visualized in Appendix Figures 10, 11, 12, and 13.

5.2 Open-loop Planning Metrics

PAIMON's open-loop advantage is strongest in orientation prediction: Table 3 shows the lowest pitch and yaw errors on HM3D and the lowest yaw errors on Habitat-GS. ACT remains stronger on HM3D position errors and Habitat-GS pitch errors, ties PAIMON on Habitat-GS position ADE, but has a higher position FDE there. Yet ACT's closed-loop SR and TR are lower across all four environments in Table 2. This contrast shows that low open-loop position error alone does not determine tracking performance under repeated execution and replanning.

Table 2: Closed-loop tracking results at 2 Hz on the four ATP evaluation environments. Best results across all tables are shown in **bold**.

Model	HM3D		H-GS		GYM		HQ-GS		Avg.	
	SR ↑	TR ↑	SR ↑	TR ↑	SR ↑	TR ↑	SR ↑	TR ↑	SR ↑	TR ↑
SINGER (Adang et al., 2025)	0.37	0.53	0.39	0.64	0.76	0.72	0.42	0.39	0.49	0.57
ACT (Zhao et al., 2023)	0.28	0.39	0.49	0.70	0.35	0.56	0.18	0.21	0.33	0.47
WALL-OSS (Zhai et al., 2025)	0.48	0.56	0.96	0.89	0.89	0.82	0.75	0.70	0.77	0.74
π_0 (Black et al., 2024)	0.38	0.50	**0.97**	0.89	0.72	0.71	0.59	0.52	0.67	0.65
$\pi_{0.5}$ (Intelligence et al., 2025)	0.26	0.40	0.87	0.85	0.62	0.59	0.58	0.53	0.58	0.59
PAIMON(Ours)	**0.67**	**0.74**	0.92	**0.95**	**0.93**	**0.89**	**0.76**	**0.85**	**0.82**	**0.86**

Table 3: Open-loop trajectory prediction errors on ATP. Position errors are in meters and angular errors are in degrees. We report the deployed trajectory selected by each model.

Model	HM3D						GS					
	Pos. ADE (m) ↓	Pos. FDE (m) ↓	Pitch ADE (deg) ↓	Pitch FDE (deg) ↓	Yaw ADE (deg) ↓	Yaw FDE (deg) ↓	Pos. ADE (m) ↓	Pos. FDE (m) ↓	Pitch ADE (deg) ↓	Pitch FDE (deg) ↓	Yaw ADE (deg) ↓	Yaw FDE (deg) ↓
SINGER (Adang et al., 2025)	0.59	0.93	3.77	4.70	14.97	25.67	0.63	1.05	2.47	2.52	12.91	22.70
ACT (Zhao et al., 2023)	**0.50**	**0.83**	3.49	4.44	12.87	23.12	**0.60**	1.06	**1.43**	**1.80**	11.53	21.28
WALL-OSS (Zhai et al., 2025)	0.59	0.92	4.62	6.07	13.81	25.07	0.70	1.17	2.37	2.87	12.47	22.76
π_0 (Black et al., 2024)	0.69	1.06	4.79	5.90	17.20	29.61	0.80	1.30	3.17	3.50	14.56	25.87
$\pi_{0.5}$ (Intelligence et al., 2025)	0.79	1.26	5.39	6.57	18.54	31.51	0.85	1.37	3.35	3.48	15.54	26.82
PAIMON(Ours)	0.53	0.84	**3.47**	**4.25**	**12.14**	**21.89**	0.60	**1.00**	1.95	2.26	**11.22**	**20.36**

5.3 Ablation Study

Table 4 examines geometric supervision, appearance transfer, and model capacity through three variants. Experimental settings remain unchanged as prior. Action-only removes the video expert and its future-depth supervision; PAIMON-Raw uses the original simulator RGB frame in both training and evaluation; and PAIMON–5B uses a 5B video backbone with a correspondingly enlarged action expert. Appendix C.6 provides the architectural details and results by scene and replanning frequency.

In Table 4, Action-only increases open-loop position errors and lowers closed-loop SR and TR in every environment, supporting the joint contribution of the video expert and future-depth supervision. The Raw variant produces the largest drop in average tracking performance, particularly on HM3D and GYM, indicating the value of appearance transfer across environments. Its higher Habitat-GS SR is accompanied by lower TR and weaker results elsewhere. The 5B variant also underperforms the reference model, so increasing capacity provides no benefit under the evaluated training configuration. The corresponding qualitative ablation trajectories appear in Appendix Figures 14, 15, 16, and 17.

5.4 Inference Cost and Real-World Experiments

Table 5 shows that PAIMON combines planning latency close to π_0 with lower peak memory use, and is faster than $\pi_{0.5}$ and WALL-OSS. The column definitions and measurement protocol are given in Appendix C.7. Its 125.75 ms median planning latency fits within the 500 ms interval of the 2 Hz evaluation. ACT and SINGER are smaller and faster, but their lower tracking scores in Table 2 reveal a trade-off between inference cost and closed-loop performance. Together, the two tables show that PAIMON's tracking gains are compatible with the evaluated replanning rate on the tested hardware.

Finally, using the deployment setup described in Appendix C.8, we evaluate the trained PAIMON policy on a quadrotor in a flight-test laboratory. Here, we conduct trials without appearance transfer at different flight altitudes, target distances, and relative speeds, including turns and temporary target occlusions. As illustrated in Fig. 4, PAIMON maintains stable tracking through changes in flight

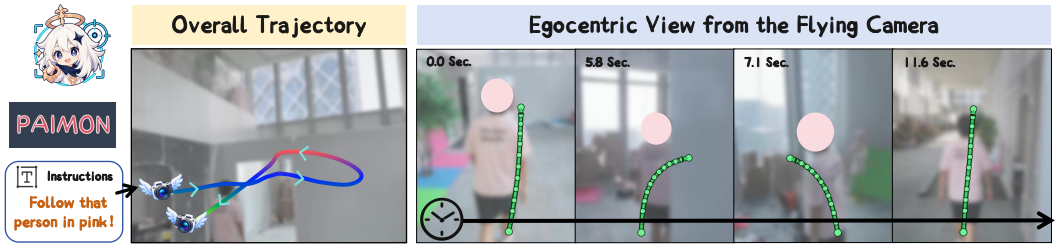

Figure 4: **Real-world PAIMON tracking.** Given an instruction, PAIMON maintains tracking through turns and temporary occlusions. All images are frosted for privacy.

Table 4: Open-loop position errors (m) and 2 Hz closed-loop tracking results for PAIMON variants.

		Pos.ADE	Pos.FDE	HM3D		H-GS		GYM		HQ-GS		Avg.	
Variant	Change	HM3D/GS ↓	HM3D/GS ↓	SR ↑	TR ↑	SR ↑	TR ↑	SR ↑	TR ↑	SR ↑	TR ↑	SR ↑	TR ↑
PAIMON(Ours)	Reference	**0.53/0.60**	**0.84/1.00**	**0.67**	**0.74**	0.92	**0.95**	**0.93**	**0.89**	**0.76**	**0.85**	**0.82**	**0.86**
PAIMON–5B	5B video	0.61/0.63	0.96/1.05	0.60	0.63	0.88	0.93	0.88	0.79	0.51	0.68	0.72	0.76
ACTION-ONLY	No video	0.60/0.62	0.93/1.02	0.58	0.67	0.71	0.81	0.75	0.83	0.70	0.70	0.68	0.75
PAIMON-Raw	Raw present	0.60/0.67	0.95/1.10	0.53	0.61	**0.97**	0.89	0.58	0.40	0.70	0.46	0.62	0.51

direction and temporary loss of visual evidence. Together, these trials indicate that PAIMON carries the action-relevant geometric behavior learned in simulation into real RGB control as viewpoint, scale, motion, and visibility change.

Table 5: Inference cost on an NVIDIA RTX 5090 GPU. All measurements use batch size 1, bf16 precision, two denoising steps, and eager inference.

Model	Params (B) ↓	Peak mem. (GiB) ↓	Plan latency p50 (ms) ↓	Plan rate (Hz) ↑	Compute (GFLOP)	Throughput (TFLOP/s)
SINGER (Adang et al., 2025)	0.1527	0.37	15.21	65.7	190.0	12.49
ACT (Zhao et al., 2023)	0.0664	0.28	6.35	157.5	9.3	1.47
WALL-OSS (Zhai et al., 2025)	4.1668	7.89	318.67	3.1	1494.8	4.69
π_0 (Black et al., 2024)	3.5014	6.67	120.66	8.3	4090.7	33.90
$\pi_{0.5}$ (Intelligence et al., 2025)	3.6168	7.12	142.62	7.0	4730.0	33.16
PAIMON (Ours)	2.7084	5.53	125.75	8.0	4284.0	34.07

6 CONCLUSION

We study simulation-to-reality transfer for closed-loop aerial target tracking. Our key observation is that simulation provides the geometric and action supervision needed for learning, but not the visual appearance of the real world. Based on this insight, we present **PAIMON**, a Mixture-of-Transformers world model that factorizes geometric supervision from visual appearance. PAIMON learns action-relevant camera motion from simulator-aligned depth and trajectories, while a structure-preserving image editor adapts simulated RGB observations toward real-world appearance. During training, future-depth prediction provides geometric guidance through asymmetric joint attention; at deployment, PAIMON directly predicts actions from RGB observations, ego-motion history, and language instructions without depth prediction or video rollout. We further introduce the ATP, which unifies expert-data collection and closed-loop evaluation across four simulation environments, including high-quality scenes reconstructed from real captures. Experiments show that removing future-depth supervision or appearance adaptation substantially degrades tracking performance, while increasing model capacity alone cannot recover these losses. PAIMON achieves the highest average success and tracking rates among the evaluated baselines, and real-world flight tests further demonstrate transfer to RGB-based closed-loop tracking under varying viewpoints, target scales, relative motion, and temporary occlusions.

AI USE STATEMENT

In this work, we used generative AI tools to edit images for data augmentation in the sim-to-real data generation pipeline. We also used generative AI tools to improve the grammar and readability of parts of the manuscript.

We did not use generative AI tools for developing theoretical models or conceptual frameworks, formulating or proving mathematical claims, proposing or refining hypotheses, designing or evaluating research methodology or experiments, implementing methods, translating, conducting qualitative analysis, or interpreting results.

The authors reviewed all AI-edited images for visual quality, semantic consistency, and suitability for the intended data augmentation setting. We also reviewed and edited all AI-assisted text to verify its accuracy and consistency with the actual work. We take responsibility for the final content of this work, including the text, data, and other artifacts produced with the aid of generative AI.

REFERENCES

Maximilian Adang, JunEn Low, Ola Shorinwa, and Mac Schwager. SINGER: An onboard generalist vision-language navigation policy for drones. *arXiv preprint arXiv:2509.18610*, 2025. URL https://arxiv.org/abs/2509.18610. 3, 7, 8, 9

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164. 7, 8, 9

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18392–18402, June 2023. doi: 10.1109/CVPR52729.2023.01764. URL https://openaccess.thecvf.com/content/CVPR2023/html/Brooks_InstructPix2Pix_Learning_To_Follow_Image_Editing_Instructions_CVPR_2023_paper.html. 3

Hyung Won Chung, Noah Constant, Xavier Garcia, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. UniMax: Fairer and more effective language sampling for large-scale multilingual pretraining. *arXiv preprint arXiv:2304.09151*, 2023. URL https://arxiv.org/abs/2304.09151. 6

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009. doi: 10.1109/CVPR.2009.5206848. 5

Alberto Dionigi, Simone Felicioni, Mirko Leomanni, and Gabriele Costante. D-VAT: End-to-end visual active tracking for micro aerial vehicles. *IEEE Robotics and Automation Letters*, 9(6):5046–5053, 2024. doi: 10.1109/LRA.2024.3385700. URL https://arxiv.org/abs/2308.16874. 2, 3

Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proceedings of the European Conference on Computer Vision*, pp. 375–391, September 2018. doi: 10.1007/978-3-030-01249-6_23. URL https://openaccess.thecvf.com/content_ECCV_2018/html/Dawei_Du_The_Unmanned_Aerial_ECCV_2018_paper.html. 3

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88 (2):303–338, 2010. doi: 10.1007/s11263-009-0275-4. 5

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1989–1998. PMLR, 2018. URL https://proceedings.mlr.press/v80/hoffman18a.html. 3

Xijie Huang, Weiqi Gai, Tianyue Wu, Congyu Wang, Zhiyang Liu, Xin Zhou, Yuze Wu, and Fei Gao. NavDreamer: Video models as zero-shot 3D navigators. *arXiv preprint arXiv:2602.09765*, 2026. URL https://arxiv.org/abs/2602.09765. 2, 3

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054. 7, 8, 9

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246. 3

Shuiwang Li, Yangxiang Yang, Dan Zeng, and Xucheng Wang. Adaptive and background-aware vision transformer for real-time uav tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13989–14000, 2023. doi: 10.1109/ICCV51070.2023.01286. URL https://openaccess.thecvf.com/content/ICCV2023/html/Li_Adaptive_and_Background-Aware_Vision_Transformer_for_Real-Time_UAV_Tracking_ICCV_2023_paper.html. 3

Yiming Li, Changhong Fu, Fangqiang Ding, Ziyuan Huang, and Geng Lu. AutoTrack: Towards high-performance visual tracking for UAV with automatic spatio-temporal regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11923–11932, 2020. doi: 10.1109/CVPR42600.2020.01194. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Li_AutoTrack_Towards_High-Performance_Visual_Tracking_for_UAV_With_Automatic_Spatio-Temporal_CVPR_2020_paper.html. 3

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. URL https://arxiv.org/abs/2511.10647. 6

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48. 5

Junjie Lu, Xuewei Zhang, Hongming Shen, Liwen Xu, and Bailing Tian. You only plan once: A learning-based one-stage planner with guidance learning. *IEEE Robotics and Automation Letters*, 9(7):6083–6090, 2024. doi: 10.1109/LRA.2024.3399589. URL https://ieeexplore.ieee.org/document/10528860. 3

Junjie Lu, Yulin Hui, Xuewei Zhang, Wencan Feng, Hongming Shen, Zhiyu Li, and Bailing Tian. YOPOv2-Tracker: An end-to-end agile tracking and navigation framework from perception to action, 2025. URL https://arxiv.org/abs/2505.06923. 3

Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking via reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3286–3295. PMLR, 2018. URL https://proceedings.mlr.press/v80/luo18a.html. 3

Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking and its real-world deployment via reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(6):1317–1332, 2020. doi: 10.1109/TPAMI.2019.2899570. 3

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aBsCjcPu_tE. 3, 4, 21

Matthias Müller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for UAV tracking. In *European Conference on Computer Vision*, volume 9905 of *Lecture Notes in Computer Science*, pp. 445–461. Springer, 2016. doi: 10.1007/978-3-319-46448-0_27. URL https://matthias.pw/publication/benchmark. 3

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3803–3810, 2018. doi: 10.1109/ICRA.2018.8460528. URL https://xbpeng.github.io/projects/SimToReal/SimToReal_2018.pdf. 3

Fereshteh Sadeghi, Alexander Toshev, Eric Jang, and Sergey Levine. Sim2Real viewpoint invariant visual servoing by recurrent control. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4691–4699, 2018. doi: 10.1109/CVPR.2018.00493. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Sadeghi_Sim2Real_Viewpoint_Invariant_CVPR_2018_paper.html. 3

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9338–9346, 2019. doi: 10.1109/ICCV.2019.00943. URL https://openaccess.thecvf.com/content_ICCV_2019/papers/Savva_Habitat_A_Platform_for_Embodied_AI_Research_ICCV_2019_paper.pdf. 4

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *European Conference on Computer Vision*, pp. 746–760. Springer, 2012. doi: 10.1007/978-3-642-33715-4_54. 5

Haowei Sun, Jinwu Hu, Zhirui Zhang, Haoyuan Tian, Xinze Xie, Yufeng Wang, Xiaohua Xie, Yun Lin, Zhuliang Yu, and Mingkui Tan. Open-world drone active tracking with goal-centered rewards. In *Advances in Neural Information Processing Systems 38*, 2025. URL https://openreview.net/forum?id=Ly2wXKIByI. 2

Nianyi Sun, Jin Zhao, Qing Shi, Chang Liu, and Peng Liu. Moving target tracking by unmanned aerial vehicle: A survey and taxonomy. *IEEE Transactions on Industrial Informatics*, 20(5): 7056–7068, 2024. doi: 10.1109/TII.2024.3363084. 2

Xiaolou Sun, Wufei Si, Wenhui Ni, Yuntian Li, Dongming Wu, Fei Xie, Runwei Guan, He-Yang Xu, Henghui Ding, Yuan Wu, Yutao Yue, Yongming Huang, and Hui Xiong. AutoFly: Vision-language-action model for UAV autonomous navigation in the wild. *arXiv preprint arXiv:2602.09657*, 2026. URL https://arxiv.org/abs/2602.09657. 3

Celine Teuliere and Eric Marchand. A dense and direct approach to visual servoing using depth maps. *IEEE Transactions on Robotics*, 30(5):1242–1249, 2014. doi: 10.1109/TRO.2014.2325991. 3

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 23–30, 2017. doi: 10.1109/IROS.2017.8202133. 3

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. URL https://arxiv.org/abs/2503.20314. 6, 7, 23

Shaoan Wang, Jiazhao Zhang, Minghan Li, Jiahang Liu, Anqi Li, Kui Wu, Fangwei Zhong, Junzhi Yu, Zhizheng Zhang, and He Wang. TrackVLA: Embodied visual tracking in the wild. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4139–4164. PMLR, 2025. URL https://proceedings.mlr.press/v305/wang25f.html. 3, 4, 5, 7, 15, 25

Yuze Wu, Mo Zhu, Xingxing Li, Yuheng Du, Yuxin Fan, Wenjun Li, Zhichao Han, Xin Zhou, and Fei Gao. Vla-an: An efficient and onboard vision-language-action framework for aerial navigation in complex environments, 2025. URL https://arxiv.org/abs/2512.15258. 3

Ziyuan Xia, Jingyi Xu, Chong Cui, Yuanhong Yu, Jiazhao Zhang, Qingsong Yan, Tao Ni, Junbo Chen, Xiaowei Zhou, Hujun Bao, Ruizhen Hu, and Sida Peng. Habitat-gs: A high-fidelity navigation simulator with dynamic gaussian splatting. *arXiv preprint arXiv:2604.12626*, 2026. URL https://arxiv.org/abs/2604.12626. 4, 5, 7, 15, 25

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666. 3

Andy Zhai, Brae Liu, Bruno Fang, Chalse Cai, Ellie Ma, Ethan Yin, Hao Wang, Hugo Zhou, James Wang, Lights Shi, et al. Igniting VLMs toward the embodied space. *arXiv preprint arXiv:2509.11766*, 2025. URL https://arxiv.org/abs/2509.11766. 7, 8, 9

Pengyu Zhang, Jie Zhao, Dong Wang, Huchuan Lu, and Xiang Ruan. Visible-thermal uav tracking: A large-scale benchmark and new baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8886–8895, 2022. doi: 10.1109/CVPR52688.2022.00868. URL https://openaccess.thecvf.com/content/CVPR2022/html/Zhang_Visible-Thermal_UAV_Tracking_A_Large-Scale_Benchmark_and_New_Baseline_CVPR_2022_paper.html. 3

Qiyao Zhang, Shuhua Zheng, Jianli Sun, Chengxiang Li, Xianke Wu, Zihan Song, Zhiyong Cui, Yisheng Lv, and Yonglin Tian. UAV-Track VLA: Embodied aerial tracking via vision-language-action models. *arXiv preprint arXiv:2604.02241*, 2026. URL https://arxiv.org/abs/2604.02241. 2, 3

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. URL https://arxiv.org/abs/2304.13705. 7, 8, 9

Fangwei Zhong, Xiao Bi, Yudi Zhang, Wei Zhang, and Yizhou Wang. Rspt: reconstruct surroundings and predict trajectory for generalizable active object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 3705–3714, 2023. 3

Fangwei Zhong, Kui Wu, Churan Wang, Hao Chen, Hai Ci, Zhoujun Li, and Yizhou Wang. Unreal-Zoo: Enriching photo-realistic virtual worlds for embodied ai. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5769–5779. IEEE, 2025. 7

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html. 3

A Data Collection and Sim-to-Real Transfer Details

A.1 Data Collection

Shared mode semantics. We use the same three modes in both subsets. Single-Target Tracking (STT) contains the designated target without distractors. Distractor Tracking (DT) adds pedestrians, so the instruction must identify the target within a crowd. Ambiguous / Appearance Tracking (AT) uses a generalized instruction, such as "follow the first person," rather than a fixed appearance description; it also contains distractors and randomizes the target avatar independently for each episode. STT and DT therefore provide appearance-specific target instructions, whereas AT tests whether the instruction can be separated from a particular target appearance.

HM3D subset. We start from HM3D tracking episodes (Wang et al., 2025), inheriting their reconstructed scenes, target routes, distractor configurations, and natural-language instructions. We replace the original ground-level observation with an independently controlled aerial camera and generate the corresponding aerial camera poses and action labels. The target is an articulated SMPL-X humanoid from a pool of 100 assets and moves online by oracle navigation over the scene navmesh; DT/AT distractors navigate to randomly sampled reachable goals. The simulated robot is frozen and hidden during rendering. HM3D follows the official TrackVLA scene partition: 703 scenes for training and 101 nominal validation scenes, with zero overlap. For STT/DT, 70 target identities are used for training and a disjoint 30 for validation, while distractors may be drawn from the full 100-person pool; AT randomizes the target avatar per episode.

We collect expert trajectories with a rule-based aerial-camera controller that computes the camera pose frame by frame. Its rear reference follows the target's smoothed frame-to-frame motion direction rather than its body orientation. When the target is stationary in the first frame, the controller predicts the initial direction from the shortest path on the navmesh. It obtains a horizontal camera anchor by moving backward along the target's walked navmesh footprint, avoiding wall intersections at corners, then rotates the anchor by a fixed 35° side angle and reprojects it onto the navmesh. If the reprojected point is invalid or occluded, the controller falls back to the direct rear position. The camera follows the resulting anchor subject to a per-frame displacement bound.

Each episode is rendered at three camera heights, 2.2, 1.9, and 1.3 m above the target's feet, corresponding to high, medium, and low viewpoints. A small sinusoidal height variation provides aerial motion. When the camera becomes too close to the target or a distractor enters its horizontal neighborhood, the controller raises the camera subject to the available ceiling clearance. If the target is occluded, the camera moves along the navmesh toward the target to recover line of sight. A camera embedded in geometry falls back to an overhead view. The follow distance is sampled per episode from a truncated Gaussian with mean 2.5 m and is seeded by the episode identifier.

Episode generation first checks the reachability of the target goals on the navmesh. It then samples a follow distance and renders the first frame, retrying with a shorter distance if the camera is embedded or the target occupies fewer than 1,000 pixels. The main loop runs for at most 400 frames and records RGB, depth, panoptic segmentation, the target box and mask, camera pose, and the future camera trajectory. We detect camera–geometry intersections with four horizontal 3D rays; at least three rays hitting static geometry mark the frame as embedded. We avoid a ground-plane projection test because high aerial viewpoints would otherwise be flagged throughout the episode. Under the HM3D zero-tolerance rule, any embedded frame, detected stall, or other invalid frame discards the entire episode. In the reported pipeline, the rejection rate decreases from approximately 72% in early collection to approximately 2.5% after the filtering procedure is stabilized.

Gaussian-splatting subset. We collect the GS subset with Habitat-GS (Xia et al., 2026) in scenes that include both indoor and outdoor environments, combining synthetic environments and InteriorGS reconstructions. The 110 training scenes contain 55 synthetic and 55 InteriorGS scenes; the 19 nominal validation scenes contain 9 synthetic and 10 InteriorGS scenes, with zero overlap. A virtual person follows a route sampled over the walkable region, and an aerial camera follows from an oblique rear viewpoint. The route contains a start point, intermediate waypoints, and an endpoint so that the person crosses the scene; motion synthesis converts it into a continuous walking animation. We prefer walking durations longer than approximately 10 seconds. GS uses six Gaussian avatars, and the same six-avatar pool is shared across training and validation scenes.

Table 6: Statistics for retained HM3D and Habitat-GS episodes. Training rows give source training-scene counts; validation rows report scenes covered by retained episodes. Durations use 10 fps for HM3D and 20 fps for Habitat-GS. Episode duration is mean ± standard deviation followed by median [IQR]; edited-frame counts cover sampled edited observations, not full videos.

Split	Scenes	Episodes	Raw frames	Duration (h)	Episode duration (s)	Edited frames
HM3D train	703	41,638	3,950,193	109.73	9.49 ± 4.27; 8.30 [6.30–11.30]	212,672
HM3D validation	80	2,000	169,843	4.72	8.49 ± 4.09; 7.40 [6.00–9.60]	9,359
HM3D subtotal	–	**43,638**	**4,120,036**	**114.45**	–	**222,031**
GS train	110	19,241	5,427,958	75.39	14.11 ± 10.73; 12.95 [10.50–14.80]	224,801
GS validation	18	200	75,458	1.05	18.86 ± 5.81; 20.65 [16.25–23.40]	3,653
GS subtotal	–	**19,441**	**5,503,416**	**76.44**	–	**228,454**
Training total	–	**60,879**	**9,378,151**	**185.12**	–	**437,473**
Validation total	–	2,200	245,301	5.77	–	13,012
All retained episodes	–	63,079	9,623,452	190.88	–	450,485

The route is the GS collection unit, and we select expert routes with a rule-based oversample-and-filter procedure. Each route is reused across three modes and three camera heights, producing nine episodes with shared target motion and controlled variation in distractors, instruction specificity, avatar appearance, and viewpoint. Routes are filtered by target in-frame fraction, walking duration, and invalid-frame count; tiny traversable regions and implausibly distorted long paths are removed. If a compact scene cannot meet its quota, the duration threshold is relaxed and recorded. The retained data are organized by mode, height, scene, and route. Table 6 summarizes the resulting dataset.

Figures 5 and 6 show one representative view per scene. The examples in Fig. 7 use the same medium camera-height setting, so mode differences are not confounded with viewpoint height.



Figure 5: Habitat-GS scene coverage across the 110 training scenes. Each tile shows a representative view of one scene.

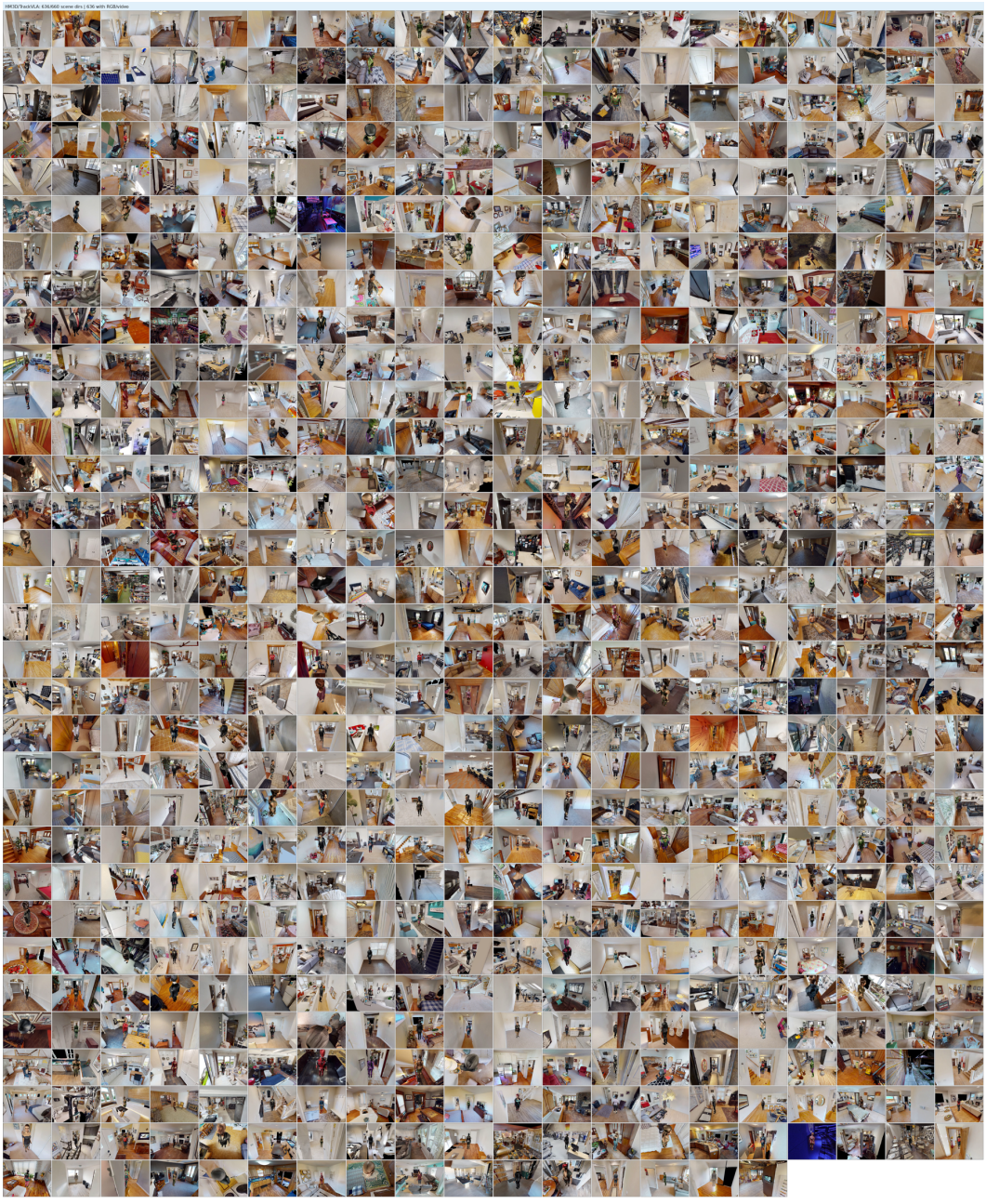

Figure 6: HM3D scene coverage across the retained scenes. Each tile shows a representative indoor layout used for aerial tracking.

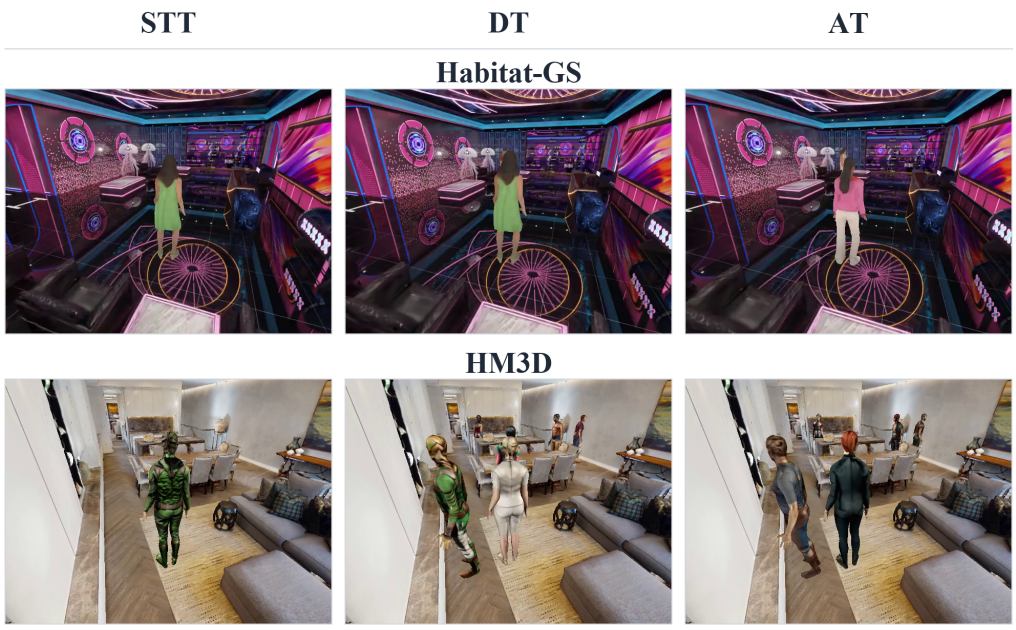

Figure 7: Representative RGB observations from Habitat-GS and HM3D across the STT, DT, and AT modes. Appearance-edited counterparts are shown in Fig. 8.

A.2 Annotation and Training Labels

Each retained episode contains an RGB video, depth, a per-frame target bounding box, camera pose, and the natural-language target instruction. In HM3D, target boxes and masks are extracted from panoptic segmentation. Camera pose is stored as 6-DoF in the FLU convention with the first camera frame as origin: X points forward, Y left, and Z up. HM3D also stores future camera waypoints, auxiliary distances, and the structured `camera_setup` description; GS provides the shared RGB/depth, target-box, and pose records.

For training, the recorded pose sequence is converted to the present-ego action representation. Roll is omitted under the zero-roll convention, giving five-dimensional increments $[d^x, d^y, d^z, \Delta\text{pitch}, \Delta\text{yaw}]$ with angular values in radians. The collector records ten future camera waypoints, which are converted during preprocessing to the twelve five-dimensional increments used by the active planner. The Gaussian-splatting episodes provide the same core RGB, depth, target-box, and 6-DoF pose records. These aligned labels preserve the correspondence between each visual observation, the target identity, and the camera motion.

A.3 Sim-to-Real Transfer

Objective and editor. The sim-to-real stage bridges the simulator's geometric supervision and the appearance of real drone imagery. We translate sampled synthetic RGB observations into photorealistic images while preserving the scene layout, camera angle, geometry, object and person positions, pose, scale, and person count. The transformation changes surface appearance, textures, materials, illumination, color rendering, and the tracked person's appearance, while the simulator-derived depth, pose, target boxes, and action labels remain paired with the transformed observation.

We use the frozen FLUX.2 Klein-9B distilled rectified-flow image editor with its native Qwen3-8B text encoder. The editor operates in a 128-channel VAE latent space with spatial downsampling by 16, uses the native four-step schedule with unit guidance, and runs in bfloat16 without fine-tuning or LoRA adaptation. The source observation is supplied both as a Kontext-style reference token sequence and as the latent used to initialize denoising.

Quantitative appearance-domain audit. We measure the direction of the appearance shift against held-out real-image reference domains with Fréchet Inception Distance (FID). For image sets S and R, we use the standard Inception-v3 pool3 definition

$$\text{FID}(S, R) = \|\mu_S - \mu_R\|_2^2 + \text{tr}\Big(\Sigma_S + \Sigma_R - 2(\Sigma_S\Sigma_R)^{1/2}\Big),$$

and define the positive-oriented edit gain as

$$G = \text{FID}(S_{\text{original}}, R) - \text{FID}(S_{\text{appearance transfer}}, R).$$

Under this definition, a positive G indicates a lower post-edit FID. FID measures appearance-distribution distance only; it does not evaluate geometry, depth, pose, boxes, actions, or tracking.

Figure 8 provides qualitative examples of layout and target preservation, complementing the distribution-level FID comparison in Table 1 in the main text.

Pixel non-degeneracy check. As a separate audit, Table 7 checks that the editor produces a non-trivial appearance transformation rather than a duplicate image. We compare 279 Habitat-GS pairs and 45 HM3D pairs. No pair is pixel-identical; the mean absolute RGB change is 16.39 and 18.69 intensity levels, respectively, while 99.82% and 98.19% of pixels change in at least one channel. These measurements are not realism or sim-to-real metrics, and are therefore kept separate from the FID evidence above.

Table 7: Pixel-level differences between paired source and appearance-edited RGB observations.

Quantity	Habitat-GS	HM3D
Paired RGB observations audited	279	45
Pixel-identical pairs	0	0
Mean $\lvert\Delta\mathrm{RGB}\rvert$	16.39	18.69
Changed-pixel fraction	99.82%	98.19%

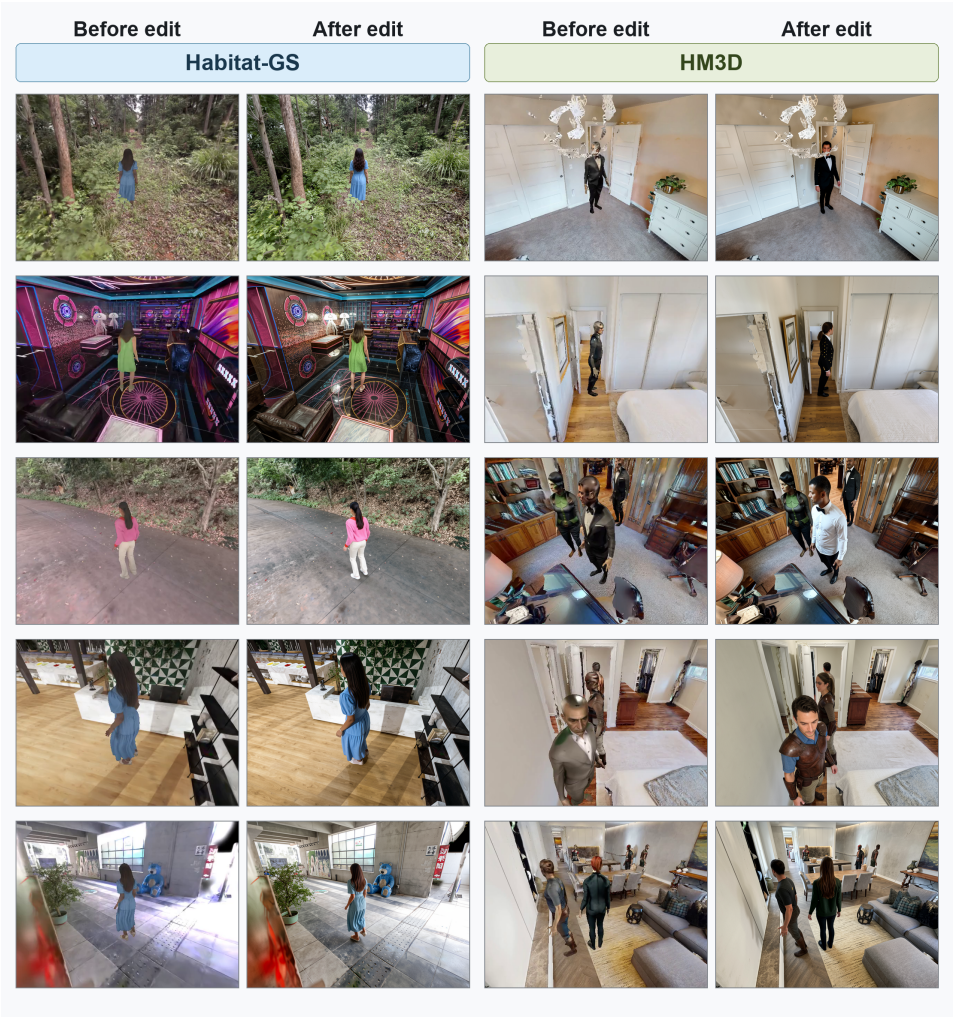

Figure 8: Qualitative examples of structure-preserving appearance transfer. Each pair shows a simulator observation and its edited counterpart, with scene geometry, layout, and target arrangement preserved.

SDEdit initialization. Starting the distilled editor from pure noise can change the tracked person's facing direction or the number of people. We instead use an SDEdit image-to-image initialization (Meng et al., 2022). Let x_0 denote the VAE latent of the source frame and let $\epsilon \sim \mathcal{N}(0, I)$. At time $t \in [0, 1]$, where $t = 1$ is pure noise, the flow interpolation is

$$x(t) = (1 - t)x_0 + t\epsilon.$$

We start denoising from a high noise level near $t = 0.96$. This leaves low-frequency structure in the latent while providing enough freedom to replace the synthetic surface appearance. Lower start values retain more of the original render and leave a stronger computer-generated appearance.

On-grid operation of the distilled sampler. The distilled model is trained at a discrete, resolution-dependent noise schedule rather than at arbitrary continuous start times. For the 640×480 frames used here, the schedule is

$$[1.0000,\ 0.9587,\ 0.8855,\ 0.7205,\ 0.0000].$$

We snap the requested strength to its nearest nonterminal node and run the native remaining tail:

$$i^* = \arg\min_i |\sigma_i - s|, \qquad t_{\text{start}} = \sigma_{i^*}, \qquad \{t\} = \{\sigma_{i^*}, \sigma_{i^*+1}, \ldots, \sigma_N\}, \quad \sigma_N = 0.$$

Here s is the requested strength and the terminal zero is retained as the final denoising time. Our setting $s = 0.96$ therefore snaps to $t_{\text{start}} = 0.9587$ and uses the native three-step tail. This prevents an off-grid initialization and keeps the denoising step count tied to the distilled schedule.

Metadata-conditioned prompts. We use a metadata-conditioned prompt-engineering procedure for every edited frame. A prompt template asks the editor to produce a natural photograph of the same scene and to keep the exact layout, camera viewpoint, geometry, perspective, object positions, person poses, and scale unchanged. The target's image position is summarized by a 3×3 grid from its bounding box. Its camera-relative orientation is estimated from the angle between the person's finite-difference heading in the world x–z plane and the camera-to-person direction. Angles below 50° are described as back-facing, angles above 130° as front-facing, and intermediate angles as side-facing. When the motion magnitude is below 0.05 m, the orientation clause is omitted. The prompt explicitly forbids rotating, adding, removing, or moving people.

If the instruction specifies clothing, the clothing phrase is extracted and used to dress the tracked person in the edited image. Generic clothing is used otherwise. The prompt ends with a compact realism description covering materials, scene-appropriate lighting, and a real-camera appearance. The resulting textual conditions are consistent with the structural constraints preserved by SDEdit.

Frame sampling and execution. Videos are recorded at 10 fps, but the editor processes one frame every 2 seconds, at indices $0, 20, 40, \ldots$. An index is retained only when at least 20 future frames remain, providing a 2-second future horizon for the downstream training pair. This produces approximately 13 editable frames per episode on average. Each edited frame is decoded from the source video, encoded by the VAE, initialized with the fixed seed 42, denoised with the three-step on-grid tail and unit guidance, decoded to 8-bit RGB, and written to the per-episode edited-frame directory.

For large-scale generation, one persistent worker keeps the 52 GB editor resident on each H20 GPU. Episodes are claimed across shared-filesystem workers through atomic exclusive-creation markers, and existing frame outputs are skipped on restart. Source-frame extraction is prefetched with a single FFmpeg decode pass, while output PNGs are written through temporary files followed by atomic renaming. The measured throughput is approximately 0.47 edited images per second per H20 GPU at the three-step setting.

B Method Details

This appendix records implementation details that are not needed to follow the method in Section 4. Table 8 gives the fine-grained module specification; the following sections provide the tensor layout, positional encodings, decoder parameterization, sampler, and optimization settings.

B.1 Tensor Interface and Positional Encoding

The training sequence is

$$[R \mid D_0, D_1, D_2, D_3 \mid A_0, \ldots, A_7 \mid E], \tag{11}$$

where each latent frame contributes 320 tokens. Thus, the training sequence has 1600 video tokens and nine action-side tokens. At inference, the depth-target tokens are removed and only the clean present latent R is retained, giving 320 video tokens and 329 joint tokens. Queries, keys, and values are projected to the shared 1536-dimensional attention space and then projected back to the video width 1536 or action width 1024. Joint attention uses 12 heads of width 128 for 30 blocks; the video and action streams use 3D and 1D RoPE, respectively. The video frame indices are $[0, 0, 1, 2, 3]$, so the present RGB latent and the present-time depth latent share a temporal phase. The action outputs have shapes $\text{reg} \in \mathbb{R}^{B \times 8 \times 12 \times 5}$ and $\text{logits} \in \mathbb{R}^{B \times 8}$.

Table 8: **Fine-grained PAIMON architecture.** Shapes include the batch dimension B. The ruled blocks explicitly separate the video expert, action expert, and cross-attention/expert-interaction modules. The 1280 depth tokens are present only during training.

Stage	Module	Exact configuration	Output / state
A. Video Expert			
Input tokens	Present VAE	Frozen causal 3D VAE; $3 \to 16$ channels, spatial $\times 1/8$	$[B, 16, 1, 32, 40]$
Target tokens	Depth VAE	13 depth maps mapped to three channels; spatial $\times 1/8$, temporal $\times 1/4$	$[B, 16, 4, 32, 40]$
Tokenization	Patch embed.	$\mathrm{Conv3D}(16, 1536)$, kernel/stride $(1, 2, 2)$; flatten $5 \times 16 \times 20$	$[B, 1600, 1536]$
Diffusion cond.	Time/AdaLN	sinusoidal-256 $\to$ MLP $256 \to 1536 \to 1536$; projection $1536 \to 9216$	token-wise 6×1536
$30\times$ block	Video FFN	AdaLN-gated Linear–GELU–Linear, $1536 \to 8960 \to 1536$	$[B, 1600, 1536]$
Output	Flow head	LN + 2-way AdaLN; Linear $1536 \to 64$; unpatchify and discard the clean RGB position	$[B, 16, 4, 32, 40]$
B. Action Expert			
Mode tokens	Trajectory encoder	Per-scalar sinusoidal PE (32-D); flatten $12 \times 5 \times 32 = 1920$; Linear–ReLU–LN–Linear, $1920 \to 1024 \to 1024$	$[B, 8, 1024]$
Ego token	Motion-history MLP	Flatten $39 \times 5 = 195$; Linear–SiLU–Linear, $195 \to 1024 \to 1024$	$[B, 1, 1024]$
Diffusion cond.	Time/AdaLN	sinusoidal-256 $\to$ MLP $256 \to 1024 \to 1024$; projection $1024 \to 6144$	sequence-wise 6×1024
$30\times$ block	Action FFN	AdaLN-gated Linear–GELU–Linear, $1024 \to 4096 \to 1024$	$[B, 9, 1024]$
Output	Regression head	Two $1024 \to 1024$ ReLU layers + Linear $1024 \to 60$; positional residual / 4 $\tanh$ angular decoding	$[B, 8, 12, 5]$
Output	Score head	$[\mathrm{Linear}(1024, 1024)$–ReLU–LN$] \times 2$; Linear $1024 \to 1$	$[B, 8]$
C. Cross-Attention and Expert Interaction			
Language ctx.	Text projection	Frozen umT5-XXL; Linear–GELU–Linear, $4096 \to 1024 \to 1024$	$[B, 128, 1024]$
History ctx.	DA3 backbone	Eight grayscale frames; frozen DA3 ViT-L features on an 18×23 patch grid	$[B, 8, 2048, 18, 23]$
History ctx.	History tokenizer	1×1 reduce $2048 \to 512$; three 3×3 convs to 1024; adaptive pool 2×3 + temporal/spatial PE	$[B, 48, 1024]$
Joint interface	Sequence sizes	1609 tokens during training; 329 tokens at inference after removing depth-target tokens	train / infer
$30\times$ coupling	Joint self-attn.	Expert-specific $q/k/v$; video 3D RoPE and action 1D RoPE; concatenate in 1536-D, 12-head masked SDPA, split, expert-specific output projections	video + action states
$30\times$ action cond.	History cross-attn.	12 heads of width 128; history tokens update all nine action tokens through a residual branch	$[B, 9, 1024]$
$30\times$ action cond.	Language cross-attn.	Independent 12-head residual branch; instruction tokens update all nine action tokens	$[B, 9, 1024]$
D. Parameter Summary			
Trainable	All modules	Video expert: 1.419B; action expert: 0.841B; history tokenizer: 17.6M	2.277B trainable parameters in total

B.2 Video Expert

The frozen Wan VAE (Wan et al., 2025) processes 256×320 inputs into $16 \times 32 \times 40$ latents. The 13-position depth clip is temporally compressed into four depth latents. Before encoding, depth is clipped to $[0, 10]$ m, invalid or sky pixels are filled with the far-depth value, and the map is copied to three channels. A $(1, 2, 2)$ patch embedding produces 320 tokens per latent frame. The internal video mask is `first_frame_causal`: R reads only R, while depth tokens read R and all depth tokens. The 3D RoPE dimensions for frame, height, and width are split as 44/42/42. The flow head returns $[B, 16, 4, 32, 40]$; the clean R position is excluded from the video loss.

B.3 Action Expert and Conditional Contexts

Each noisy 12×5 trajectory is folded into one mode token. A scalar-wise sinusoidal expansion of dimension 32 is followed by

$$\mathrm{Linear}(1920, 1024) \to \mathrm{ReLU} \to \mathrm{LayerNorm} \to \mathrm{Linear}(1024, 1024). \tag{12}$$

There is no anchor-ID embedding: mode identity is represented by the distinct anchor-centered noisy trajectory. The ego token is obtained from the physical, unnormalized 39×5 camera-motion history using a $195 \to 1024 \to 1024$ MLP; cold-start slots use the sentinel value -300.

Frozen umT5-XXL features have shape 128×4096 and are projected to 128×1024 instruction tokens. The preceding eight grayscale frames produce frozen DA3/DINOv2 ViT-L features with 2048 channels; g_{hist} reduces these features and adaptively pools a 2×3 grid, yielding 48 history tokens with temporal and spatial positional embeddings. These two context streams use independent residual cross-attention branches over the nine action tokens.

The action decoder predicts a correction δ and one score per mode. Position dimensions use residual decoding and angular dimensions use a bounded output:

$$\hat{y}_{0,\text{pos}} = x_{\tau_a,\text{pos}} + \delta_{\text{pos}}, \qquad \hat{y}_{0,\text{ang}} = 4\tanh(\delta_{\text{ang}}). \tag{13}$$

The regression head is zero-initialized so that its initial prediction follows the anchor.

B.4 Joint Attention Mask

The implementation uses the visibility sets in Eq. 8 with the token order in Eq. 11. This gives 1609 tokens in training and 329 tokens at inference. Language and history tokens remain outside this sequence and are injected by the two action-side cross-attention branches.

B.5 Diffusion, Objectives, and Optimization

The video branch uses continuous flow matching with 1000 training timesteps and shift 5.0. The action branch uses an anchor-centered truncated DDIM scheduler with $t_{\max} = 50$ and $\bar{\alpha}_{\min} = 0.99$. Inference uses the deterministic schedule $[50, 25, 0]$ for all eight modes in parallel and reads the final-step scores. For each transition from τ to $\tau' < \tau$, the DDIM scheduler estimates the noise and updates the anchor-centered trajectory as follows:

$$\hat{\epsilon}^{(m)} = \frac{\boldsymbol{X}_\tau^{(m)} - \sqrt{\bar{\alpha}_\tau}\,\hat{\boldsymbol{Y}}_0^{(m)}}{\sqrt{1-\bar{\alpha}_\tau}}, \qquad \boldsymbol{X}_{\tau'}^{(m)} = \sqrt{\bar{\alpha}_{\tau'}}\,\hat{\boldsymbol{Y}}_0^{(m)} + \sqrt{1-\bar{\alpha}_{\tau'}}\,\hat{\epsilon}^{(m)}. \tag{14}$$

The training objective is

$$\mathcal{L} = \lambda_{\text{video}}\mathcal{L}_{\text{video}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}}, \qquad (\lambda_{\text{video}}, \lambda_{\text{reg}}, \lambda_{\text{cls}}) = (1.0, 0.15, 1.0). \tag{15}$$

The target mode, regression term, and independent per-anchor sigmoid focal BCE are defined in Eqs. 9 and 10, with $\gamma = 2$. The implementation evaluates the classification term from logits with a numerically stable binary-cross-entropy routine.

We train with gradient clipping 1.0 and batch size 16, using gradient checkpointing on four H20 GPUs. The shared optimizer, learning rates, schedule, training budget, and precision are summarized in Appendix C.1.

C Experiments

C.1 Training Configuration

All reported models share the same optimizer, learning-rate schedule, training budget, and precision. Table 9 summarizes these settings alongside run management and the PAIMON-specific initialization.

C.2 Closed-Loop Evaluation Environments

We evaluate closed-loop tracking in four complementary environments. HM3D and Habitat-GS provide simulated reference domains, GYM-unrealCV tests interactive control in Unreal Engine, and HQ-GS provides high-quality scenes reconstructed from real captures. At each step, the policy predicts a camera trajectory from the current observation and history, and the environment executes it before returning a new observation. This protocol evaluates the full perception–action loop.

Table 9: Shared optimization and run-management settings. The PAIMON-specific initialization is marked explicitly.

Setting	Configuration
Optimizer	AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay $= 10^{-2}$
Learning rates	Video trunk: 10^{-5}; action expert and history tokenizer: 10^{-4}
Schedule	Linear warmup for 2,000 steps (5% of training), followed by cosine decay toward zero
Training budget	40,000 stage-2 steps, starting at global step 0 without resuming a prior run
Precision	bfloat16
PAIMON initialization	Warm-start the video trunk from its pretrained checkpoint; initialize the action expert and heads from scratch
Gradient accumulation	1 step
Data loading	8 workers
Exponential moving average	Not used
Validation	Every 1,500 training steps
Checkpointing	Every 2,500 steps; save model weights without optimizer state

C.2.1 Training Environments

HM3D. HM3D provides reconstructed indoor environments with varied room layouts, corridors, furniture arrangements, and traversable regions (Wang et al., 2025). Its synchronized RGB, depth, camera poses, target annotations, and navigation structure support expert trajectories with geometric supervision. We vary target identities, distractors, viewpoints, and instructions to cover the tracking and camera-control behaviors required by the task.

Habitat-GS. Habitat-GS is a high-fidelity navigation simulator based on dynamic Gaussian splatting (Xia et al., 2026). It complements HM3D with synthetic and InteriorGS scenes spanning indoor and outdoor layouts. Its renderer produces detailed surfaces, vegetation, and scene-specific lighting, while the simulator provides aligned depth, pose, target-box, and action supervision. We use HM3D and Habitat-GS for training, held-out open-loop validation, and simulated closed-loop evaluation. HQ-GS adds an evaluation setting reconstructed from real captures.

C.2.2 GYM-unrealCV

GYM-unrealCV connects Unreal Engine environments to UnrealCV and the OpenAI Gym interface. Through a standard step loop, the agent receives an observation and issues an action, after which Unreal Engine updates the rendered view, object states, and interactions. This setting tests policy stability when actions change subsequent viewpoints and target visibility.

The benchmark contains four scenes: *SimpleRoom* for basic indoor tracking, *Garage* with occlusion and low light, *UrbanCity* with reflective urban roads, and *UrbanRoad* with a similar road layout and fewer obstacles. Together, they cover indoor and outdoor tracking under varied visibility and appearance. At each control step, the predicted trajectory moves the UnrealCV camera. The environment then updates the target and distractor states and returns the next rendered observation. This evaluation measures tracking performance under repeated replanning.

C.2.3 HQ-GS: High-Quality Scenes Reconstructed from Real Captures

HQ-GS provides high-quality scenes reconstructed from synchronized 4K panoramic-camera and LiDAR captures, combining wide-field appearance with metric geometry. Each location is represented as a 3D Gaussian Splatting (3D-GS) scene, from which observations are rendered as the policy changes viewpoint. The evaluation is simulated, while the scene appearance and geometry are derived from real captures.

Figure 9 shows the two reconstructions. Juzizhoutou contains open paved areas, lawns, railings, and a large sculpted landmark, whereas Qianfoshan contains dense foliage, traditional buildings, signage, and entrance structures.



Figure 9: Views of the high-quality HQ-GS reconstructions: Juzizhoutou in the top row and Qianfoshan in the bottom row, each from three viewpoints.

During closed-loop interaction, camera motion changes the viewpoint and subsequent observations within each reconstruction. This enables repeatable testing with captured scene appearance and geometry.

C.3 Closed-Loop Evaluation Metrics

We evaluate closed-loop aerial person tracking on ATP with episode-level and frame-level metrics. For episode e, let $\mathcal{I}_e$ be the model-controlled frames in the evaluation window and $L_e = |\mathcal{I}_e|$. Let $v_{e,i}$ denote target visibility and $d_{e,i}$ the evaluator-computed horizontal camera–target distance. We compute each episode's score independently, then average across episodes for each domain, model, and replanning frequency.

C.3.1 Primary Metrics

We report two complementary primary metrics: TR measures tracking success at the frame level, while SR measures success at the episode level.

Tracking rate (TR). For each frame, define the tracking-success indicator

$$q_{e,i}^{(3)} = \mathbf{1}[v_{e,i} = 1 \ \wedge \ d_{e,i} \leq 3.0 \text{ m}] \,. \tag{16}$$

The episode-level tracking rate and the domain-level tracking rate are

$$\text{TR}_e = \frac{1}{L_e} \sum_{i \in \mathcal{I}_e} q_{e,i}^{(3)}, \tag{17}$$

$$\text{TR} = \frac{1}{N} \sum_{e=1}^{N} \text{TR}_e, \tag{18}$$

Table 10: Auxiliary closed-loop metrics. Frame-level quantities are averaged within each episode and then across episodes.

Metric	Computation	Meaning	Direction
VR	$L_e^{-1} \sum_i v_{e,i}$	Target visibility without a distance constraint	↑
Survive	Frames before the first run of 10 invisible frames, divided by L_e, or 1 if no such run occurs	Time survived before declaring a loss	↑
Center err.	Box-center offset from the image center, normalized by half the image diagonal	Target centering (zero indicates centered)	↓
Occluded	Fraction with visible target pixels / box area < 0.4	Partial-occlusion frequency	↓
Navmesh	Fraction of frames with a wall-clipping event	Navmesh-violation frequency	↓
Standoff	$L_e^{-1} \sum_i \mathbf{1}[d_{e,i} < 1.2\,\mathrm{m}]$	Fraction of frames that are too close	↓
Speed	Mean camera speed in m/s	Motion magnitude	–
Jerk	Mean finite-difference camera jerk magnitude in m/s^3	Motion smoothness	↓
Kin. viol.	Fraction with $\lvert\dot{\psi}\rvert > 60°$/s or $\lvert\dot{\theta}\rvert > 30°$/s	Kinematic-limit violations	↓
Distractor	Fraction where the target is invisible and a distractor has center error < 0.25	Distractor-driven target loss	↓

where $N = 100$ for each domain. TR is the fraction of frames in which the target is visible and within 3.0 m horizontally. It lies in $[0, 1]$ and has no lower distance bound, so frames with $d < 1.2$ m also count as successful.

Success rate (SR). SR uses a wider distance threshold and assigns a binary success value to each episode. We first compute the fraction of qualifying frames:

$$r_e^{(4)} = \frac{1}{L_e} \sum_{i \in \mathcal{I}_e} \mathbf{1}[v_{e,i} = 1 \ \wedge \ d_{e,i} \le 4.0\,\mathrm{m}]\,. \tag{19}$$

The episode success value and the aggregated success rate are

$$y_e = \mathbf{1}\left[r_e^{(4)} > 0.80\right], \tag{20}$$

$$\mathrm{SR} = \frac{1}{N} \sum_{e=1}^{N} y_e. \tag{21}$$

The threshold is strict: an episode with exactly 80% qualifying frames is unsuccessful. SR is the fraction of episodes with more than 80% qualifying frames.

TR and SR summarize tracking at the frame and episode levels, respectively, using distance thresholds of 3.0 m and 4.0 m.

C.3.2 Auxiliary Diagnostics

We report auxiliary metrics for visibility, tracking geometry, and camera motion. Table 10 defines these metrics.

C.4 Closed-Loop Evaluation Results

The closed-loop baseline results are organized by scene source and replanning frequency. Tables 11–12 compare all models on the simulated Habitat-GS and HM3D environments at 2 Hz and 1 Hz. At 2 Hz, the model executes the linearly interpolated action at step 2.5. Tables 13–16 extend the same comparison to four interactive GYM-unrealCV scenes at 2 Hz, while Tables 17–18 cover the two high-quality HQ-GS reconstructions at 2 Hz. The shared metric columns make the model comparisons consistent across these settings. Representative qualitative comparisons are shown for Habitat-GS (Figure 10), HM3D (Figure 11), the GYM-unrealCV Garage scene (Figure 12), and HQ-GS Juzizhoutou (Figure 13). The TR displayed in each figure is specific to the illustrated episode; the tables report metrics aggregated over the evaluation episodes.

Table 11: Baseline closed-loop results on Habitat-GS (2 Hz and 1 Hz).

Model	Freq.	Simulation environment: Habitat-GS											
		TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Navmesh ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓	Distractor ↓
PAIMON(Ours)	2 Hz	0.95	0.92	0.96	0.96	0.10	0.02	0.06	0.06	1.42	58.50	0.04	0.01
	1 Hz	0.84	0.65	0.88	0.68	0.20	0.04	0.12	0.08	2.06	54.80	0.07	0.01
SINGER	2 Hz	0.64	0.39	0.66	0.66	0.28	0.02	0.10	0.27	1.59	26.70	0.02	0.02
	1 Hz	0.62	0.30	0.64	0.64	0.28	0.03	0.10	0.26	1.55	21.10	0.01	0.02
ACT	2 Hz	0.70	0.49	0.74	0.74	0.25	0.03	0.12	0.13	1.65	32.00	0.03	0.02
	1 Hz	0.67	0.42	0.69	0.69	0.30	0.05	0.09	0.19	1.63	28.70	0.03	0.01
WALL-OSS	2 Hz	0.89	0.96	0.97	0.98	0.09	0.03	0.08	0.02	1.61	156.90	0.11	0.00
	1 Hz	0.79	0.69	0.86	0.86	0.19	0.04	0.08	0.06	1.89	169.40	0.15	0.00
π_0	2 Hz	0.89	0.97	0.98	0.99	0.14	0.02	0.07	0.05	1.52	122.10	0.10	0.00
	1 Hz	0.82	0.80	0.91	0.92	0.19	0.04	0.07	0.07	1.65	112.50	0.10	0.00
$\pi_{0.5}$	2 Hz	0.85	0.87	0.95	0.95	0.14	0.03	0.08	0.05	1.48	118.70	0.08	0.00
	1 Hz	0.66	0.55	0.79	0.80	0.20	0.04	0.09	0.11	1.78	120.60	0.11	0.01

Table 12: Baseline closed-loop results on HM3D (2 Hz and 1 Hz).

		TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓	Distractor ↓
PAIMON(Ours)	2 Hz	0.74	0.67	0.84	0.86	0.19	0.16	0.07	1.48	65.90	0.13	0.02
	1 Hz	0.54	0.43	0.73	0.73	0.22	0.18	0.06	1.34	57.20	0.12	0.06
SINGER	2 Hz	0.53	0.37	0.69	0.70	0.27	0.19	0.09	1.03	31.70	0.03	0.03
	1 Hz	0.46	0.25	0.62	0.62	0.27	0.20	0.07	0.97	25.40	0.02	0.03
ACT	2 Hz	0.39	0.28	0.68	0.69	0.22	0.20	0.05	1.03	43.00	0.07	0.05
	1 Hz	0.39	0.27	0.70	0.70	0.22	0.20	0.03	1.00	40.80	0.05	0.03
WALL-OSS	2 Hz	0.56	0.48	0.76	0.76	0.15	0.18	0.05	1.62	181.30	0.26	0.05
	1 Hz	0.48	0.43	0.75	0.76	0.20	0.18	0.06	1.58	160.20	0.25	0.05
π_0	2 Hz	0.50	0.38	0.74	0.75	0.22	0.18	0.05	1.43	150.30	0.17	0.03
	1 Hz	0.42	0.30	0.67	0.68	0.25	0.21	0.05	1.36	129.20	0.16	0.06
$\pi_{0.5}$	2 Hz	0.40	0.26	0.68	0.67	0.22	0.19	0.06	1.39	171.60	0.23	0.04
	1 Hz	0.34	0.12	0.61	0.59	0.26	0.21	0.04	1.25	136.50	0.21	0.04

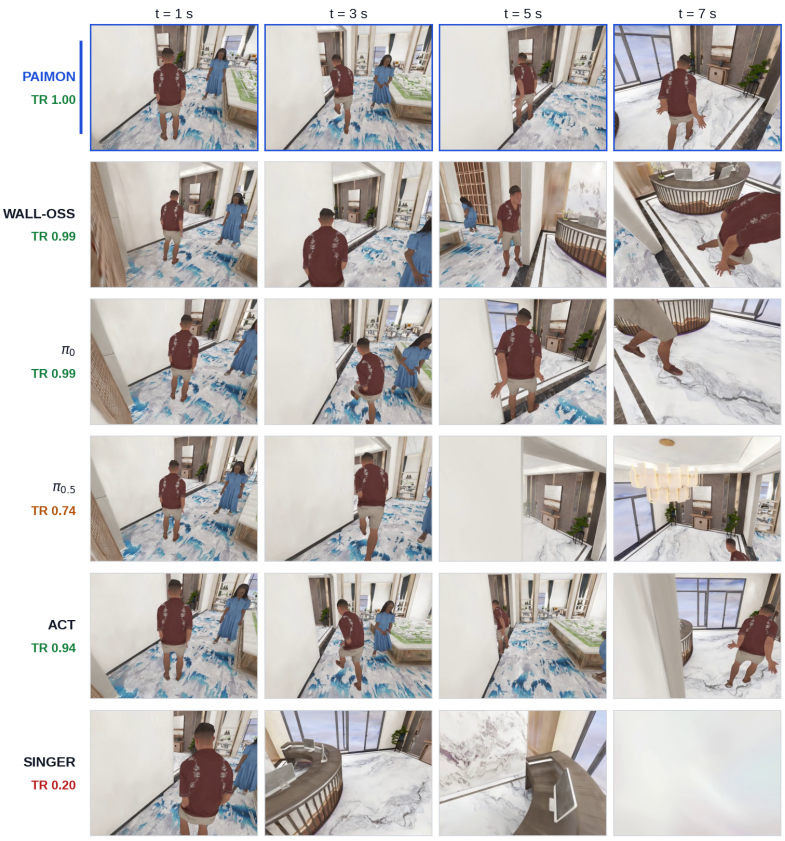

Figure 10: Qualitative baseline comparison on Habitat-GS. Columns show observations at $t = 1, 3, 5, 7$ s, and rows compare PAIMON with baseline policies. The displayed TR values are episode-specific.

C.4.1 GYM-UNREALCV CLOSED-LOOP RESULTS

Each GYM-unrealCV scene is evaluated on 100 test episodes at 2 Hz. Tables 13–16 show that PAIMON attains the highest TR in all four scenes, ranging from 0.84 to 0.93, and reaches SR$= 1.00$ on UrbanCity, UrbanRoad, and SimpleRoom. Garage is more challenging: its SR is 0.71, below WALL-OSS (0.86) and SINGER (0.76), although PAIMON retains the highest TR (0.84). The auxiliary columns show that navmesh and distractor rates remain low across the four scenes.

Table 13: Baseline closed-loop results on the GYM-unrealCV UrbanCity scene (2 Hz).

Model	Freq.	Simulation environment: GYM-unrealCV – UrbanCity											
		TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Navmesh ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓	Distractor ↓
PAIMON(Ours)	2 Hz	0.93	1.00	0.95	0.98	0.14	0.06	0.01	0.16	1.71	53.20	0.04	0.00
SINGER	2 Hz	0.63	0.94	0.97	0.98	0.13	0.07	0.01	0.04	1.53	19.90	0.01	0.00
ACT	2 Hz	0.60	0.36	0.73	0.73	0.25	0.07	0.01	0.15	1.86	27.10	0.03	0.03
WALL-OSS	2 Hz	0.84	0.85	0.93	0.95	0.14	0.10	0.03	0.09	1.82	139.30	0.15	0.00
π_0	2 Hz	0.74	0.82	0.92	0.93	0.15	0.10	0.01	0.08	1.89	118.00	0.13	0.00
$\pi_{0.5}$	2 Hz	0.55	0.65	0.92	0.94	0.17	0.08	0.00	0.05	1.82	114.20	0.09	0.01

Table 14: Baseline closed-loop results on the GYM-unrealCV UrbanRoad scene (2 Hz).

Model	Freq.	Simulation environment: GYM-unrealCV – UrbanRoad											
		TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Navmesh ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓	Distractor ↓
PAIMON(Ours)	2 Hz	0.90	1.00	0.95	0.98	0.15	0.06	0.01	0.14	1.69	52.90	0.05	0.00
SINGER	2 Hz	0.72	0.89	0.95	0.97	0.13	0.07	0.00	0.04	1.53	21.00	0.02	0.00
ACT	2 Hz	0.55	0.38	0.67	0.68	0.25	0.07	0.03	0.12	2.16	30.80	0.04	0.02
WALL-OSS	2 Hz	0.89	0.93	0.97	0.97	0.13	0.11	0.02	0.06	1.80	134.90	0.18	0.00
π_0	2 Hz	0.71	0.73	0.90	0.91	0.13	0.09	0.01	0.08	1.88	116.40	0.13	0.00
$\pi_{0.5}$	2 Hz	0.60	0.67	0.88	0.89	0.18	0.07	0.00	0.07	1.77	119.20	0.11	0.00

Table 15: Baseline closed-loop results on the GYM-unrealCV Garage scene (2 Hz).

Model	Freq.	Simulation environment: GYM-unrealCV – Garage											
		TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Navmesh ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓	Distractor ↓
PAIMON(Ours)	2 Hz	0.84	0.71	0.87	0.90	0.15	0.03	0.03	0.21	1.77	56.10	0.05	0.00
SINGER	2 Hz	0.77	0.76	0.89	0.91	0.15	0.03	0.02	0.09	1.53	30.40	0.03	0.00
ACT	2 Hz	0.53	0.32	0.70	0.70	0.23	0.06	0.05	0.13	2.13	45.00	0.06	0.00
WALL-OSS	2 Hz	0.71	0.86	0.94	0.94	0.14	0.10	0.02	0.05	1.83	160.00	0.18	0.00
π_0	2 Hz	0.61	0.55	0.82	0.83	0.17	0.07	0.04	0.11	1.93	140.30	0.16	0.01
$\pi_{0.5}$	2 Hz	0.48	0.47	0.83	0.85	0.18	0.06	0.08	0.05	1.72	130.60	0.12	0.01

Table 16: Baseline closed-loop results on the GYM-unrealCV SimpleRoom scene (2 Hz).

Model	Freq.	Simulation environment: GYM-unrealCV – SimpleRoom											
		TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Navmesh ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓	Distractor ↓
PAIMON(Ours)	2 Hz	0.91	1.00	1.00	1.00	0.11	0.04	0.00	0.08	1.70	216.20	0.10	0.00
SINGER	2 Hz	0.76	0.45	0.77	0.79	0.23	0.04	0.00	0.27	2.03	238.10	0.13	0.00
ACT	2 Hz	0.58	0.34	0.74	0.74	0.33	0.09	0.00	0.12	2.11	285.80	0.16	0.00
WALL-OSS	2 Hz	0.84	0.92	0.96	0.96	0.11	0.11	0.01	0.09	1.92	261.40	0.22	0.00
π_0	2 Hz	0.77	0.79	0.92	0.93	0.16	0.09	0.01	0.11	2.05	262.50	0.20	0.00
$\pi_{0.5}$	2 Hz	0.70	0.67	0.88	0.88	0.18	0.06	0.00	0.08	1.99	292.80	0.16	0.00

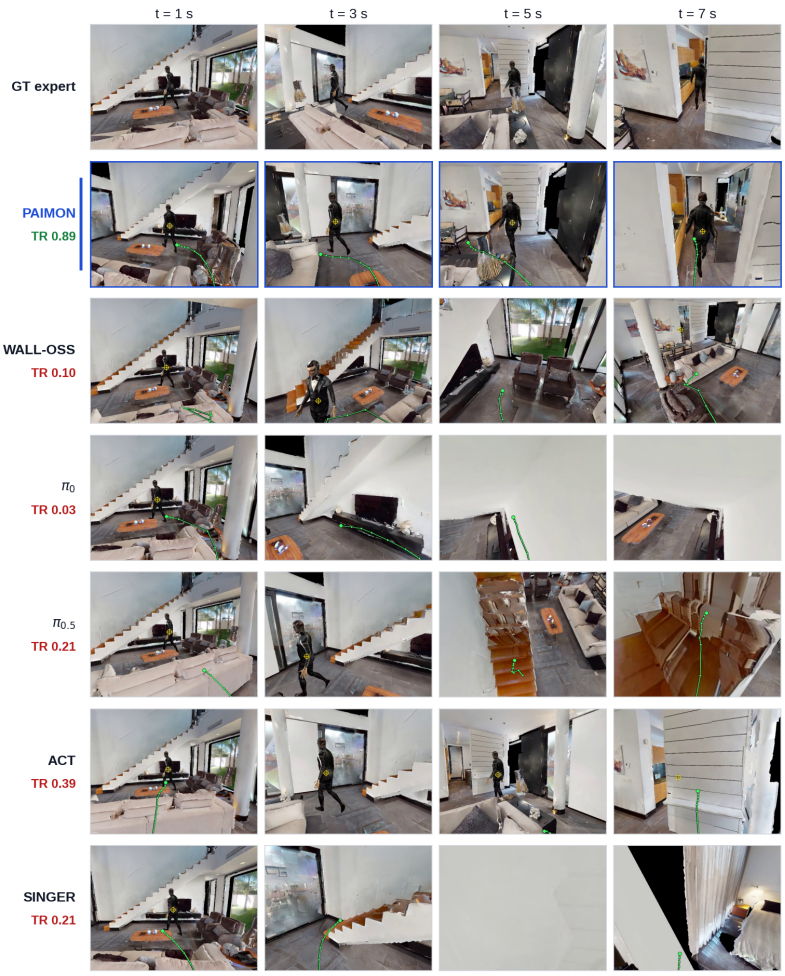

Figure 11: Qualitative baseline comparison on HM3D. Columns show observations at $t = 1, 3, 5, 7$ s, and rows show the expert reference, PAIMON, and baseline policies. The displayed TR values are episode-specific.

C.4.2 HQ-GS Closed-Loop Results

Each HQ-GS scene is evaluated on 100 test episodes at 2 Hz. Tables 17 and 18 show that PAIMON obtains the highest TR in both scenes, reaching 0.90 on Juzizhoutou and 0.79 on Qianfoshan, compared with 0.71 and 0.68 for the strongest baseline, WALL-OSS. SR varies with scene content: WALL-OSS is higher on Juzizhoutou (0.91 versus 0.86), whereas PAIMON leads on Qianfoshan (0.66 versus 0.59). The auxiliary metrics also show lower jerk for PAIMON than WALL-OSS in both scenes, while its kinematic-violation rates remain at 0.07 and 0.10, respectively.

C.5 Open-Loop Planning Metrics

We evaluate open-loop planning on the fixed HM3D and Habitat-GS validation sets described in Appendix A. For each observation, every model predicts a 12-step camera trajectory at 0.2 s intervals. We compare these predictions with the recorded expert trajectories without executing the predicted actions or conditioning on subsequent observations.

Let $\hat{\mathbf{p}}_{t,k}$ and $\mathbf{p}_{t,k}$ denote the predicted and expert relative positions at horizon step k. Let $\hat{\phi}_{t,k}$, $\phi_{t,k}$ and $\hat{\psi}_{t,k}$, $\psi_{t,k}$ denote the corresponding pitch and yaw angles. We report average displacement error

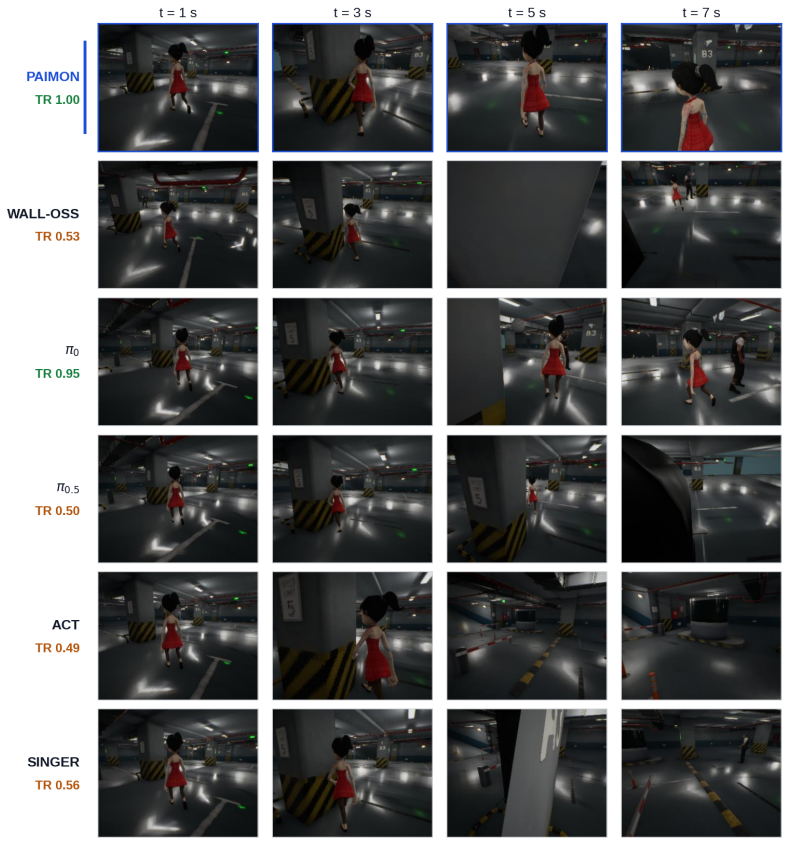

Figure 12: Qualitative baseline comparison on the GYM-unrealCV Garage scene. Columns show observations at $t = 1, 3, 5, 7$ s, and rows compare PAIMON with baseline policies. The displayed TR values are episode-specific.

Table 17: Baseline closed-loop results on the HQ-GS Juzizhoutou scene (2 Hz).

Model	Simulation environment: HQ-GS – Juzizhoutou									
	TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓
PAIMON (Ours)	0.90	0.86	0.94	0.94	0.11	0.03	0.04	2.03	64.90	0.07
SINGER	0.44	0.50	0.98	0.98	0.18	0.02	0.00	1.61	19.60	0.00
ACT	0.19	0.10	0.94	0.94	0.21	0.02	0.00	1.09	20.60	0.01
WALL-OSS	0.71	0.91	1.00	1.00	0.09	0.02	0.00	1.93	132.50	0.11
π_0	0.61	0.77	0.98	0.98	0.17	0.02	0.01	1.99	119.30	0.12
$\pi_{0.5}$	0.59	0.76	0.97	0.97	0.13	0.03	0.02	1.85	100.00	0.10

(ADE) over the prediction horizon and final displacement error (FDE):

$$\mathrm{ADE}_{\mathrm{pos}} = \frac{1}{T}\sum_{k=1}^{T} \|\hat{\mathbf{p}}_{t,k} - \mathbf{p}_{t,k}\|_2 , \qquad \mathrm{FDE}_{\mathrm{pos}} = \|\hat{\mathbf{p}}_{t,T} - \mathbf{p}_{t,T}\|_2 . \tag{22}$$

For pitch and yaw, we apply the same definitions to the wrapped angular difference $\Delta_\angle(a, b) = \mathrm{wrap}(a - b)$ and convert the resulting errors from radians to degrees:

$$\mathrm{ADE}_{\mathrm{ang}} = \frac{1}{T}\sum_{k=1}^{T} \left|\Delta_\angle(\hat{\theta}_{t,k}, \theta_{t,k})\right| , \qquad \mathrm{FDE}_{\mathrm{ang}} = \left|\Delta_\angle(\hat{\theta}_{t,T}, \theta_{t,T})\right| , \tag{23}$$

where θ denotes pitch or yaw. Lower values indicate more accurate trajectory predictions. Table 3 in the main text reports ADE and FDE for position, pitch, and yaw.

Table 18: Baseline closed-loop results on the HQ-GS Qianfoshan scene (2 Hz).

Model	Simulation environment: HQ-GS – Qianfoshan									
	TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓
PAIMON (Ours)	0.79	0.66	0.87	0.79	0.15	0.12	0.05	2.03	71.80	0.10
SINGER	0.34	0.34	0.80	0.74	0.22	0.12	0.01	1.63	20.30	0.02
ACT	0.23	0.26	0.77	0.69	0.20	0.09	0.00	1.57	29.40	0.02
WALL-OSS	0.68	0.59	0.83	0.79	0.14	0.13	0.02	2.52	260.30	0.26
π_0	0.42	0.41	0.71	0.72	0.19	0.10	0.03	2.33	189.70	0.22
$\pi_{0.5}$	0.48	0.40	0.74	0.70	0.18	0.11	0.03	2.15	166.20	0.20

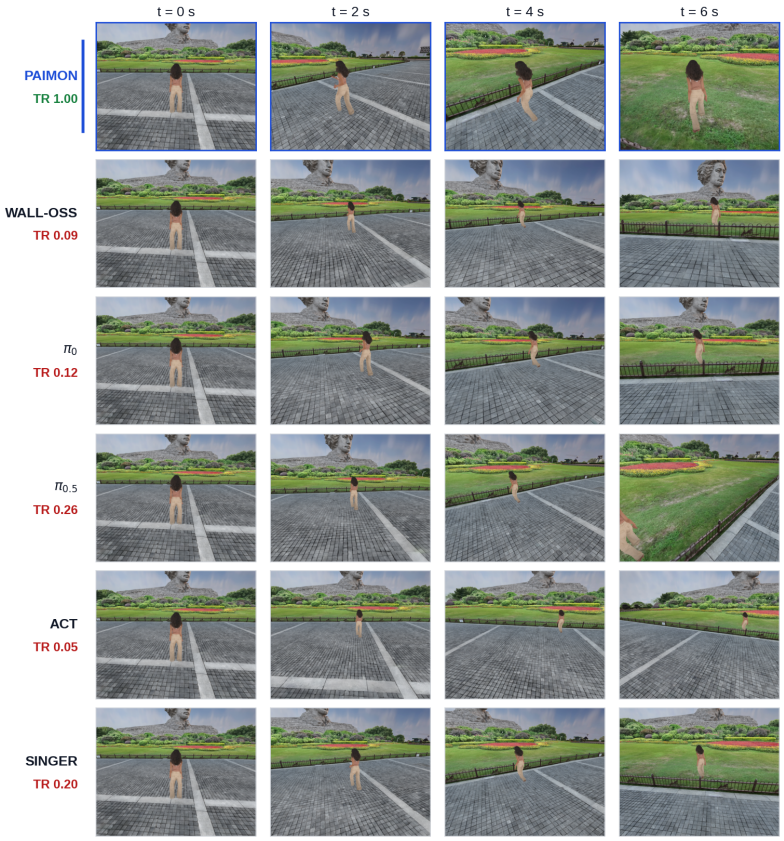

Figure 13: Qualitative baseline comparison on HQ-GS Juzizhoutou. Columns show observations at $t = 0, 2, 4, 6$ s, and rows compare PAIMON with baseline policies. The displayed TR values are episode-specific.

C.6 Ablation Study

C.6.1 Ablation Model Details

The reference model, PAIMON, uses an appearance-edited current RGB frame, a 1.3B video expert, and future-depth reconstruction as a training-time geometric signal. We compare three variants under the same task and evaluation settings to examine these design choices.

Action-only. This variant removes the video expert and its VAE pathway from PAIMON. It therefore neither predicts future depth nor receives the video-expert loss. The action expert uses the current RGB observation and motion history to predict the camera trajectory. This comparison

assesses the combined contribution of the video expert and its training-time geometric supervision to action planning.

PAIMON-Raw. This variant replaces the edited current RGB input with the original simulator-rendered frame while retaining the video expert and future-depth pathway. It tests the contribution of structure-preserving appearance transfer to sim-to-real generalization.

PAIMON–5B. This variant replaces the 1.3B video expert with a 5B video backbone to examine model-capacity scaling. Because the experts share an attention-head configuration at their joint interface, we also widen the action expert to preserve architectural compatibility. Thus, this variant increases the capacity of both experts.

C.6.2 Closed-Loop Ablation Results

The closed-loop ablation tables follow the same organization as the baseline results. Tables 19–20 compare PAIMON with three variants on the simulated Habitat-GS and HM3D environments at both frequencies. Tables 21–24 transfer this comparison to the four GYM-unrealCV scenes at 2 Hz, and Tables 25–26 evaluate the same variants on the two HQ-GS reconstructions at 2 Hz. This organization keeps the reference model and ablation variants fixed across the scene sources. Representative qualitative ablation comparisons are shown for Habitat-GS (Figure 14), HM3D (Figure 15), the GYM-unrealCV Garage scene (Figure 16), and HQ-GS Juzizhoutou (Figure 17). The TR displayed in each figure is specific to the illustrated episode; the tables report metrics aggregated over the evaluation episodes.

Table 19: Closed-loop ablation results on Habitat-GS (2 Hz and 1 Hz).

Model	Freq.	Simulation environment: Habitat-GS											
		TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Navmesh ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓	Distractor ↓
PAIMON(Ours)	2 Hz	0.95	0.92	0.96	0.96	0.10	0.02	0.06	0.06	1.42	58.50	0.04	0.01
	1 Hz	0.84	0.65	0.88	0.68	0.20	0.04	0.12	0.08	2.06	54.80	0.07	0.01
PAIMON–5B	2 Hz	0.93	0.88	0.94	0.95	0.13	0.03	0.07	0.06	1.51	60.60	0.04	0.00
	1 Hz	0.75	0.59	0.79	0.79	0.21	0.04	0.09	0.11	1.70	50.90	0.04	0.00
Action-only	2 Hz	0.81	0.71	0.82	0.83	0.13	0.02	0.07	0.10	1.72	65.60	0.13	0.01
	1 Hz	0.55	0.26	0.57	0.57	0.21	0.05	0.12	0.11	2.06	63.90	0.17	0.00
PAIMON-Raw	2 Hz	0.89	0.97	0.98	0.99	0.11	0.02	0.07	0.02	1.28	54.10	0.01	0.00
	1 Hz	0.74	0.68	0.83	0.83	0.18	0.04	0.08	0.08	1.47	44.70	0.01	0.01

Table 20: Closed-loop ablation results on HM3D (2 Hz and 1 Hz).

		TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓	Distractor ↓
PAIMON(Ours)	2 Hz	0.74	0.67	0.84	0.86	0.19	0.16	0.07	1.48	65.90	0.13	0.02
	1 Hz	0.54	0.43	0.73	0.73	0.22	0.18	0.06	1.34	57.20	0.12	0.06
PAIMON–5B	2 Hz	0.63	0.60	0.80	0.81	0.20	0.16	0.06	1.37	64.90	0.10	0.04
	1 Hz	0.48	0.35	0.69	0.70	0.23	0.19	0.04	1.23	52.90	0.08	0.03
Action-only	2 Hz	0.67	0.58	0.78	0.78	0.20	0.16	0.07	1.38	61.00	0.12	0.03
	1 Hz	0.48	0.27	0.67	0.68	0.24	0.19	0.05	1.26	52.90	0.10	0.05
PAIMON-Raw	2 Hz	0.61	0.53	0.73	0.71	0.22	0.19	0.01	1.02	53.20	0.05	0.05
	1 Hz	0.54	0.41	0.69	0.70	0.23	0.18	0.02	0.97	47.40	0.06	0.04

C.7 Inference Cost

Table 5 reports the evaluated policy in the first column and six complementary measures of inference cost in the remaining columns. "Params" is the total model parameter count in billions; "Peak mem." is the maximum GPU memory allocated during one planning call, in GiB; "Plan latency p50" is the median wall-clock time per planning call, in milliseconds; "Plan rate" is the corresponding inverse

Table 21: Closed-loop ablation results on the GYM-unrealCV UrbanCity scene (2 Hz).

Model	Freq.	Simulation environment: GYM-unrealCV – UrbanCity											
		TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Navmesh ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓	Distractor ↓
PAIMON(Ours)	2 Hz	0.93	1.00	0.95	0.98	0.14	0.06	0.01	0.16	1.71	53.20	0.04	0.00
PAIMON–5B	2 Hz	0.85	0.87	0.93	0.94	0.14	0.08	0.01	0.05	1.74	59.20	0.06	0.01
Action-only	2 Hz	0.82	0.72	0.87	0.87	0.15	0.06	0.01	0.14	1.82	62.50	0.12	0.01
PAIMON-Raw	2 Hz	0.68	0.89	0.96	0.97	0.13	0.06	0.01	0.05	1.56	48.80	0.02	0.00

Table 22: Closed-loop ablation results on the GYM-unrealCV UrbanRoad scene (2 Hz).

Model	Freq.	Simulation environment: GYM-unrealCV – UrbanRoad											
		TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Navmesh ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓	Distractor ↓
PAIMON(Ours)	2 Hz	0.90	1.00	0.95	0.98	0.15	0.06	0.01	0.14	1.69	52.90	0.05	0.00
PAIMON–5B	2 Hz	0.75	0.84	0.92	0.93	0.14	0.09	0.02	0.05	1.66	57.90	0.04	0.00
Action-only	2 Hz	0.87	0.86	0.92	0.94	0.13	0.07	0.00	0.12	1.68	63.20	0.11	0.00
PAIMON-Raw	2 Hz	0.50	0.88	0.97	0.98	0.14	0.07	0.00	0.02	1.46	49.60	0.01	0.00

Table 23: Closed-loop ablation results on the GYM-unrealCV Garage scene (2 Hz).

Model	Freq.	Simulation environment: GYM-unrealCV – Garage											
		TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Navmesh ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓	Distractor ↓
PAIMON(Ours)	2 Hz	0.84	0.71	0.87	0.90	0.15	0.03	0.03	0.21	1.77	56.10	0.05	0.00
PAIMON–5B	2 Hz	0.69	0.88	0.95	0.96	0.13	0.05	0.04	0.05	1.54	60.40	0.06	0.00
Action-only	2 Hz	0.83	0.79	0.89	0.91	0.14	0.05	0.04	0.10	1.69	61.80	0.12	0.00
PAIMON-Raw	2 Hz	0.15	0.13	0.92	0.89	0.18	0.04	0.06	0.00	0.85	46.40	0.00	0.00

Table 24: Closed-loop ablation results on the GYM-unrealCV SimpleRoom scene (2 Hz).

Model	Freq.	Simulation environment: GYM-unrealCV – SimpleRoom											
		TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Navmesh ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓	Distractor ↓
PAIMON(Ours)	2 Hz	0.91	1.00	1.00	1.00	0.11	0.04	0.00	0.08	1.70	216.20	0.10	0.00
PAIMON–5B	2 Hz	0.87	0.92	0.96	0.97	0.11	0.06	0.00	0.05	1.74	224.40	0.13	0.00
Action-only	2 Hz	0.78	0.63	0.83	0.84	0.16	0.07	0.00	0.14	1.91	220.00	0.28	0.00
PAIMON-Raw	2 Hz	0.26	0.43	1.00	1.00	0.10	0.03	0.00	0.00	1.61	269.40	0.05	0.00

Table 25: Closed-loop ablation results on the HQ-GS Juzizhoutou scene (2 Hz).

Model	Simulation environment: HQ-GS – Juzizhoutou									
	TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓
PAIMON (Ours)	0.90	0.86	0.94	0.94	0.11	0.03	0.04	2.03	64.90	0.07
PAIMON–5B	0.73	0.69	0.84	0.85	0.17	0.03	0.03	2.03	67.20	0.06
Action-only	0.87	0.91	0.96	0.96	0.09	0.02	0.02	1.99	66.80	0.12
PAIMON-Raw	0.52	0.94	0.99	1.00	0.11	0.02	0.00	1.75	49.60	0.02



Figure 14: Qualitative ablation comparison on Habitat-GS. Columns show observations at $t = 1, 3, 5, 7$ s, and rows compare PAIMON with its ablated variants. The displayed TR values are episode-specific.

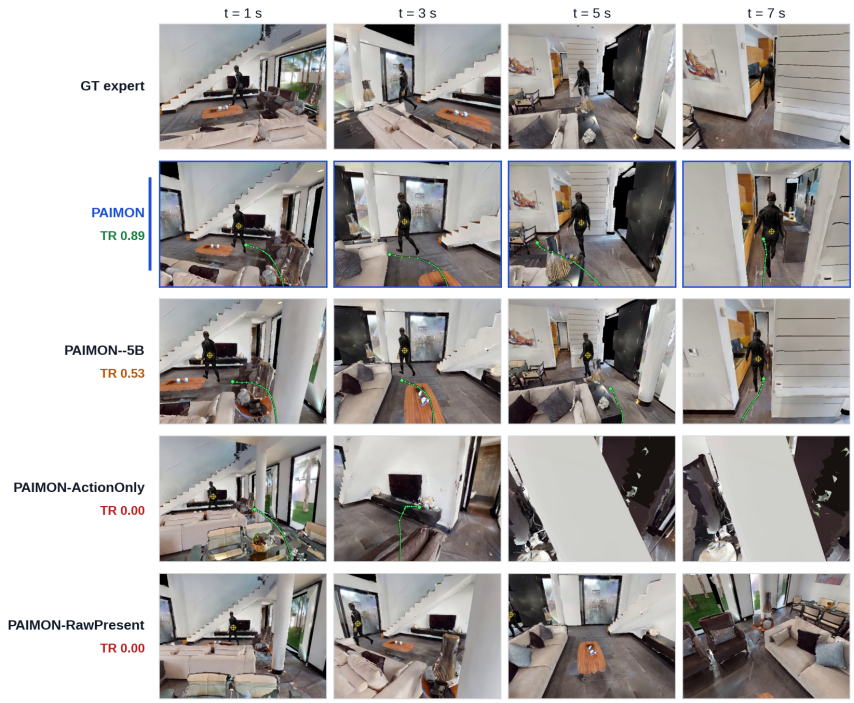

Figure 15: Qualitative ablation comparison on HM3D. Columns show observations at $t = 1, 3, 5, 7$ s, and rows compare PAIMON with its ablated variants. The displayed TR values are episode-specific.

latency, in Hz; "Compute" is the floating-point operation count per planning call, in GFLOP; and "Throughput" is the resulting arithmetic rate, in TFLOP/s, obtained from the per-call operation count and measured execution time. Downward arrows indicate lower resource use, whereas the upward arrow for plan rate indicates a higher attainable replanning frequency. Measurements use an NVIDIA RTX 5090 GPU on a host with a 16-core CPU and 48 GB of system memory, batch size

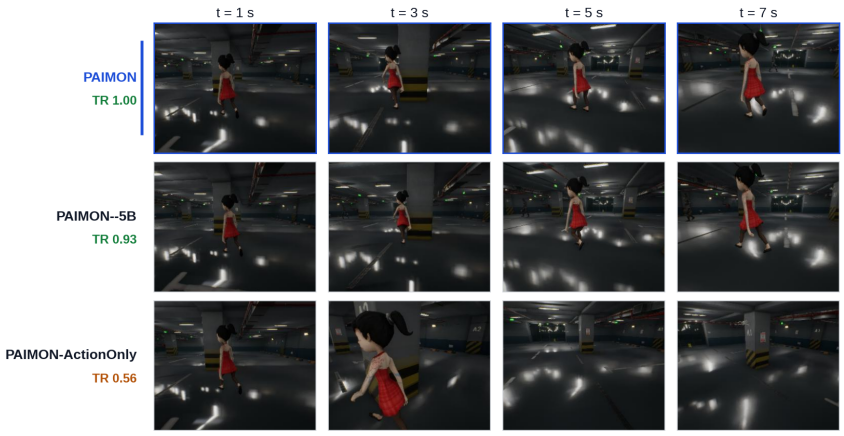

Figure 16: Qualitative ablation comparison on the GYM-unrealCV Garage scene. Columns show observations at $t = 1, 3, 5, 7$ s, and rows compare PAIMON with its ablated variants. The displayed TR values are episode-specific.

Table 26: Closed-loop ablation results on the HQ-GS Qianfoshan scene (2 Hz).

Model	Simulation environment: HQ-GS – Qianfoshan									
	TR ↑	SR ↑	VR ↑	Survive ↑	Center err. ↓	Occluded ↓	Standoff ↓	Speed (m/s) –	Jerk (m/s^3) ↓	Kin. viol. ↓
PAIMON (Ours)	0.79	0.66	0.87	0.79	0.15	0.12	0.05	2.03	71.80	0.10
PAIMON–5B	0.62	0.32	0.69	0.65	0.17	0.10	0.05	2.52	83.20	0.15
Action-only	0.53	0.48	0.70	0.69	0.14	0.08	0.03	2.02	63.70	0.13
PAIMON-Raw	0.41	0.46	0.81	0.73	0.15	0.11	0.00	1.71	53.60	0.07

one, bf16 precision, two denoising steps, and eager inference, matching the hardware and scheduling conditions of the 2 Hz closed-loop evaluation.

C.8 Real-World Deployment

We deploy and test PAIMON on a quadrotor unmanned aerial vehicle (UAV). The RGB camera is rigidly mounted on the airframe, so the camera pose is identical to the UAV pose used by the flight controller. The camera has a horizontal field of view (FOV) of $119°$. To separate model computation from flight operation, we use a distributed architecture in which PAIMON runs on a remote processing server while the UAV executes the returned commands. The server is equipped with an NVIDIA RTX 5090 GPU, a 16-core CPU, and 48 GB of memory.

The UAV and server communicate bidirectionally over a dedicated local-area network (LAN). The uplink continuously sends the RGB image, the current UAV state, and the language instruction to the server. The input RGB frames have a resolution of 960×540, which are horizontally cropped to 720×540 and then resized before being fed into PAIMON. The UAV state is obtained from a motion-capture system. PAIMON processes these multimodal inputs and returns velocity commands over the same connection, consisting of forward velocity, yaw angular velocity, and vertical velocity. The end-to-end communication and inference loop has a round-trip latency of less than 0.5 s, supporting a command execution rate of approximately 2 Hz. The UAV flight-control system applies these commands to execute the tracking policy.

We conduct the real-world experiments in an indoor flight laboratory equipped with a motion-capture system. In each episode, the UAV follows a moving person over a total flight distance of approximately 15 m, while the target moves at an average speed of approximately 1.3 m/s. We evaluate PAIMON over 10 episodes, in which the UAV consistently maintains close tracking of the target throughout the flight.



Figure 17: Qualitative ablation comparison on HQ-GS Juzizhoutou. Columns show observations at $t = 0, 2, 4, 6$ s, and rows compare PAIMON with its ablated variants. The displayed TR values are episode-specific.