OPENNWM: A GENERALIST NAVIGATION WORLD MODEL WITH LATENT ACTION PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Navigation world models predict future observations conditioned on navigation actions to support planning, but their generalization across real-world environments remains constrained by the limited coverage of action-labeled data. Action-free navigation videos provide diverse visual experience, yet lack the recorded actions required for action-conditioned training. We introduce OpenNWM, a generalist navigation world model with latent action pretraining. It learns universal latent actions from large-scale action-free videos through visual prediction pretraining, and further aligns them with physical motions via action-conditioned post-training. To support video pretraining, we curate NavAnywhere, a large-scale action-free navigation video dataset comprising 17.5 million visual observations from 15 heterogeneous sources. It captures navigation by ground robots, humans, and drones, with diverse viewpoints and motion patterns across five broad scene types: residential areas, public and commercial spaces, urban settings, parks, and natural landscapes. Experiments demonstrate improved visual prediction on in-domain and out-of-domain benchmarks, together with improved navigation performance. These results show that latent-action pretraining enables diverse, action-free videos to improve dynamics learning and generalization for open-world navigation.

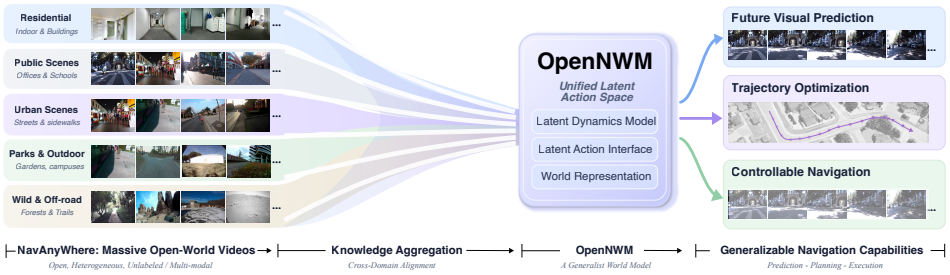

Figure 1: **Overview of OpenNWM.** OpenNWM learns navigation dynamics from heterogeneous action-free videos via latent-action pretraining, then adapts to real waypoint actions for action-conditioned prediction and planning.

1 INTRODUCTION

Visual navigation is a fundamental capability that enables embodied agents to autonomously navigate and accomplish tasks in real world. Recent generalist navigation policies directly map visual observations and goal images to actions, demonstrating promising generalization across diverse environments and embodiments (Shah et al., 2023b;c; Sridhar et al., 2024). Beyond direct policy learning, navigation world models explicitly model environmental dynamics by predicting future visual observations conditioned on candidate actions (Zhu et al., 2026; Yu et al., 2025), enabling agents to imagine and evaluate potential trajectories for flexible planning (Bar et al., 2025; Zhang et al., 2026b). However, existing NWMs rely heavily on action-labeled navigation trajectories, limiting their ability to exploit the breadth of visual experience required for open-world navigation.

Action-labeled trajectories provide a direct connection between navigation actions and their visual consequences, but collecting them across diverse environments and embodiments is costly (Huang

et al., 2025). In contrast, large-scale action-free videos cover diverse egocentric, driving, and scene exploration scenarios without requiring recorded actions (Grauman et al., 2022; Liu et al., 2025; Ling et al., 2024; Akhtyamov et al., 2025). These videos contain rich visual and temporal information, yet do not directly supervise the relationship between executable actions and observed transitions (Wan et al., 2026). NWM partially addresses this gap through time-conditioned prediction on unlabeled videos (Bar et al., 2025), but a temporal offset specifies *when* to predict rather than explicitly describing the motion underlying a transition. This reveals a central tension: action-labeled trajectories provide control supervision but limited coverage, while action-free videos provide broad visual experience without explicit control signals.

Latent action models (LAMs) provide a promising direction by inferring proxy actions from visual transitions, enabling learning from videos without explicit action annotations (Schmidt & Jiang, 2024; Ye et al., 2025; Gao et al., 2025; Garrido et al., 2026). However, reconstruction-based latent actions may capture appearance changes and external dynamics rather than agent-controllable motion (Nikulin et al., 2025). Recent approaches such as villa-X and LatBot incorporate physical supervision to better connect visual transitions with executable motion (Chen et al., 2025; Li et al., 2025). For navigation world models, this raises two connected questions: *what supervision makes latent actions useful for learning navigation dynamics, and how can the resulting predictive dynamics be adapted to real navigation actions?* Addressing both questions is important for turning diverse video experience into effective action-conditioned prediction and planning.

To this end, we introduce OpenNWM (see Figure 1), a generalist navigation world model built upon a universal latent-action space for open-world navigation. OpenNWM learns Universal Navigation Latent Actions (UNLAs) that work as an intermediate training signal for learning predictive dynamics from videos. By optimizing a unified visual-physical objective, OpenNWM learns latent representations that support both future visual prediction and physical motion prediction.

We construct NavAnywhere, an action-free navigation video dataset comprising 17.5 million visual observations from 15 sources. It aggregates public datasets and online videos spanning ground robots, humans, and drones. The dataset covers residential environments, public and commercial spaces, urban and transportation settings, parks and gardens, and natural and off-road environments. We study the effects of latent-action pretraining and LAM supervision on navigation world modeling. As prediction quality alone does not ensure effective decision-making (Zhang et al., 2026a), we evaluate OpenNWM on in-domain and out-of-domain prediction, autoregressive rollouts, and planning under real actions. Our main contributions are as follows:

- We introduce OpenNWM, a generalist navigation world model that learns a universal latent-action space from large-scale heterogeneous visual experiences, enabling scalable navigation learning across diverse environments, embodiments, and motion patterns.
- We curate NavAnywhere, a large-scale heterogeneous navigation dataset comprising 17.5 million visual observations from 15 sources across diverse environments and embodiments. We further propose Universal Navigation Latent Actions (UNLAs), which disentangle agent motion from environmental dynamics and unify heterogeneous visual experiences with navigation signals within a shared latent space.
- OpenNWM achieves state-of-the-art perceptual prediction and downstream navigation performance on the evaluated benchmarks. Systematic ablations further characterize how latent-action pretraining and different LAM supervision strategies contribute to dynamics modeling and cross-domain generalization.

2 Related Work

2.1 From Navigation Policies to Navigation World Models

Learning-based visual navigation has evolved from task-specific controllers to generalist policies that transfer across environments and embodiments. Early visual-goal navigation systems combined learned reachability models with topological memory, enabling long-horizon navigation from offline visual experience (Savinov et al., 2018; Shah et al., 2021b; 2022; Shah & Levine, 2022). This line of work subsequently converged on shared navigation interfaces: GNM provided a unified interface across heterogeneous robot trajectories through embodiment-agnostic waypoint prediction (Shah

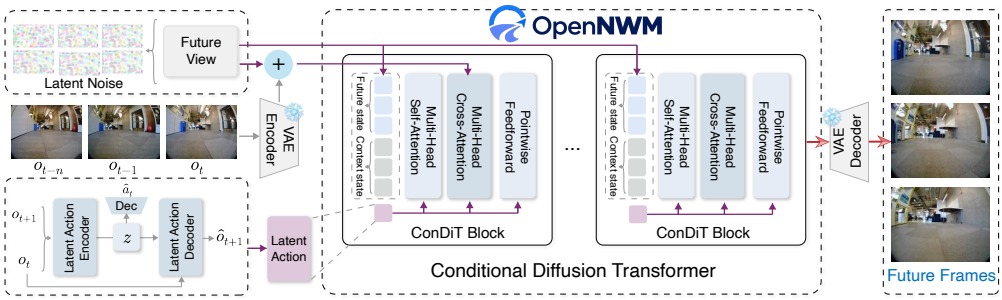

Figure 2: **Framework of OpenNWM.**

et al., 2023b); ViNT extended this interface with large-scale Transformer pretraining (Shah et al., 2023c); and NoMaD employed diffusion-based action generation to unify goal-conditioned navigation and goal-agnostic exploration (Sridhar et al., 2024). Together, these advances highlighted cross-dataset training for transferable navigation policies.

In parallel, pretrained vision–language models extend navigation beyond visual goals to semantic instructions and general tasks, integrating language, vision, and navigation modules for long-horizon reasoning and execution (Shah et al., 2023a; Zhou et al., 2024). Recent video-based navigation models integrate temporal visual context, language understanding, and action prediction, improving generalization across tasks, datasets, and sim-to-real transfer (Zhang et al., 2024; 2025). NaVILA further couples high-level VLA reasoning with embodiment-specific locomotion skills, enabling language-guided navigation for legged robots (Cheng et al., 2025). Meanwhile, automatic extraction of motion supervision from web-scale walking and driving videos expands the coverage of scenes and behaviors available for policy learning (Liu et al., 2025).

Unlike policy-centric methods that implicitly encode dynamics, Navigation World Models explicitly predict action-conditioned futures for planning (Bar et al., 2025). OpenNWM adopts latent actions as a shared control interface, enabling the learning of controllable dynamics jointly from action-labeled trajectories and action-free videos.

2.2 Unified Embodied Learning from Heterogeneous Data

The development of generalist embodied models increasingly relies on jointly learning from diverse experiences across tasks, environments, and robot embodiments. Studies on large-scale robot datasets demonstrate that cross-dataset training facilitates knowledge transfer across robotic platforms, while flexible architectures accommodate heterogeneous observation modalities and action spaces (O'Neill et al., 2024; Octo Model Team et al., 2024). However, such unified learning paradigms still assume that every trajectory is paired with executable robot actions. Their scalability is therefore constrained by the high costs of robot deployment, teleoperated data collection, and alignment across heterogeneous action spaces.

Human egocentric recordings and Internet videos provide large-scale embodied experiences, spanning substantially more diverse scenes and behaviors. Recent studies show that carefully curated human videos can yield more transferable pretrained representations than an equivalent amount of robot data (Ma et al., 2026). Large-scale human-video pretraining also enables world models to acquire broad interaction and environmental priors before adaptation to specific robotic platforms (Gao et al., 2026). However, human and Internet videos lack robot-compatible actions, while embodiment-specific pose estimation and motion retargeting hinder a unified control interface.

Latent actions unify heterogeneous embodied data beyond predefined action spaces, supporting action-free pretraining, task transfer, and visual-action alignment (Schmidt & Jiang, 2024; Ye et al., 2025; Bu et al., 2025; Lin et al., 2026; Chen et al., 2025; Li et al., 2025). OpenNWM extends latent-action learning beyond manipulation and task-specific models to generalist navigation, unifying action-free videos and heterogeneous motion signals through a controllable world-model interface.

3 METHOD

3.1 PROBLEM SETUP

Dataset. Given N heterogeneous visual observations, we construct $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i, c_i)\}_{i=1}^{N}$, where $\mathbf{x}_i = (x_{i,0}, \ldots, x_{i,T_i})$ denotes a visual sequence, $\mathbf{y}_i$ contains optional camera poses, and c_i encodes the data source, embodiment, and sampling rate. For action-free videos, $\mathbf{y}_i = \varnothing$.

Learning Target. Given visual observations up to time τ, we denote the encoded visual states of the most recent h frames as $\mathbf{s}_{\tau-h+1:\tau}$, where h is the visual context length. For a prediction horizon k, the target time is defined as $m = \tau + k$. A latent action is inferred from the visual transition as $z_{\tau\to m} \sim q_\phi(\cdot \mid x_\tau, x_m)$, where q_ϕ discovers a compact latent representation of visual changes without requiring explicit action annotations. Conditioned on the inferred latent action, OpenNWM learns future visual dynamics by optimizing:

$$\mathcal{J}_{\text{pred}}(\theta, \phi) = \mathbb{E}_{z_{\tau:m}\sim q_\phi(\cdot|x_\tau, x_m)} \left[\log p_\theta(s_m \mid \mathbf{s}_{\tau-h+1:\tau}, z_{\tau:m}, k)\right] \tag{1}$$

Here, p_θ denotes a diffusion-parameterized conditional distribution over future visual states. The latent action serves as a shared interface for visual transitions, enabling OpenNWM to model multi-modal future dynamics without explicit action labels.

Discussion. Conventional Navigation World Models rely on action-labeled trajectories, while OpenNWM discovers and grounds latent actions from heterogeneous visual experiences, enabling unified learning from action-free videos and robot data.

3.2 UNIVERSAL NAVIGATION LATENT ACTIONS

Visual transitions may contain a mixture of agent motion, environmental dynamics, and observation variations. Latent actions learned solely from visual prediction may capture arbitrary changes rather than meaningful agent behaviors. We address this ambiguity by learning a continuous latent action representation that captures visual transitions while preserving controllable motion semantics.

Variational Transition Modeling. Given an initial observation x_τ and a future observation x_m, we infer a continuous latent action representation $z_{\tau\to m}$ that captures the corresponding visual transition. The resulting variational objective is formulated as:

$$\mathcal{L}_{\text{vis}} = -\mathbb{E}_{q_\phi} \left[\log p_\psi(x_m \mid x_\tau, z_{\tau\to m})\right] + \beta D_{\text{KL}} \left(q_\phi(z_{\tau\to m} \mid x_\tau, x_m) \| p(z)\right) \tag{2}$$

Here, q_ϕ denotes a diagonal Gaussian posterior, $p(z) = \mathcal{N}(0, I)$ is the latent prior, and p_ψ is an auxiliary transition predictor. The reconstruction term preserves transition information, while KL regularization constrains the latent space. The predictor only regularizes latent action learning, with high-fidelity generation handled by the Navigation World Model.

Physical Motion Grounding. Visual transitions may capture environmental and observation changes beyond controllable agent motion. For trajectories with physical annotations, we align latent actions with motion transitions and robot actions through the following objective:

$$\mathcal{L}_{\text{phy}} = -\mathbb{E}_{q_\phi} \left[\log p_\omega \left(\mathbf{y}_{\tau:m} \mid z_{\tau\to m}\right)\right] \tag{3}$$

Here, $\mathbf{y}\tau : m$ contains available future motion states. The predictor p_ω maps latent actions to physical transitions. This objective is applied only to trajectories with available physical supervision.

Heterogeneous Supervision. We learn the latent-action representation from all visual observations, while applying physical-motion supervision only to trajectories with available annotations. The resulting joint objective is formulated as:

$$\mathcal{L}_{\text{latent}} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} \left[\mathcal{L}_{\text{vis}}\right] + \lambda_{\text{phy}} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}_{\text{phy}}} \left[\mathcal{L}_{\text{phy}}\right] \tag{4}$$

Here, $\mathcal{D}_{\text{phy}} \subseteq \mathcal{D}$ denotes the subset with available physical motion annotations, and λ_{phy} controls the contribution of physical supervision. The objective is applied to all observation sequences, while the physical objective is activated for motion-annotated sequences to ground latent actions in navigation dynamics. The learned latent actions serve as unified conditioning variables for OpenNWM.

3.3 LATENT-CONDITIONED NAVIGATION WORLD MODELING

Given the learned universal latent-action space, the posterior distribution q_ϕ infers latent actions from both action-free videos and motion-annotated trajectories. Each observation x_i is encoded into a visual latent state s_i, and the historical context at time τ is denoted by $\mathbf{s}_{0:\tau}$. We model the conditional distribution of future visual states given historical context and latent actions (see Figure 2).

Latent Conditioning. For a transition from τ to m, we combine the embeddings of the latent action, temporal horizon, and diffusion timestep to construct the conditioning representation:

$$\xi^{(t)}_{\tau:m} = \psi_z(z_{\tau\to m}) + \psi_k(k) + \psi_t(t) \tag{5}$$

Here, ψ_z, ψ_k, and ψ_t denote the embedding functions for the latent action, temporal horizon, and diffusion timestep, respectively. The conditioning representation $\xi^{(t)}_{\tau:m}$ is injected through adaptive layer normalization, while historical visual states $\mathbf{s}_{0:\tau}$ are incorporated via cross-attention.

Conditional Diffusion Dynamics. At diffusion timestep t, the forward process gradually perturbs the target future state s_m with Gaussian noise according to a predefined noise schedule:

$$s^{(t)}_m = \sqrt{\alpha_t} s_m + \sqrt{1-\alpha_t}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I) \tag{6}$$

Here, $s^{(t)}_m$ denotes the noisy future state at diffusion timestep t, and α_t is determined by the noise schedule. Conditioned on the historical context $\mathbf{s}_{0:\tau}$ and the latent-action conditioning representation $\xi^{(t)}_{\tau:m}$, the denoising model predicts the future state:

$$\widehat{s}_m = F_\theta(s^{(t)}_m \mid \mathbf{s}_{\tau-h+1:\tau}, \xi^{(t)}_{\tau:m})$$

Iterating the reverse diffusion process produces a multimodal distribution of future visual states conditioned on the inferred latent action.

Unified World-Model Learning. OpenNWM is optimized using a hybrid objective combining future-state prediction and a variational diffusion objective:

$$\mathcal{L}_{\text{NWM}} = \mathbb{E}\left[\|s_m - \widehat{s}_m\|_2^2\right] + \lambda_{\text{vlb}}\mathcal{L}_{\text{vlb}} \tag{7}$$

Here, the first term supervises future-state prediction, while $\mathcal{L}_{\text{vlb}}$ denotes the variational lower bound for learning the reverse diffusion model, and λ_{vlb} controls its contribution. Since latent actions are available for all visual transitions, the same diffusion objective applies to both action-free and motion-annotated data. Physical annotations only ground the latent-action space, while OpenNWM generates multimodal future states for navigation planning.

3.4 NAVIGATION PLANNING

Given the visual context $\mathbf{s}_{\tau-h+1:\tau}$ and goal observation s^g, the post-trained OpenNWM evaluates candidate real waypoint-action sequences through action-conditioned future visual rollouts. We retain energy-based receding-horizon planning while optimizing directly over real waypoint actions. Latent actions are used for world-model pretraining rather than as online planning variables.

Action-Conditioned Trajectory Optimization. Let $\mathbf{a} = (a_0, \ldots, a_{H-1})$ denote a candidate real waypoint-action sequence over a planning horizon H. Conditioned on the visual context and each candidate action sequence, OpenNWM rolls out the corresponding future visual states $\widehat{\mathbf{s}} = (\widehat{s}_1, \ldots, \widehat{s}_H)$. We evaluate each candidate rollout using the following planning energy:

$$\mathcal{E}(\widehat{\mathbf{s}}, \mathbf{a}; s^g, c) = -\mathcal{S}_{\text{goal}}(\widehat{s}_H, s^g) + \mathcal{C}_{\text{nav}}(\widehat{\mathbf{s}}, \mathbf{a}, c). \tag{8}$$

Here, $\mathcal{S}_{\text{goal}}$ measures the similarity between the predicted terminal state and the goal observation, while $\mathcal{C}_{\text{nav}}$ aggregates navigation constraints, including predicted-state safety and physical-action feasibility under the embodiment context c. The optimal action sequence is selected by minimizing the expected trajectory energy:

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} \mathbb{E}_{\widehat{\mathbf{s}}\sim p_\theta(\cdot|\mathbf{s}_{0:\tau},\mathbf{a})}\left[\mathcal{E}(\widehat{\mathbf{s}}, \mathbf{a}; s^g, c)\right]. \tag{9}$$

Here, $p_\theta(\cdot \mid \mathbf{s}_{0:\tau}, \mathbf{a})$ denotes the rollout distribution induced by the post-trained, real-action-conditioned world model, with the fixed temporal offset per planning step omitted for brevity. We

Algorithm 1 Action-Conditioned Receding-Horizon Planning

Require: Post-trained world model p_θ, visual encoder E, goal observation x^g, context c
Require: Horizon H, context length h, candidates N, elites K, rollouts M, CEM iterations J

$s^g \leftarrow E(x^g)$
while the goal is not reached **do**
  Observe x_τ and update the encoded visual context $\mathbf{s}_{\tau-h+1:\tau}$ using E
  Initialize $\mathcal{Q}_0 = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$
  for $j = 0, \dots, J-1$ **do**
    for $n = 1, \dots, N$ **do**
      Sample $\mathbf{a}^{(n)} \sim \mathcal{Q}_j$, $\quad \overline{\mathcal{E}}_n \leftarrow 0$
      for $r = 1, \dots, M$ **do**
        $\widehat{\mathbf{s}}^{(n,r)} \sim p_\theta\big(\cdot \mid \mathbf{s}_{\tau-h+1:\tau}, \mathbf{a}^{(n)}\big)$
        $\overline{\mathcal{E}}_n \leftarrow \overline{\mathcal{E}}_n + \frac{1}{M}\mathcal{E}\big(\widehat{\mathbf{s}}^{(n,r)}, \mathbf{a}^{(n)}; s^g, c\big)$
      end for
    end for
    $\mathcal{I}_{\text{elite}} \leftarrow \arg\operatorname{sort}_K\{\overline{\mathcal{E}}_n\}_{n=1}^N$
    Refit $\mathcal{Q}_{j+1}$ using $\{\mathbf{a}^{(n)} : n \in \mathcal{I}_{\text{elite}}\}$
  end for
  $n^\star \leftarrow \arg\min_{n\in\{1,\dots,N\}} \overline{\mathcal{E}}_n$
  $\mathbf{a}^\star \leftarrow \mathbf{a}^{(n^\star)}$
  $a_\tau^\star \leftarrow a_0^\star$
  Execute $a_\tau^\star$ through the robot controller and advance to the next observation
end while

solve this optimization using the cross-entropy method (CEM) (Rubinstein, 1997), iteratively updating a proposal distribution toward elite action sequences evaluated through world-model rollouts.

Action Execution. The first optimized action a_0^* is executed by the robot. The auxiliary physical predictor p_ω is used only for motion grounding during latent-action learning and is not required for online planning. Upon receiving a new observation, OpenNWM updates its visual context and repeats action-space optimization. This receding-horizon procedure enables predictive planning directly in the real waypoint-action space.

Action-Conditioned Planning. OpenNWM solves goal-conditioned navigation via receding-horizon optimization over real waypoint-action sequences. Candidates are sampled from a Gaussian proposal distribution and evaluated through real-action-conditioned world-model rollouts. Their energies jointly account for goal similarity, predicted-state safety, and physical-action feasibility. The lowest-energy candidates are used to update the proposal distribution over J CEM iterations. Only the first action of the selected sequence is passed to the robot controller before replanning from updated observations. Pseudocode is provided in Algorithm 1. We use $H = 8$ planning steps, corresponding to 2 seconds with 0.25-second intervals, and evaluate $N = 80$ candidate sequences with $M = 3$ stochastic rollouts per candidate in each CEM iteration.

3.5 NavAnyWhere

We construct NavAnywhere, a large-scale heterogeneous visual navigation corpus, by integrating 15 open-source datasets. It contains 17.5M frames of observations, spanning action-free videos across diverse environments. NavAnywhere enables OpenNWM to jointly learn scalable visual dynamics and physically grounded latent actions. NavAnywhere details are in Appendix A; preprocessing and dataset usage are described in Appendix B.

4 Experiments

4.1 Experiment Setups

Datasets. We pretrain OpenNWM on NavAnywhere, a large-scale action-free video corpus covering diverse indoor, outdoor, urban, and natural environments. For downstream training, we

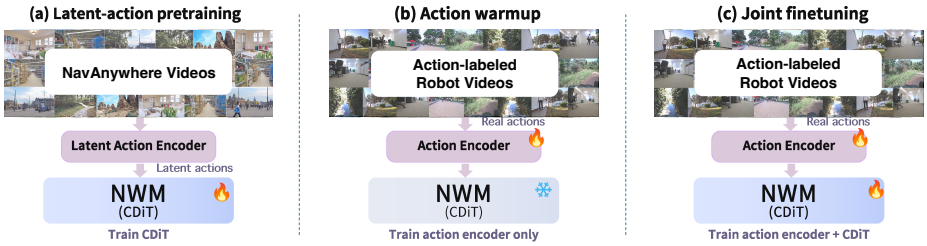

Figure 3: Three-stage training pipeline of OpenNWM. (a) Latent-action pretraining on action-free NavAnywhere videos. (b) Action-encoder warmup with the CDiT backbone frozen. (c) Joint post-training of the entire model.

Table 1: Direct visual prediction at 4 s. DS: DreamSim; NWM†: CDiT-XL + Ego4D. ID/OOD denote in-domain/out-of-domain evaluation. Best and second-best results are **bold** and underlined, respectively.

Method	Params. (M)	ID			OOD (PSNR↑)			
		LPIPS↓	DS↓	PSNR↑	Stanford	TUM	Rover	Go2
NWM	279	0.405	0.230	13.811	11.323	8.957	11.739	10.750
NWM†	1,099	0.419	0.227	13.523	11.444	8.884	11.906	**11.083**
RAE-NWM	867	0.442	0.254	13.957	11.528	9.007	9.449	10.500
OpenNWM	280	**0.368**	**0.196**	**14.299**	**11.591**	**9.135**	**12.574**	11.047

Table 2: Navigation performance on RECON, SCAND, and Office Go2 (OOD). Lower ATE and RPE indicate better performance. Best and second-best results are **bold** and underlined, respectively.

Method	Params. (M)	RECON		SCAND		Office Go2	
		ATE↓	RPE↓	ATE↓	RPE↓	ATE↓	RPE↓
GNM	9	1.87	0.73	2.12	0.61	–	–
NoMaD	19	1.95	0.53	2.24	0.49	–	–
NWM	279	1.13	**0.35**	1.01	0.28	3.03	0.68
RAE-NWM	867	1.36	0.37	1.14	0.28	3.01	0.67
Garrido et al. (2026)	–	1.40	0.40	–	–	–	–
CompACT	–	1.33	0.39	1.36	0.34	–	–
OpenNWM	280	**1.12**	**0.35**	**0.98**	**0.27**	**2.89**	**0.65**

use SCAND (Karnan et al., 2022), TartanDrive (Triest et al., 2022), RECON Shah et al. (2021a), and HuRoN (SACSoN) (Hirose et al., 2023), whose pose annotations are converted into relative translations and rotations. Translational displacements are normalized by the average step length of each dataset, and backward-motion transitions are removed. We evaluate in-domain performance on these four datasets and assess generalization to unseen environments on Go Stanford, the TUM RGB-D SLAM Dataset, Office-Go2, and Planetary Rover. Office-Go2 contains real-world office navigation sequences that we collected using a quadruped robot, while Planetary Rover comprises lunar and Martian rover navigation data sourced from the Internet. Details are provided in the Appendix B.2.

Evaluation Metrics. ATE and RPE assess trajectory accuracy and local pose consistency (Sturm et al., 2012). PSNR, LPIPS (Zhang et al., 2018), and DreamSim (Fu et al., 2023) measure visual fidelity, while FID (Heusel et al., 2017) assesses image distribution quality. We also evaluate LAM reconstruction under ID and OOD settings. Details are provided in the Appendix C.2

Baselines. We compare OpenNWM with NWM (Bar et al., 2025) and RAE-NWM (Zhang et al., 2026b) for visual prediction and planning, and include NWM CDiT-XL + Ego4D (Grauman et al., 2022) for visual prediction. Navigation baselines additionally include GNM (Shah et al., 2023b), NoMaD (Sridhar et al., 2024), CompACT (Kim et al., 2026), and the latent-action controller of Garrido et al. (2026).

Implementation Details. As shown in Figure 3. we train the LAM on NavAnywhere together with motion-annotated trajectories from RECON, HuRoN, and SCAND, and use its inferred latent actions to pretrain the world model. We then post-train on four navigation datasets with camera poses for physical grounding. OpenNWM uses a CDiT-B/2 backbone and a Stable Diffusion VAE (Blattmann et al., 2023), with four context frames and four sampled prediction horizons per trajectory. Both LAM and NWM are trained on 8 NVIDIA L20 GPUs.

4.2 Comparison Results

Visual Prediction. Table 1 reports ID averages over RECON, SCAND, HuRoN, and TartanDrive, and OOD PSNR on Go Stanford (Stanford), TUM RGB-D (TUM), Planetary Rover (Rover), and Office Go2 (Go2). OpenNWM achieves the best results among the compared methods on all three

Table 3: Image reconstruction performance of LAMs. ID (in-domain): average over RECON, SCAND, and HuRoN. OOD (out-of-domain): unseen Go Stanford. Best results are in **bold** and second best are underlined.

Model	Source	ID		OOD	
		PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓
villa-X	Chen et al. 2026	15.327	0.516	17.560	0.550
LARA	Liu et al. 2026	15.752	0.538	17.742	0.527
Olaf-World	Jiang et al. 2026b	14.764	0.635	16.599	0.614
DiLA	Zhang et al. 2026c	13.180	0.597	14.078	0.621
DreamDojo	Gao et al. 2026	18.950	0.529	20.713	0.485
Ours		**21.071**	**0.505**	**21.912**	**0.472**

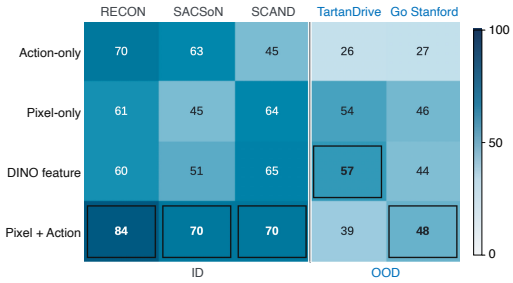

Figure 4: Semantic consistency of latent actions. Detailed metric definitions are provided in the appendix.

metrics. Relative to the strongest baseline for each metric, it reduces LPIPS and DreamSim by 9.1% and 13.7%, respectively, and improves PSNR by 0.342 dB. These gains are obtained with a 280M-parameter world model, comparable in size to NWM (279M) and substantially smaller than RAE-NWM (867M) and NWM CDiT-XL + Ego4D (1,099M). These results suggest that OpenNWM learns a more accurate model of navigation dynamics, enabling more faithful action-conditioned predictions of future observations.

Navigation Performance. Table 2 evaluates navigation performance on RECON, SCAND, and Office Go2 using ATE and RPE. OpenNWM achieves the lowest ATE and the best or tied-best RPE across all three datasets. On the in-domain benchmarks, it reduces ATE from 1.13 to 1.12 on RECON and from 1.01 to 0.98 on SCAND relative to NWM. It also ties with NWM for the best RPE of 0.35 on RECON and achieves the lowest RPE of 0.27 on SCAND, compared with 0.28 for both NWM and RAE-NWM. Together with the visual prediction results, these findings support the effectiveness of OpenNWM as a predictive model for navigation planning.

Out-of-Distribution Generalization. We evaluate generalization to unseen domains through both visual prediction and navigation planning. For visual prediction, OpenNWM achieves the highest PSNR on three of the four OOD datasets in Table 1, outperforming the strongest baseline on Go Stanford, TUM RGB-D, and Planetary Rover. On Office Go2, it ranks second with a PSNR of 11.047 dB, only 0.036 dB below NWM CDiT-XL + Ego4D. Beyond visual prediction, Office Go2 also serves as an OOD navigation benchmark in Table 2. OpenNWM achieves the lowest ATE of 2.89 and RPE of 0.65, improving upon the strongest navigation baseline RAE-NWM on this dataset. Notably, OpenNWM achieves the best reported navigation accuracy on Office Go2 despite ranking second in PSNR, highlighting the importance of assessing generalization through downstream planning as well as visual prediction. Overall, these results support the transferability of OpenNWM to the evaluated unseen navigation domains.

LAM Reconstruction and Generalization. As shown in Table 3, our LAM achieves the best RGB reconstruction results in both in-domain (ID) and out-of-domain (OOD) settings, demonstrating consistent advantages in pixel fidelity and perceptual similarity. For ID evaluation, our method achieves a PSNR of 21.071 dB and an LPIPS of 0.505, improving upon the strongest baseline for each metric by 2.121 dB and 2.1%, respectively. For OOD evaluation, it achieves a PSNR of 21.912 dB and an LPIPS of 0.472, outperforming DreamDojo by 1.199 dB and 2.7%, respectively. These results suggest that the learned latents retain key visual information and generalize to unseen domains, supporting downstream world-model prediction.

4.3 Ablation Studies

To assess the effectiveness of our components, we focus on two questions: (1) Do latent actions benefit navigation world models? (2) Which LAM supervision best supports navigation world models? We evaluate visual prediction using LPIPS for direct prediction on Office Go2 at 4 s and autoregressive prediction on Go Stanford at 4 FPS up to 4 s. Due to computational constraints, we use a smaller subset of NavAnywhere for pretraining in these ablations. Details are provided in the Appendix C.1.

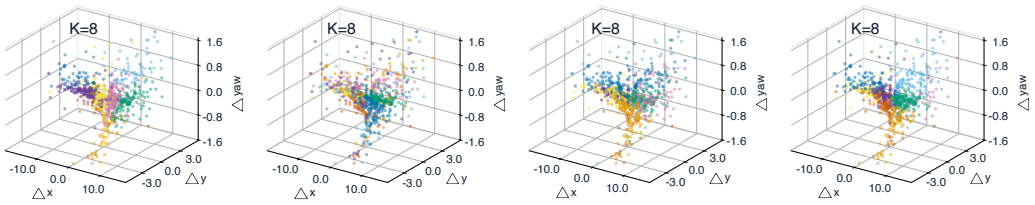

Figure 5: Latent-action clustering under four LAM supervision paradigms.

Table 4: Pretraining ablation. Effect of different DiT conditioning signals. Best and second-best results are bold and underlined, respectively.

DiT Conditioning		Reconstruction		Navigation			
Time	Latent Action	Direct	Autoregressive	RECON		SCAND	
		LPIPS ↓	LPIPS ↓	ATE ↓	RPE ↓	ATE ↓	RPE ↓
✗	✗	0.628	0.667	1.198	0.367	1.252	0.316
✓	✗	0.595	0.678	1.174	0.365	1.045	0.286
✗	✓	0.609	0.667	1.198	0.359	1.067	0.289
✓	✓	**0.587**	**0.658**	**1.117**	**0.356**	**0.950**	**0.267**

Table 5: LAM supervision ablation. Effect of different supervision on latent-action learning. Best/second-best results are **bold**/underlined.

LAM Supervision		Reconstruction		Navigation			
Pixel	Action	Direct	Autoregressive	RECON		SCAND	
		LPIPS ↓	LPIPS ↓	ATE ↓	RPE ↓	ATE ↓	RPE ↓
✗	✓	0.598	0.670	1.138	0.365	0.981	0.274
✓	✗	0.605	0.674	1.176	**0.352**	0.982	0.274
✓	✓	**0.587**	**0.658**	**1.117**	0.356	**0.950**	**0.267**

Pretraining and Conditioning Ablation. Table 4 compares different conditioning signals during pretraining. Joint time and latent-action conditioning performs best across all six metrics, reducing SCAND ATE from 1.045 with time conditioning alone to 0.950. This suggests that motion information complements the temporal prediction horizon.

LAM Supervision Ablation. Table 5 compares world model performance using different LAMs learned with pixel-only, action-only, and joint supervision. As shown in Figure 4, Pixel+Action supervision achieves the highest Macro-6 Recall on four of five datasets. It consistently outperforms action-only supervision, suggesting that visual reconstruction complements physical motion grounding in aligning latent actions with navigation semantics. Joint supervision leads on five of six metrics. Figure 5 visualizes latent-cluster assignments in the physical motion space $(\Delta x, \Delta y, \Delta\mathrm{yaw})$ under four LAM supervision paradigms. Compared with pixel-only supervision, joint supervision exhibits more coherent cluster organization with respect to translation and rotation, suggesting stronger alignment between latent actions and physical motion semantics. Together, these quantitative results and qualitative patterns suggest that visual prediction and physical-motion grounding provide complementary supervision for learning effective latent actions. The Appendix F provides LAM paradigm definitions, qualitative analysis, and semantic-consistency evaluation.

5 Conclusion

We introduced OpenNWM, a generalist world model for open-world visual navigation. At its core is a shared latent-action interface that unifies learning from heterogeneous visual trajectories with different action spaces and levels of physical supervision. By coupling large-scale action-free video pretraining with physical-motion grounding, OpenNWM learns transition representations that are both predictive of future observations and aligned with executable navigation motion. These representations condition a diffusion world model and support receding-horizon planning after post-training. Experiments across diverse environments demonstrate strong performance in visual future prediction, navigation planning, and generalization to unseen scenes. Taken together, our results establish latent actions as a scalable means of converting open-domain videos into control-relevant navigation supervision, providing a promising foundation for generalist Navigation World Models. Future work will extend this framework to longer-horizon prediction, higher-dimensional action spaces, and online adaptation in dynamic environments.

AI USE STATEMENT

In this work, we have not used generative AI tools for any tasks with required disclosure, and all such tasks are not applicable to this work. We used generative AI tools only to polish the writing of parts of the manuscript. The authors reviewed all AI-assisted edits to ensure that they preserve the intended technical meaning and contain no unsupported claims. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

For reproducibility, the main paper and appendices detail our methodology and implementation. We will release the complete source code and model weights.

REFERENCES

Timur Akhtyamov, Mohamad Al Mdfaa, Javier Antonio Ramirez, Sergey Bakulin, German Devchich, Denis Fatykhov, Alexander Mazurov, Kristina Zipa, Malik Mohrat, Pavel Kolesnik, Ivan Sosin, and Gonzalo Ferrer. EgoWalk: A multimodal dataset for robot navigation in the wild. *arXiv preprint arXiv:2505.21282*, 2025. URL https://arxiv.org/abs/2505.21282.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15791–15801, 2025. URL https://arxiv.org/abs/2412.03572.

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

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. In *Proceedings of Robotics: Science and Systems (RSS)*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.014.

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, Jianyu Chen, and Jiang Bian. villa-X: Enhancing latent action modeling in vision-language-action models. *arXiv preprint arXiv:2507.23682*, 2025. URL https://arxiv.org/abs/2507.23682.

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, et al. Villa-x: enhancing latent action modeling in vision-language-action models. In *International Conference on Learning Representations*, volume 2026, pp. 70673–70703, 2026.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged robot vision-language-action model for navigation. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.018.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. DreamSim: Learning new dimensions of human visual similarity using synthetic data. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. AdaWorld: Learning adaptable world models with latent actions. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 18744–18771. PMLR, 2025. URL https://arxiv.org/abs/2503.18938.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild. In *Forty-third International Conference on Machine Learning*, 2026.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18995–19012, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6626–6637, 2017.

Noriaki Hirose, Dhruv Shah, Ajay Sridhar, and Sergey Levine. Sacson: Scalable autonomous control for social navigation. *IEEE Robotics and Automation Letters*, 9(1):49–56, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2world: Crafting video diffusion models to interactive world models. *arXiv preprint arXiv:2505.14357*, 2025.

Peng Jiang, Kasi Viswanath, Akhil Nagariya, George Chustz, Maggie Wigness, Philip Osteen, Timothy Overbye, Christian Ellis, Long Quang, Jia Huang, et al. Go: The great outdoors multimodal dataset. In *2026 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1496–1502. IEEE, 2026a.

Yuxin Jiang, Yuchao Gu, Ivor W Tsang, and Mike Zheng Shou. Olaf-world: Orienting latent actions for video world modeling. *arXiv preprint arXiv:2602.10104*, 2026b.

Faith Johnson, Kristin Dana, Bryan Bo Cao, Shubham Jain, and Ashwin Ashok. A landmark-aware visual navigation dataset for map representation learning. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 1026–1031. IEEE, 2025.

Haresh Karnan, Anirudh Nair, Xuesu Xiao, Garrett Warnell, Sören Pirk, Alexander Toshev, Justin Hart, Joydeep Biswas, and Peter Stone. Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation. *IEEE Robotics and Automation Letters*, 7 (4):11807–11814, 2022.

Dongwon Kim, Gawon Seo, Jinsung Lee, Minsu Cho, and Suha Kwak. Planning in 8 tokens: A compact discrete tokenizer for latent world model. *arXiv preprint arXiv:2603.05438*, 2026.

Zuolei Li, Xingyu Gao, Xiaofan Wang, and Jianlong Fu. LatBot: Distilling universal latent actions for vision-language-action models. *arXiv preprint arXiv:2511.23034*, 2025. URL https://arxiv.org/abs/2511.23034.

Yihan Lin, Haoyang Li, Yang Li, Haitao Shen, Yihan Zhao, Chao Shao, and Jing Zhang. From pixels to tokens: A systematic study of latent action supervision for vision-language-action models. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://arxiv.org/abs/2605.04678.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169, 2024. URL https://arxiv.org/abs/2312.16256.

Mengya Liu, Baoxiong Jia, Jiangyong Huang, Jingze Zhang, and Siyuan Huang. Lara: Latent action representation alignment for vision-language-action models. *arXiv preprint arXiv:2606.07100*, 2026.

Xinhao Liu, Jintong Li, Yicheng Jiang, Niranjan Sujay, Zhicheng Yang, Juexiao Zhang, John Abanes, Jing Zhang, and Chen Feng. CityWalker: Learning embodied urban navigation from web-scale videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6875–6885, 2025. URL https://arxiv.org/abs/2411.17820.

Yuanzhi Liu, Yujia Fu, Minghui Qin, Yufeng Xu, Baoxin Xu, Fengdong Chen, Bart Goossens, Poly ZH Sun, Hongwei Yu, Chun Liu, et al. Botanicgarden: A high-quality dataset for robot navigation in unstructured natural environments. *IEEE Robotics and Automation Letters*, 9(3): 2798–2805, 2024.

Juncheng Ma, Jianxin Bi, Yufan Deng, Xuanran Zhai, Kewei Zhang, Ye Huang, Bo Liang, Shukai Gong, Jiankai Tu, Xiaotian Tang, Jiaxin Li, Kaiqi Chen, Duomin Wang, Yuqi Wang, Bingyi Kang, Eric Huang, Zhiyang Dou, Zhen Dong, Enze Xie, Wojciech Matusik, Tat-Seng Chua, and Daquan Zhou. HumanScale: Egocentric human video can outperform real-robot data for embodied pretraining. *arXiv preprint arXiv:2606.20521*, 2026. URL https://arxiv.org/abs/2606.20521.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. *arXiv preprint arXiv:2502.00379*, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024. doi: 10.15607/RSS.2024.XX.090.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Alex Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open X-embodiment: Robotic learning datasets and RT-X models. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

Reuven Y Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997.

Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *International Conference on Learning Representations*, 2018.

Florian Scalvini, Camille Bordeau, Maxime Ambard, Cyrille Migniot, Mathilde Vergnaud, and Julien Dubois. ub-visiogeoloc: An image sequences dataset of pedestrian navigation including geolocalised-inertial information and spatial sound rendering of the urban environment's obstacles. *Data in brief*, 53:110088, 2024.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2312.10812.

Fabian Schmidt, Julian Daubermann, Marcel Mitschke, Constantin Blessing, Stephan Meyer, Markus Enzweiler, and Abhinav Valada. Rover: A multiseason dataset for visual slam. *IEEE Transactions on Robotics*, 41:4005–4022, 2025.

Dhruv Shah and Sergey Levine. ViKiNG: Vision-based kilometer-scale navigation with geographic hints. In *Proceedings of Robotics: Science and Systems*, 2022. doi: 10.15607/RSS.2022.XVIII.019.

Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Rapid exploration for open-world navigation with latent goal models. *arXiv preprint arXiv:2104.05859*, 2021a.

Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. ViNG: Learning open-world navigation with visual goals. In *IEEE International Conference on Robotics and Automation*, pp. 13215–13222, 2021b. doi: 10.1109/ICRA48506.2021.9561936.

Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. RECON: Rapid exploration for open-world navigation with latent goal models. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 674–684. PMLR, 2022.

Dhruv Shah, Błażej Osiński, Brian Ichter, and Sergey Levine. LM-Nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 492–504. PMLR, 2023a.

Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7226–7233. IEEE, 2023b.

Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. ViNT: A foundation model for visual navigation. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 711–733. PMLR, 2023c. URL https://proceedings.mlr.press/v229/shah23a.html.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Krishna Kumar Singh, Kayvon Fatahalian, and Alexei A Efros. Krishnacam: Using a longitudinal, single-person, egocentric dataset for scene understanding tasks. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–9. IEEE, 2016.

Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 63–70. IEEE, 2024.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580, 2012.

Hailiang Tang, Tisheng Zhang, Liqiang Wang, Xin Ding, Man Yuan, Zhiyu Xiang, Jujin Chen, Yuhan Bian, Shuangyan Liu, Yuqing Wang, et al. I2nav-robot: A large-scale indoor-outdoor robot dataset for multi-sensor fusion navigation and mapping. *arXiv preprint arXiv:2508.11485*, 2025.

Samuel Triest, Matthew Sivaprakasam, Sean J Wang, Wenshan Wang, Aaron M Johnson, and Sebastian Scherer. Tartandrive: A large-scale dataset for learning off-road dynamics models. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2546–2552. IEEE, 2022.

Shashank Venkataramanan, Mamshad Nayeem Rizve, João Carreira, Yuki Asano, and Yannis Avrithis. Is imagenet worth 1 video? learning strong image encoders from 1 long unlabelled video. In *International Conference on Learning Representations*, volume 2024, pp. 31952–31973, 2024.

Sagar M Waghmare, Kimberly Wilber, Dave Hawkey, Xuan Yang, Matthew Wilson, Stephanie Debats, Cattalyya Nuengsigkapian, Astuti Sharma, Lars Pandikow, Huisheng Wang, et al. Sanpo: A scene understanding accessibility and human navigation dataset. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 7855–7864, 2025.

Shenghua Wan, Le Gan, and De-Chuan Zhan. Learning to be uncertain: Pre-training world models with horizon-calibrated uncertainty. In *International Conference on Learning Representations*, volume 2026, pp. 106107–106127, 2026.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.11758.

Wei Yu, Songheng Yin, Steve Easterbrook, and Animesh Garg. Egosim: Egocentric exploration in virtual worlds with multi-modal conditioning. In *International Conference on Learning Representations*, volume 2025, pp. 43061–43075, 2025.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal Patel, Paul Liang, et al. World-in-world: World models in a closed-loop world. In *International Conference on Learning Representations*, volume 2026, pp. 55660–55699, 2026a.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. NaVid: Video-based VLM plans the next step for vision-and-language navigation. In *Proceedings of Robotics: Science and Systems*, 2024.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-NaVid: A video-based vision-language-action model for unifying embodied navigation tasks. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.013.

Mingkun Zhang, Wangtian Shen, Fan Zhang, Haijian Qin, Zihao Pei, and Ziyang Meng. Rae-nwm: Navigation world model in dense visual representation space. *arXiv preprint arXiv:2603.09241*, 2026b.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Tianqiu Zhang, Muyang Lyu, Yufan Zhang, Fang Fang, and Si Wu. Dila: Disentangled latent action world models. *arXiv preprint arXiv:2605.15725*, 2026c.

Gengze Zhou, Yicong Hong, and Qi Wu. NavGPT: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7641–7649, 2024. doi: 10.1609/aaai.v38i7.28597.

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

Yixuan Zhu, Jiaqi Feng, Wenzhao Zheng, Yuan Gao, Xin Tao, Pengfei Wan, Jiwen Lu, and Jie Zhou. Astra: General interactive world model with autoregressive denoising. In *International Conference on Learning Representations*, volume 2026, pp. 79167–79184, 2026.

A NavAnywhere Dataset Details

We introduce NavAnywhere, a large-scale, action-free visual navigation dataset assembled from 15 sources, including public datasets and our in-house recordings. The collection consists of RGB videos and sampled image sequences without action annotations, comprising 70,756 sequences, 17,496,570 sampled frames, and 1,188.65 hours of source video. NavAnywhere covers a broad spectrum of environments encountered in everyday life, ranging from residential and public indoor spaces to urban streets, parks, and natural outdoor environments. (see Figure 6)

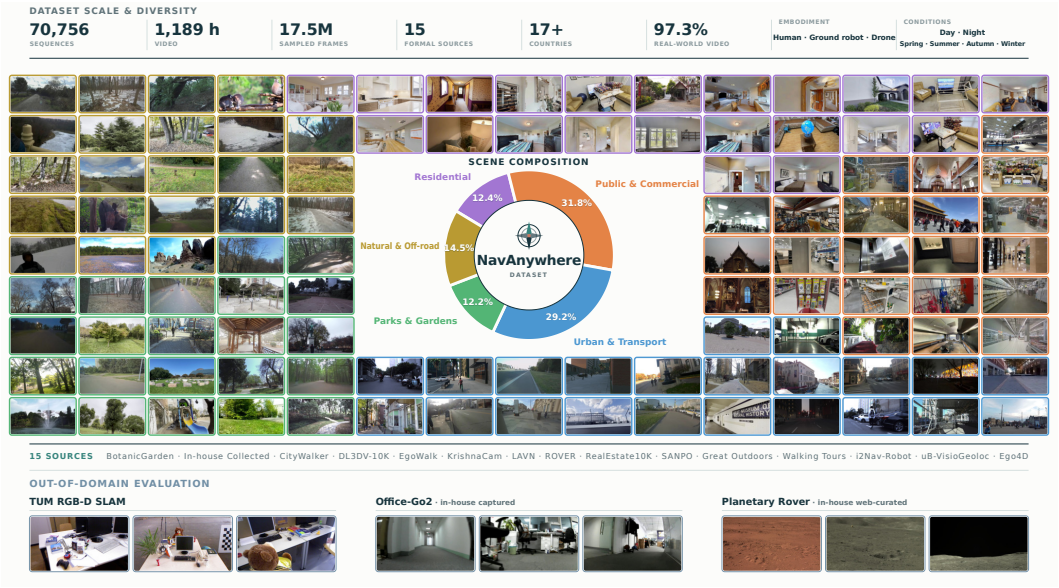

Figure 6: **Overview of NavAnywhere and its complementary out-of-domain evaluation sets.** NavAnywhere covers diverse everyday navigation environments. The complementary evaluation sets are separate from NavAnywhere and are described in Section B.2.

A.1 Dataset Composition

NavAnywhere comprises the following 15 sources:

- **BotanicGarden (Liu et al., 2024):** Robot-mounted observations of botanical gardens, featuring vegetation, pathways, and other outdoor structures.
- **CityWalker Liu et al. (2025):** Urban navigation sequences featuring sidewalks, intersections, pedestrians, and vehicles.
- **DL3DV-10K (Ling et al., 2024):** Scene-exploration videos spanning diverse indoor and outdoor environments, viewpoints, and camera motions.
- **Ego4D Grauman et al. (2022):** A navigation-related subset of egocentric videos depicting everyday activities and movement through indoor and outdoor environments.
- **EgoWalk (Akhtyamov et al., 2025):** First-person walking videos covering urban, residential, public, and recreational environments.
- **KrishnaCam (Singh et al., 2016):** Long-term egocentric recordings of daily activities, including walking, commuting, and visits to indoor and outdoor spaces.
- **LAVN (Johnson et al., 2025):** A collection combining real-world recordings with simulated navigation sequences.
- **ROVER (Schmidt et al., 2025):** Robot navigation recordings collected in gardens, parks, and campus environments across different seasons and illumination conditions.
- **RealEstate10K (Zhou et al., 2018):** Clips from real-estate videos depicting residential interiors, building exteriors, and surrounding areas, including aerial footage captured by drones.

- **SANPO (Waghmare et al., 2025):** Real and synthetic egocentric navigation sequences covering sidewalks, roads, parks, and other outdoor environments.

- **The Great Outdoors (Jiang et al., 2026a):** Robot-mounted observations of natural and off-road environments, including forests, unpaved trails, and uneven terrain.

- **Walking Tours (Venkataramanan et al., 2024):** Long-form internet videos depicting walking tours and outdoor exploration.

- **i2Nav-Robot (Tang et al., 2025):** Robot navigation sequences covering campus buildings, streets, parking areas, and transitions between indoor and outdoor spaces.

- **uB-VisioGeoloc (Scalvini et al., 2024):** Pedestrian-view urban sequences covering streets, public spaces, and parks, together with synthetic sequences.

- **In-house collected data:** Our in-house collection of nine RGB videos, covering offices, residential neighborhoods, and public landmarks across indoor and outdoor environments under daytime and nighttime conditions.

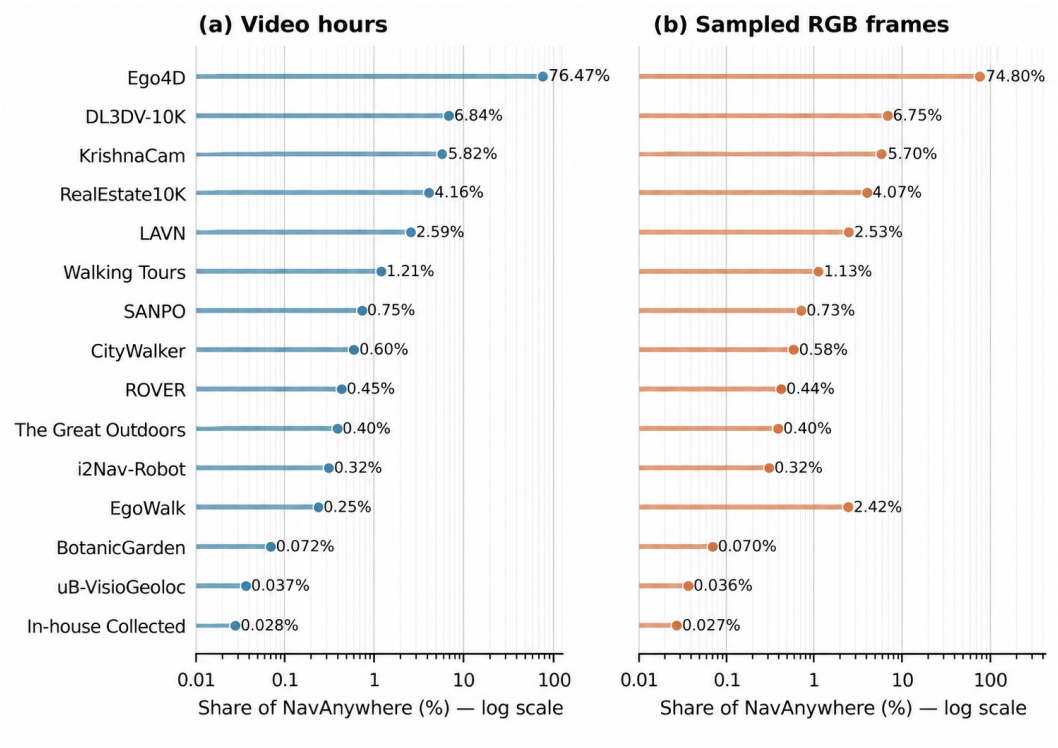

Figure 7: **Source composition of NavAnywhere.** Statistics cover the 15 action-free sources.

Table 6: **NavAnywhere dataset statistics.** Sequences denote processed video clips or trajectories, and frames denote sampled RGB images. Hours measure source-video duration. All 15 sources are used as RGB-only, action-free data.

Dataset	Sequences	Frames	Hours
Ego4D	1,619	13,088,098	908.95
DL3DV-10K	20,352	1,180,382	81.36
KrishnaCam	436	996,666	69.21
RealEstate10K	45,010	712,145	49.44
LAVN	312	443,244	30.78
Walking Tours	10	198,097	14.40
SANPO	2,662	128,233	8.91
CityWalker	18	102,083	7.09
ROVER	39	76,442	5.31
The Great Outdoors	17	69,162	4.80
i2Nav-Robot	10	55,202	3.83
EgoWalk	239	423,389	2.94
BotanicGarden	7	12,272	0.85
uB-VisioGeoloc	16	6,382	0.44
In-house Collected	9	4,773	0.33
NavAnywhere total	**70,756**	**17,496,570**	**1,188.65**

Table 6 and Figure 7 reports the complete NavAnywhere dataset statistics.

A.2 Scene and Environment Diversity

NavAnywhere spans five broad scene categories:

- **Residential environments:** Homes, residential buildings, and surrounding neighborhoods.
- **Public and commercial environments:** Offices, schools, shops, restaurants, and cultural facilities.
- **Urban and transportation environments:** Streets, sidewalks, intersections, parking areas, and transit spaces.
- **Parks and managed outdoor environments:** Gardens, campuses, recreational areas, and maintained green spaces.
- **Natural and off-road environments:** Forests, trails, uneven terrain, and unstructured outdoor spaces.

Across these categories, the constituent sources exhibit substantial variation in viewpoints, illumination conditions, scene layouts, and motion patterns.

B Dataset Details and Preprocessing

In addition to NavAnywhere, we use five action-annotated navigation datasets and three complementary out-of-domain evaluation sets. These datasets are not part of NavAnywhere and are excluded from its corpus statistics. This section describes these datasets, the preprocessing procedures applied to all data sources, and their respective roles in OpenNWM training and evaluation.

B.1 Action-Annotated Datasets

Following the dataset usage in NWM (Bar et al., 2025), we separately use RECON, HuRoN, SCAND, TartanDrive, and Go Stanford. Their processed trajectories provide RGB observations, planar positions, and yaw, from which relative waypoint actions $(\Delta x, \Delta y, \Delta\psi)$ are derived in the reference frame of the current observation.

Go Stanford is reserved exclusively for unseen-environment evaluation. The other four datasets provide motion supervision according to the LAM and OpenNWM training protocols described in Section B.4.

Table 7: **Statistics of the action-annotated datasets.** Sequences denote processed trajectories, and frames denote sampled RGB images. Go Stanford is used only for evaluation.

Dataset	Sequences	Frames	Data used
RECON	11,835	610,911	RGB + position/yaw
HuRoN	2,839	230,911	RGB + position/yaw
Go Stanford	3,696	198,126	RGB + position/yaw
SCAND	604	103,457	RGB + position/yaw
TartanDrive	1,251	62,884	RGB + position/yaw

Table 7 summarizes the processed collections.

B.2 Complementary Out-of-Domain Evaluation Sets

We curate three challenging real-world out-of-domain evaluation sets to assess transfer beyond the training sources: TUM RGB-D, Planetary Rover, and Office-Go2. All three are held out from training.

TUM RGB-D. We select five sequences from the TUM RGB-D SLAM benchmark (Sturm et al., 2012), using RGB observations and the provided ground-truth camera poses. The processed subset contains 1,189 sampled frames.

Planetary Rover. We curate extraterrestrial navigation observations from publicly available online sources, using released imagery and official pose metadata from the Chang'e-4/Yutu-2 lunar mission and the Tianwen-1/Zhurong Martian mission. The collection contains 66 lunar and 173 Martian RGB images, organized into 28 sequences with 211 adjacent transitions whose relative motions are derived from pose metadata.

Office-Go2. We collect navigation recordings in-house using a Unitree Go2 quadruped robot in a real office environment. The collection includes RGB videos and synchronized camera-pose measurements. Its seven recordings are processed into 15 sequences containing 2,292 sampled frames.

B.3 Data Processing

Temporal sampling. We organize sampled frames in temporal order within each sequence.

Most NavAnywhere sources are sampled at 4 Hz, corresponding to a frame interval of 0.25 s. EgoWalk, in contrast, is originally recorded at 100,Hz to capture relatively fast ego-motion. So EgoWalk sequences have an effective sampling rate of approximately 40 Hz, corresponding to a 0.025 s interval.

Image-pair sampling for LAM training. For LAM training, image pairs are sampled within the same sequence using signed frame offsets $\delta \in \{-8, \dots, 8\}$. These offsets correspond to temporal displacements of up to 2 s for 4 Hz sources and approximately 0.2 s for EgoWalk.

Context and target sampling for pretraining. For pretraining, each example contains four consecutive context frames and four target frames sampled at signed offsets within $[-64, 64]$ relative to the latest context frame. At 4 Hz, these offsets span $[-16, 16]$ s; for EgoWalk, they span approximately $[-1.6, 1.6]$ s. Latent-action conditioning is available for local image pairs satisfying $|\delta| \leq 8$.

Processing of action-annotated and evaluation data. The five external action-annotated datasets are processed into 4 Hz trajectories with image-aligned planar positions and yaw. TUM RGB-D and Office-Go2 similarly associate sampled RGB frames with camera poses. Planetary Rover preserves the ordering of the released observations and their endpoint poses without temporal image interpolation.

B.4 Dataset Usage in OpenNWM

Action-free learning from NavAnywhere. NavAnywhere provides action-free visual observations for latent-action learning and world-model pretraining. Its training inputs consist only of RGB observations, even when the original sources provide additional sensing modalities or annotations. Latent actions are inferred from image pairs by the learned latent action model.

Physical-motion supervision and evaluation. Physical-motion grounding of the LAM uses the training splits of RECON, HuRoN, and SCAND. Action-conditioned post-training uses RECON, HuRoN, SCAND, and TartanDrive. Go Stanford and all three complementary out-of-domain evaluation sets are excluded from these training stages.

C Architecture and Implementation Details

C.1 Training and Inference Details

Table 8 summarizes the training and inference configurations of OpenNWM used in the main experiments.

Latent-action learning. Our Pixel+Action navigation LAM follows the spatiotemporal Transformer architecture of Gao et al. (2026). It contains approximately 709M parameters, comprising a 24-block spatiotemporal encoder and a 24-block spatial decoder, both with a hidden width of 1,024 and 16 attention heads. The encoder maps a pair of RGB frames to a 32-dimensional variational latent. Frames are center-cropped to a 4:3 aspect ratio and resized to 240×320 (height $\times$ width), with 16×16 patches. Frame pairs use uniformly sampled integer temporal offsets in $[-8, 8]$.

LAM training combines NavAnywhere with the training splits of RECON, HuRoN, and SCAND, reserving 25% of each batch for action-labeled samples. The objective combines RGB reconstruction MSE, action MSE on labeled samples, and KL regularization, with respective weights 1, 0.171244, and 10^{-6}. We train from scratch using AdamW with a global batch size of 160, a peak learning rate of 2.5×10^{-5}, and weight decay of 0.01, using BF16 on eight NVIDIA L20 GPUs. The learning-rate schedule is configured for 100K steps, with 12.8K steps of linear warm-up followed by cosine decay toward 10% of the peak rate. We use the 60K-step LAM checkpoint to extract deterministic posterior means as latent actions for pretraining.

World-model pretraining. OpenNWM uses a CDiT-B/2 diffusion transformer with approximately 197M parameters and a frozen VAE tokenizer. Images are resized to 224×224. Each observation provides four context frames and four sampled prediction targets. Pretraining uses the NavAnywhere subset, with signed target frame offsets in $[-64, 64]$. All samples receive relative-time conditioning. Valid pairs within $[-8, 8]$ additionally receive their layer-normalized 32-dimensional latent actions; pairs outside this range contribute to the diffusion objective without latent-action conditioning.

We use the 60K-step pretraining checkpoint, trained with AdamW at a learning rate of 2×10^{-4}, weight decay of 0.01, and gradient clipping at 2.0. Training uses BF16 on eight NVIDIA L20 GPUs, with 48 observations per GPU, giving a global batch of 384 observations and 1,536 context-target pairs per update.

World model post-training. For post-training, we replace the latent action adapter with a newly initialized adapter for three-dimensional physical motion $(\Delta x, \Delta y, \Delta\psi)$. We first train this adapter for 3K steps with the backbone frozen, followed by 100K steps of joint post-training of the adapter and backbone. The adapter learning rate is 10^{-4} throughout post-training, and the backbone learning rate is 10^{-4} during the joint phase. Both phases use AdamW with weight decay of 0.01 and the same batch size, precision, and hardware as pretraining.

Post-training uses the training splits of RECON, HuRoN, SCAND, and TartanDrive. Go Stanford is excluded from training. Physical-motion targets are expressed relative to the latest context frame, with translations divided by dataset-specific waypoint spacing before fixed min–max normalization.

Inference and planning. Visual prediction uses 250-step DDPM (Ho et al., 2020) sampling with fixed evaluation windows and seeds. Planning evaluates sequences of eight physical-motion actions at $0.25\,\mathrm{s}$ intervals, corresponding to a $2\,\mathrm{s}$ horizon. The cross-entropy method (CEM) uses 80 candidates, five elites, and one proposal update. Candidate costs are computed using LPIPS-Alex

Table 8: **Training and inference configurations of OpenNWM.** Batch sizes are global, and image resolutions are reported as height $\times$ width.

Module	Configuration	Value
Latent Action Model	Architecture	Pixel+Action spatiotemporal Transformer
	Model size	$\sim$709M
	Encoder / decoder blocks	24 / 24
	Hidden width / attention heads	1,024 / 16
	Latent-action dimension	32
	Input frames	2
	Input resolution / patch size	240×320 / 16×16
	Temporal offsets	Integers in $[-8, 8]$
	Action-labeled batch fraction	25%
	RGB / action / KL loss weights	1 / 0.171244 / 10^{-6}
	Training steps	60K
	Global batch size	160
	Optimizer	AdamW
	Peak learning rate / weight decay	2.5×10^{-5} / 0.01
	LR warm-up / schedule length	12.8K / 100K steps
	LR decay / minimum LR ratio	Cosine / 0.1
	Precision / training hardware	BF16 / 8$\times$ NVIDIA L20
Navigation World Model	Backbone / model size	CDiT-B/2 / $\sim$197M
	Image tokenizer	Frozen SD-VAE
	Input resolution	224×224
	Context frames / targets per observation	4 / 4
	Target frame offsets	Integers in $[-64, 64]$
	Latent-action conditioning range	$\lvert\Delta t\rvert \leq 8$
	Pretraining steps	60K
	Pretraining learning rate	2×10^{-4}
	Physical-action adapter warm-up	3K steps
	Joint post-training steps	100K
	Post-training LR: adapter / backbone	10^{-4} / 10^{-4}
	Global observation / prediction batch	384 / 1,536
	Optimizer / weight decay	AdamW / 0.01
	Gradient clipping	2.0
	Precision / training hardware	BF16 / 8$\times$ NVIDIA L20
Inference and Planning	Diffusion sampler / sampling steps	DDPM / 250
	Planning action representation	Physical motion $(\Delta x, \Delta y, \Delta\psi)$
	Planning horizon	2 s
	Action interval / sequence length	0.25 s / 8
	CEM candidates	80
	CEM elites	5
	CEM updates	1
	Stochastic evaluations per candidate	3
	Planning cost	LPIPS-Alex

and averaged over three stochastic evaluations. Planning conditions the post-trained world model directly on physical motion.

Ablation settings. The ablations retain the same CDiT-B/2 backbone, image resolution, context and target counts, and training data sources as the main experiments. The main difference are less NavAnywhere data to train the LAM, 200K pretraining steps at a learning rate of 10^{-4}, and a global batch of 128 observations. Post-training uses a 10K-step adapter warm-up followed by 100K joint updates, with adapter and backbone learning rates of 10^{-4} and 10^{-5}, respectively. The ablations use the same 250-step DDPM and CEM80 planning protocol described above.

C.2 Evaluation Metrics

We evaluate visual prediction quality using three metrics: Peak Signal-to-Noise Ratio (PSNR), Learned Perceptual Image Patch Similarity (LPIPS), and Fréchet Inception Distance (FID), covering pixel-level fidelity, structural consistency, perceptual similarity, and distributional similarity.

Pixel-Level Fidelity. Peak Signal-to-Noise Ratio (PSNR) measures reconstruction fidelity on a logarithmic scale. Given the predicted image $\hat{\mathbf{I}}$ and the ground-truth image $\mathbf{I}$, it is defined as:

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{I_{\max}^2}{\frac{1}{N} \sum_{i=1}^{N} (\hat{I}_i - I_i)^2} \right), \tag{10}$$

where N denotes the total number of image elements, including all spatial locations and color channels, and $I_{\max}$ denotes the maximum possible intensity value under the adopted image representation. PSNR is expressed in decibels (dB), with higher values indicating higher pixel-level fidelity.

Perceptual Similarity. Learned Perceptual Image Patch Similarity (LPIPS) measures perceptual discrepancy in the feature space of a pretrained vision network. Given channel-normalized feature representations $\phi_l(\cdot)$ at layer l, it is defined as:

$$\mathrm{LPIPS}(\hat{\mathbf{I}}, \mathbf{I}) = \sum_{l} \frac{1}{H_l W_l} \sum_{h,w} \left\| \mathbf{w}_l \odot \left(\phi_l(\hat{\mathbf{I}})_{hw} - \phi_l(\mathbf{I})_{hw} \right) \right\|_2^2, \tag{11}$$

where $\mathbf{w}_l$ denotes the learned channel-wise weights, $\odot$ denotes element-wise multiplication, and $H_l \times W_l$ is the spatial resolution of the corresponding feature map. Lower LPIPS indicates higher perceptual similarity, complementing pixel-level and structural measures.

Distributional Similarity. Fréchet Inception Distance (FID) (Heusel et al., 2017) measures the discrepancy between the feature distributions of predicted and ground-truth image sets. Features are extracted using a pretrained Inception-v3 network, and each feature distribution is approximated by a multivariate Gaussian. Let $(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ and $(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$ denote the empirical means and covariance matrices of the predicted and ground-truth image features, respectively. FID is defined as:

$$\mathrm{FID} = \|\boldsymbol{\mu}_p - \boldsymbol{\mu}_r\|_2^2 + \mathrm{Tr}\left(\boldsymbol{\Sigma}_p + \boldsymbol{\Sigma}_r - 2 \left(\boldsymbol{\Sigma}_p \boldsymbol{\Sigma}_r \right)^{1/2} \right), \tag{12}$$

where $\mathrm{Tr}(\cdot)$ denotes the matrix trace and $(\cdot)^{1/2}$ denotes the matrix square root. Lower FID indicates closer agreement between the feature distributions of predicted and ground-truth images. Unlike PSNR and LPIPS, FID is computed over image sets rather than individual image pairs and does not directly measure prediction–target correspondence.

Together, these four metrics provide complementary assessments of visual prediction quality, from paired reconstruction fidelity to distribution-level similarity.

D Extended Quantitative Results

Consistency across Datasets. Table 9 shows that OpenNWM improves over NWM on every visual prediction metric across all four in-domain datasets. In comparison, RAE-NWM achieves stronger results on HuRoN but underperforms NWM on TartanDrive, revealing greater variation in its relative performance across datasets. The advantage of OpenNWM therefore extends beyond an improvement in the average score: its gains are distributed across the evaluated sources without degrading any reported dataset–metric combination relative to NWM. This consistency also extends to navigation (Table 10), where OpenNWM reduces ATE on every dataset while matching or improving RPE.

Visual Fidelity and Navigation Utility. SCAND reveals a mismatch between visual prediction and navigation rankings: RAE-NWM achieves better PSNR and DreamSim than OpenNWM, yet OpenNWM obtains lower ATE and RPE. Moreover, the visual metrics themselves disagree, with LPIPS favoring OpenNWM. These results indicate that individual image-similarity metrics are incomplete proxies for downstream navigation performance. Direct prediction measures agreement with an observed future, but does not directly assess whether predicted transitions support effective action selection. The navigation results therefore provide complementary evidence of the model's utility for control.

Table 9: **In-distribution (ID) direct visual prediction at the nominal 4 s horizon.** DS: DreamSim. Best and second-best results are bold and underlined, respectively.

Method	RECON			SCAND			HuRoN			TartanDrive		
	LPIPS↓	DS↓	PSNR↑	LPIPS↓	DS↓	PSNR↑	LPIPS↓	DS↓	PSNR↑	LPIPS↓	DS↓	PSNR↑
NWM	<u>0.323</u>	<u>0.120</u>	15.487	0.406	0.300	12.778	0.463	0.275	13.377	0.427	0.223	<u>13.602</u>
NWM XL (with Ego4D)	0.359	0.140	14.850	<u>0.394</u>	<u>0.260</u>	13.068	0.501	0.295	12.604	<u>0.424</u>	<u>0.213</u>	13.569
RAE-NWM	0.353	0.134	<u>15.491</u>	0.413	**0.190**	**14.419**	**0.339**	**0.171**	**15.079**	0.662	0.519	10.840
OpenNWM	**0.320**	**0.116**	**15.614**	**0.385**	0.271	<u>13.247</u>	<u>0.362</u>	<u>0.193</u>	<u>14.329</u>	**0.405**	**0.203**	**14.005**

Table 10: Navigation performance on RECON, SCAND, and TartanDrive. Best and second-best available results are bold and underlined, respectively.

Method	RECON		SCAND		TartanDrive	
	ATE↓	RPE↓	ATE↓	RPE↓	ATE↓	RPE↓
GNM	1.87	0.73	2.12	0.61	6.65	1.62
NoMaD	1.95	0.53	2.24	0.49	6.32	1.31
NWM	<u>1.13</u>	**0.35**	<u>1.01</u>	<u>0.28</u>	<u>5.72</u>	<u>1.19</u>
RAE-NWM	1.36	<u>0.37</u>	1.14	<u>0.28</u>	5.99	1.25
Garrido et al.	1.40	0.40	–	–	–	–
CompACT	1.33	0.39	1.36	0.34	–	–
OpenNWM (Ours)	**1.12**	**0.35**	**0.98**	**0.27**	**5.53**	**1.16**

E Scaling and Ablation Studies

We examine how the amount of unlabeled video used for latent-action learning affects the navigation world model. For this ablation, we compare four latent-action pretraining settings that retain 25%, 50%, 75%, or 100% of the unlabeled data pool. We denote this percentage by U and keep the action-label availability fixed. Each resulting model is then fine-tuned with real actions using the same downstream architecture, hyperparameters, and fine-tuning schedule. We evaluate all four models on the same 500 RECON windows for direct 4 s prediction. We report the mean DreamSim over these predictions; lower values indicate greater perceptual similarity to the ground-truth images.

Figure 8 shows a monotonic decrease in RECON DreamSim at $U = 25\%, 50\%, 75\%$, and 100%, respectively. Using the full unlabeled data pool reduces DreamSim by 6.08% relative to the 25% setting. The improvement continues from 75% to 100%. This supports the benefit of scaling unlabeled video for perceptual prediction quality.

F Latent-Action Visualization and Analysis

We further analyze whether learned Universal Navigation Latent Actions (UNLAs) capture physically grounded navigation semantics and generalize across heterogeneous environments. Specifically, we investigate two aspects: (1) physical alignment between latent actions and induced motions, and (2) cross-dataset consistency of latent action semantics.

F.1 Latent action learning paradigms.

As shown in Figure 9, we compare four paradigms for learning continuous latent actions from visual transitions (x_t, x_{t+1}). Each infers a variational latent z_t and applies KL regularization toward a standard Gaussian prior. **Action** predicts the observed navigation action a_t from z_t using an action head. **Pixel** reconstructs x_{t+1} conditioned on (x_t, z_t). **DINO feature** operates in a frozen DINOv3 (Siméoni et al., 2025) feature space: its encoder infers z_t from (f_t, f_{t+1}), where $f_t = \phi(x_t)$, and its decoder predicts f_{t+1} from (f_t, z_t). **Pixel + Action** jointly optimizes pixel reconstruction and action prediction from the same latent, applying action supervision only to labeled samples. All prediction losses use mean squared error; the joint objective weights the action term by λ. These paradigms examine how action supervision, visual reconstruction, and their combination shape the learned representation.

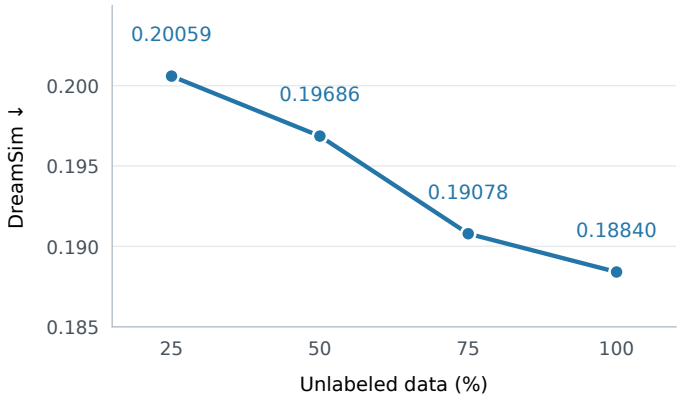

Figure 8: **Scaling unlabeled data ablation.** The horizontal axis gives the percentage of the unlabeled data pool used for latent-action pretraining, with action-label availability fixed.

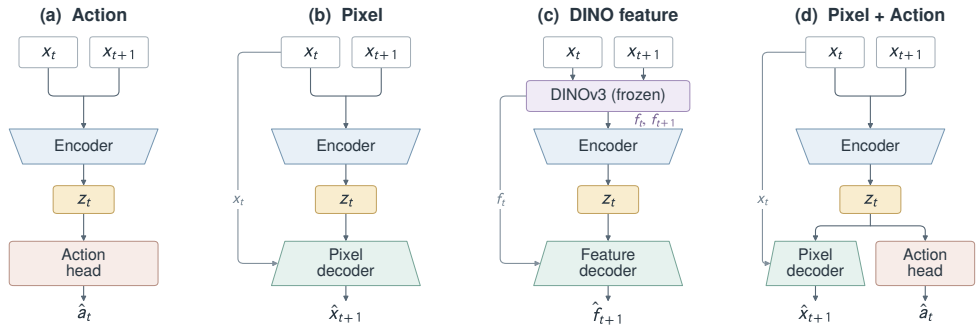

Figure 9: Four latent action learning paradigms. An encoder maps a visual transition to a variational latent z_t. The prediction targets are (a) the observed action a_t, (b) the future image x_{t+1}, (c) frozen DINOv3 features f_{t+1}, and (d) both the image and action. Reconstruction decoders additionally receive the current image or its features. In (d), pixel reconstruction uses all samples, while the action loss uses only labeled samples. All variants include the same form of KL regularization.

F.2 Physical Alignment of Latent Actions

To investigate whether latent actions encode meaningful navigation dynamics, we analyze the correspondence between latent actions and their associated physical motions. Given an anchor observation and a target observation, each model generates a 32-dimensional latent action representation. For each supervision paradigm, we independently perform K-means clustering in the original 32-dimensional latent space (rather than the projected visualization space) with $K = 8$, obtaining eight latent action clusters $C_1, \ldots, C_8$.

Importantly, clustering is performed solely in the original latent space, rather than the three-dimensional motion space used for visualization. After clustering, each sample is visualized in the physical motion space $(\Delta x, \Delta y, \Delta yaw)$, where each point represents an anchor-target observation pair and the color indicates its latent cluster assignment.

Compact clusters in the physical space indicate that samples sharing the same latent action cluster correspond to similar navigation motions. Conversely, highly overlapping clusters suggest that the latent action representation is affected by non-motion factors such as scene appearance, texture, or observation variations.

Figure 5 shows the clustering results under different supervision paradigms. Action supervision produces the most compact and clearly separated motion clusters, with different clusters aligning well with translational and rotational motion patterns. Joint Action-Pixel supervision achieves sim-

ilar motion-oriented structures, suggesting that visual supervision improves representation richness while physical supervision preserves motion semantics.

DINO-based self-supervision discovers partial motion structures, but different clusters exhibit substantial overlap in the physical space. This indicates that semantic visual features can capture high-level scene variations but are insufficient to fully distinguish controllable agent motion from other visual changes. In contrast, Pixel-level self-supervision produces the most entangled clusters, where different latent clusters often correspond to similar physical motions. This suggests that pixel-level objectives tend to encode appearance-related variations rather than navigation-specific dynamics.

Note that the presence of eight clusters alone does not indicate meaningful latent actions, since K-means always produces the predefined number of clusters. The key criterion is whether samples within the same cluster exhibit consistent physical effects. The results demonstrate that physical supervision plays a critical role in grounding latent actions in controllable navigation semantics.

F.3 Cross-Dataset Consistency of Latent Action Semantics

Beyond within-dataset motion alignment, we further evaluate whether latent action semantics remain consistent across heterogeneous environments and datasets. Unlike the previous analysis, which only examines motion structure within each dataset, this evaluation tests whether the learned latent action space provides a transferable navigation interface.

For each supervision paradigm, K-means clustering is first performed on latent representations from in-domain datasets using the original 32-dimensional latent space. The obtained cluster assignments are then fixed and directly transferred to unseen datasets without additional clustering or adaptation. Specifically, we evaluate transfer from the in-domain datasets RECON, HuRoN, and SCAND to the out-of-domain datasets TartanDrive and Go Stanford.

To quantify semantic consistency, we discretize each continuous physical motion transition $(\Delta x, \Delta y, \Delta\mathrm{yaw})$ into one of six navigation categories, where Δx and Δy denote the two planar displacement components and $\Delta\mathrm{yaw}$ denotes the change in yaw angle. The category set is defined as $\mathcal{C} = \{\mathrm{F}, \mathrm{B}, \mathrm{FL}, \mathrm{FR}, \mathrm{BL}, \mathrm{BR}\}$, corresponding to forward, backward, forward-left, forward-right, backward-left, and backward-right, respectively.

Each latent cluster is assigned the most frequent ground-truth motion category among the in-domain samples belonging to that cluster. The predicted motion category of a sample is then determined by the category assigned to its latent cluster. We measure the agreement between predicted and ground-truth motion categories using Macro-6 Recall, defined as the unweighted mean of the recalls over all six categories:

$$\text{Macro-6 Recall} = \frac{1}{6}\sum_{c\in\mathcal{C}} R_c = \frac{1}{6}\sum_{c\in\mathcal{C}} \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}, \tag{13}$$

where R_c denotes the recall for category c, TP_c is the number of samples whose ground-truth and predicted categories are both c, and FN_c is the number of samples whose ground-truth category is c but whose predicted category differs from c. This metric assigns equal weight to each motion category, regardless of its frequency in the evaluation set.

As shown in Figure 10, Action-only supervision achieves strong motion-semantic consistency on in-domain datasets, demonstrating that explicit motion supervision can align latent actions with navigation behaviors. However, its performance decreases on unseen datasets, suggesting that motion semantics learned from action-labeled data may still be affected by dataset-specific distributions.

Pixel-level self-supervision achieves weaker transfer performance, as visual representations may capture scene appearance and observation variations rather than consistent motion semantics. DINO-based self-supervision improves semantic transferability but still exhibits ambiguity in fine-grained navigation motions.

Joint Action-Pixel supervision achieves the strongest cross-dataset consistency, combining the scalability of visual supervision with the physical grounding provided by motion annotations. These results demonstrate that physical grounding not only improves within-dataset motion alignment but also enables latent actions to maintain consistent navigation semantics across heterogeneous environments. Overall, the analysis demonstrates that UNLAs are not arbitrary latent codes but a shared

and transferable interface that bridges large-scale visual experiences and physically grounded navigation dynamics.

Figure 10: Visualization of the semantic separability of latent actions learned by four LAM paradigms across different navigation action categories. TartanDrive and Go Stanford are out-of-domain datasets.

G QUALITATIVE RESULTS

We present qualitative examples of OpenNWM's visual predictions, navigation trajectories, and LAM image reconstructions across diverse environments.

Figure 11: Video generation examples on RECON. NWM, RAE-NWM and OpenNWM are conditioned on a single first image, and a ground truth trajectory and autoregressively predicts the next up to 16 seconds at 4 FPS. We plot the generated results from 0 to 16 seconds, every 2 second.

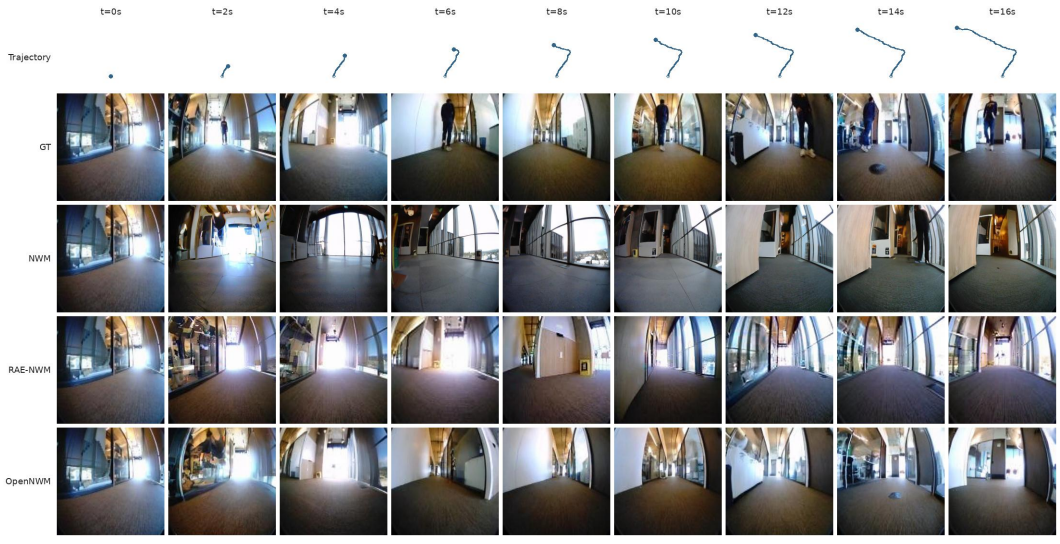

Figure 12: Video generation examples on HuRoN. NWM, RAE-NWM and OpenNWM are conditioned on a single first image, and a ground truth trajectory and autoregressively predicts the next up to 16 seconds at 4 FPS. We plot the generated results from 0 to 16 seconds, every 2 second.

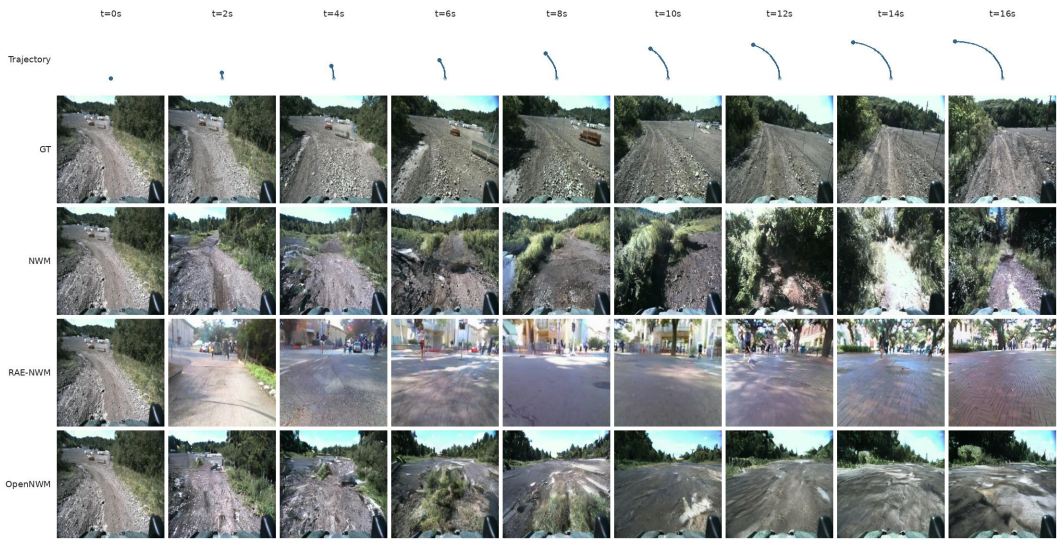

Figure 13: Video generation examples on TartanDrive. NWM, RAE-NWM and OpenNWM are conditioned on a single first image, and a ground truth trajectory and autoregressively predicts the next up to 16 seconds at 4 FPS. We plot the generated results from 0 to 16 seconds, every 2 second.

Figure 14: Video generation examples on Go Stanford (OOD). NWM, RAE-NWM and OpenNWM are conditioned on a single first image, and a ground truth trajectory and autoregressively predicts the next up to 4 seconds at 4 FPS. We plot the generated results from 0 to 4 seconds, every 1 second.

Figure 15: Qualitative results of our LAM on HuRoN. Each row shows the input image, ground-truth target image, reconstruction, and ground-truth versus decoded actions. Endpoint error (EPE, m) and yaw error (°) are reported for each example.

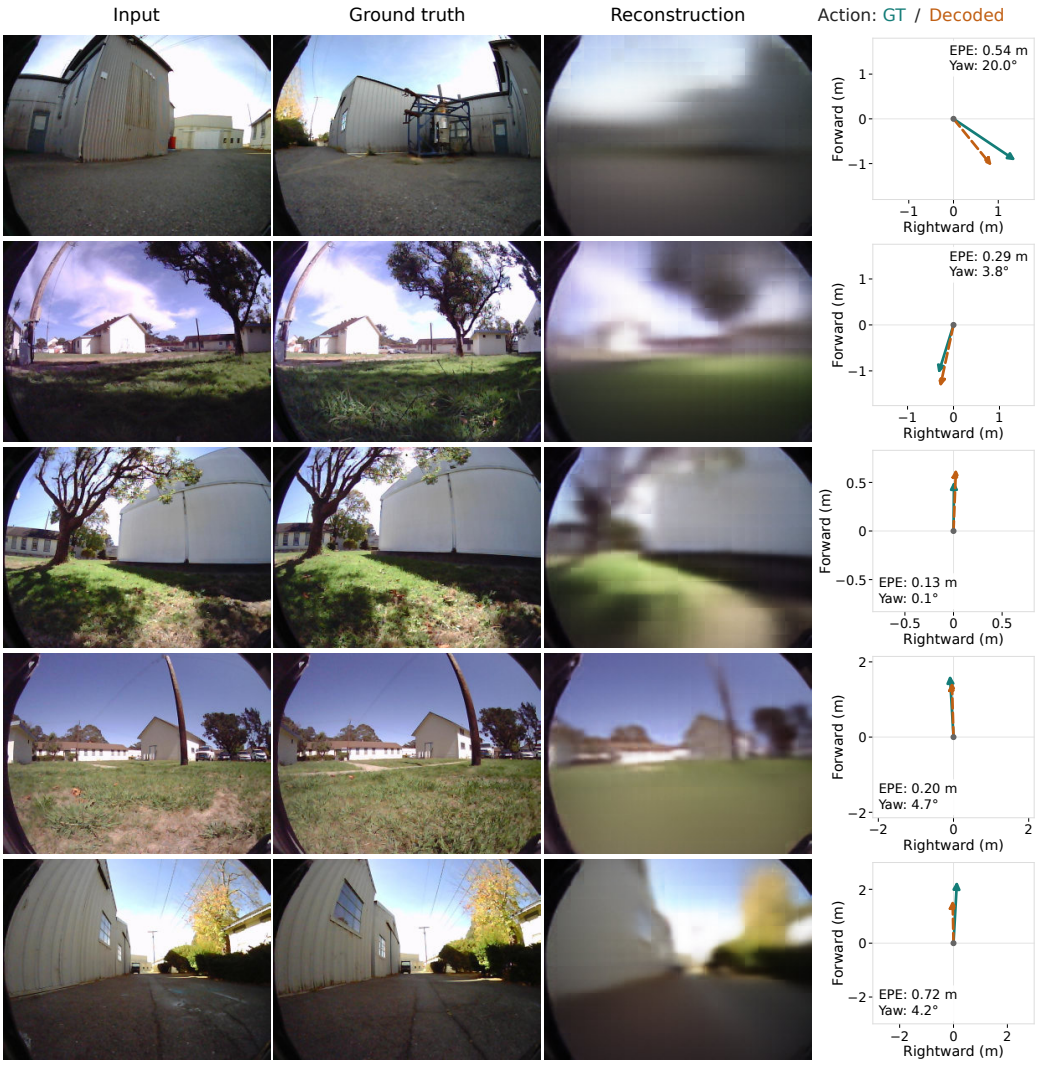

Figure 16: Qualitative results of our LAM on RECON. Each row shows the input image, ground-truth target image, reconstruction, and ground-truth versus decoded actions. Endpoint error (EPE, m) and yaw error (°) are reported for each example.

Figure 17: Qualitative results of our LAM on SCAND. Each row shows the input image, ground-truth target image, reconstruction, and ground-truth versus decoded actions. Endpoint error (EPE, m) and yaw error (°) are reported for each example.

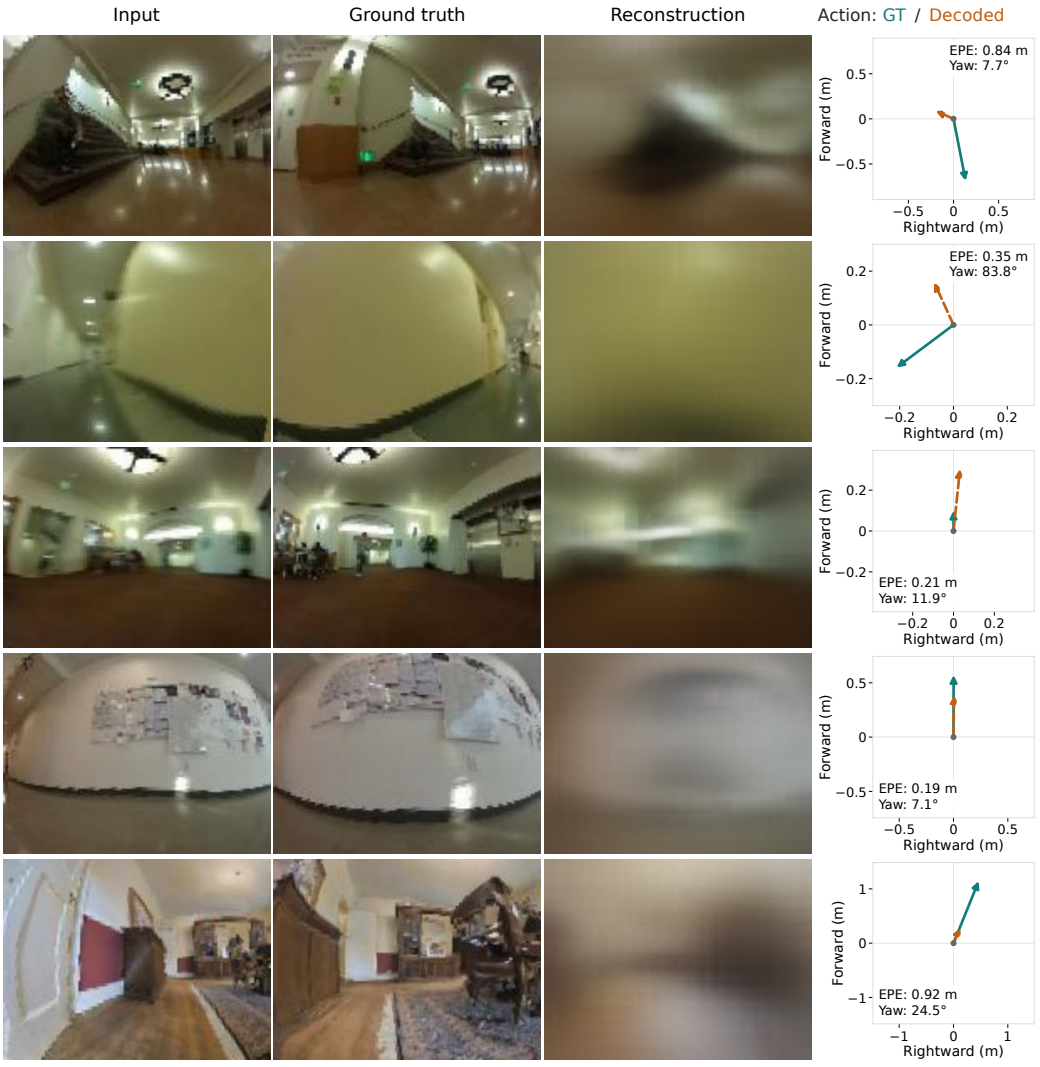

Figure 18: Qualitative results of our LAM on Go Stanford (OOD). Each row shows the input image, ground-truth target image, reconstruction, and ground-truth versus decoded actions. Endpoint error (EPE, m) and yaw error (°) are reported for each example.

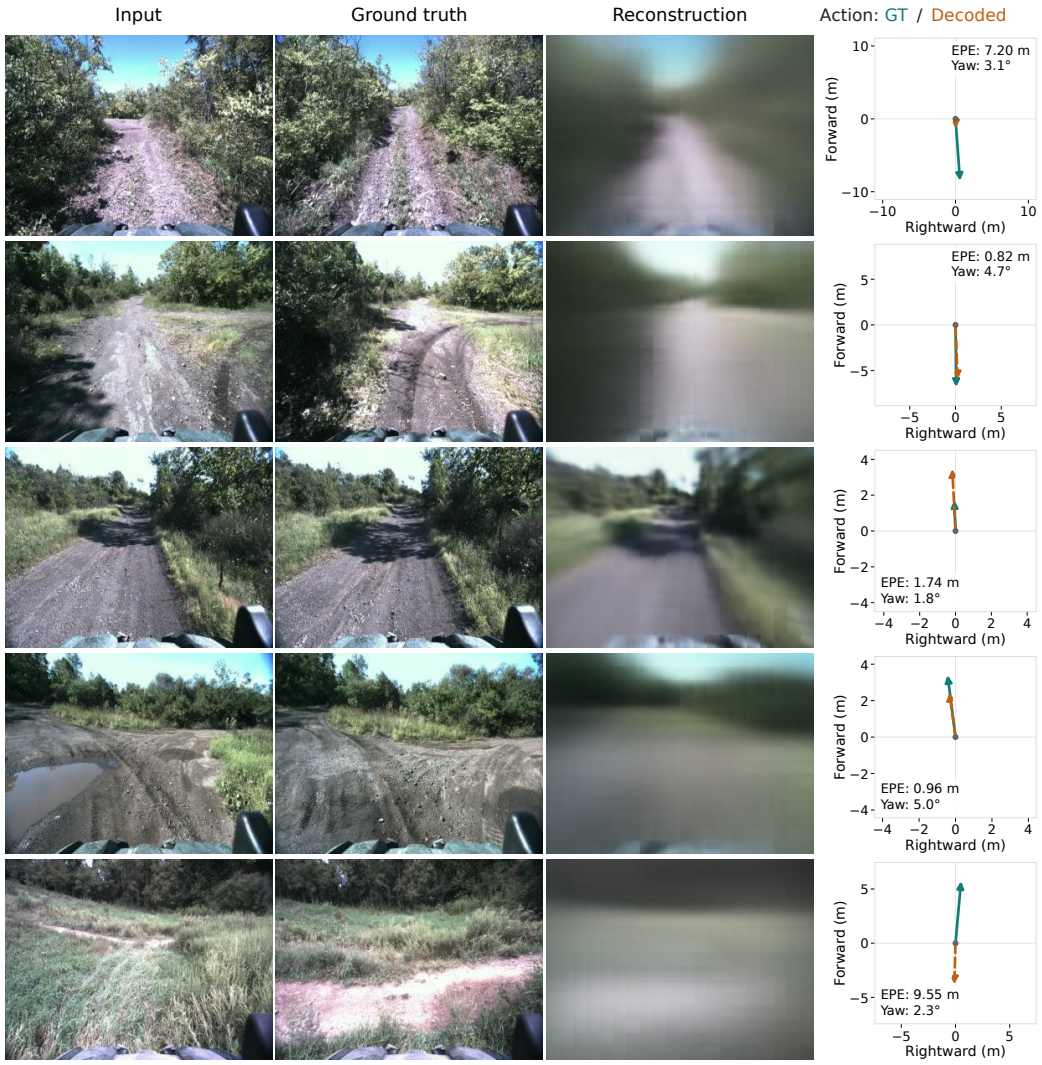

Figure 19: Qualitative results of our LAM on TartanDrive (OOD). Each row shows the input image, ground-truth target image, reconstruction, and ground-truth versus decoded actions. Endpoint error (EPE, m) and yaw error (°) are reported for each example.

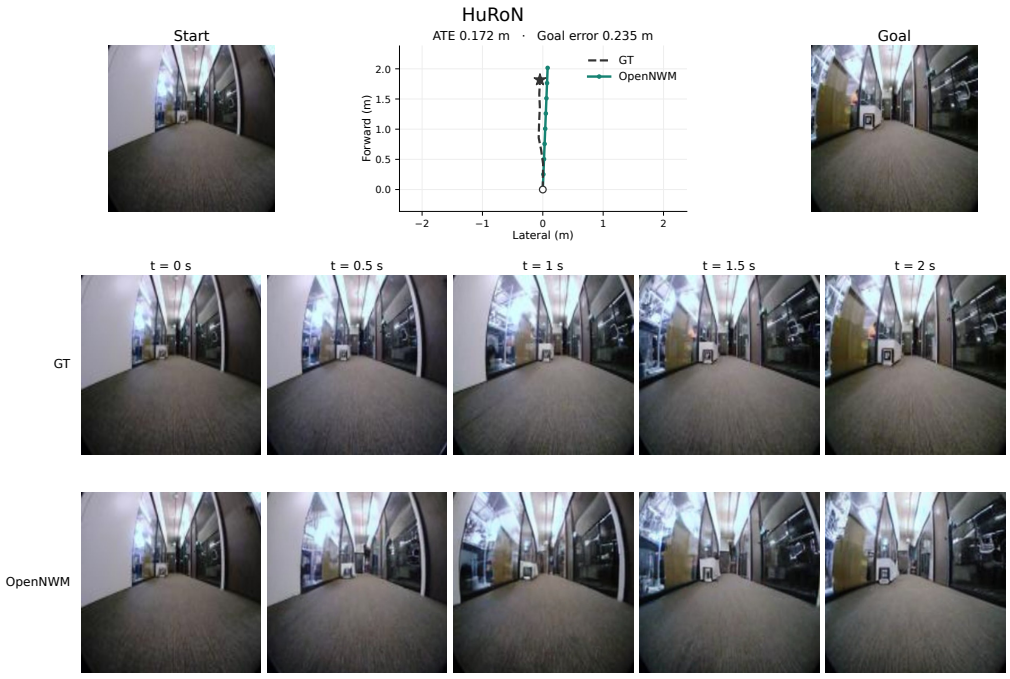

Figure 20: Qualitative navigation results of OpenNWM on HuRoN. The top row shows the start and goal images alongside ground-truth and predicted trajectories, with ATE and goal error reported in meters. The bottom rows compare ground-truth and predicted observations at 0.5-second intervals over a 2-second horizon.

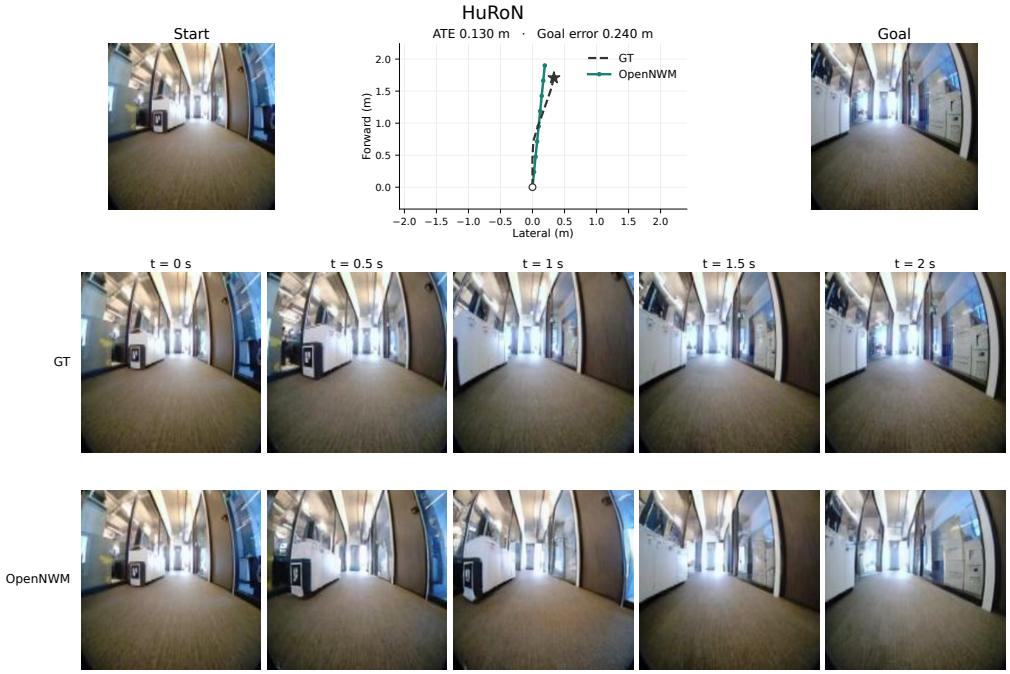

Figure 21: Qualitative navigation results of OpenNWM on HuRoN. The top row shows the start and goal images alongside ground-truth and predicted trajectories, with ATE and goal error reported in meters. The bottom rows compare ground-truth and predicted observations at 0.5-second intervals over a 2-second horizon.

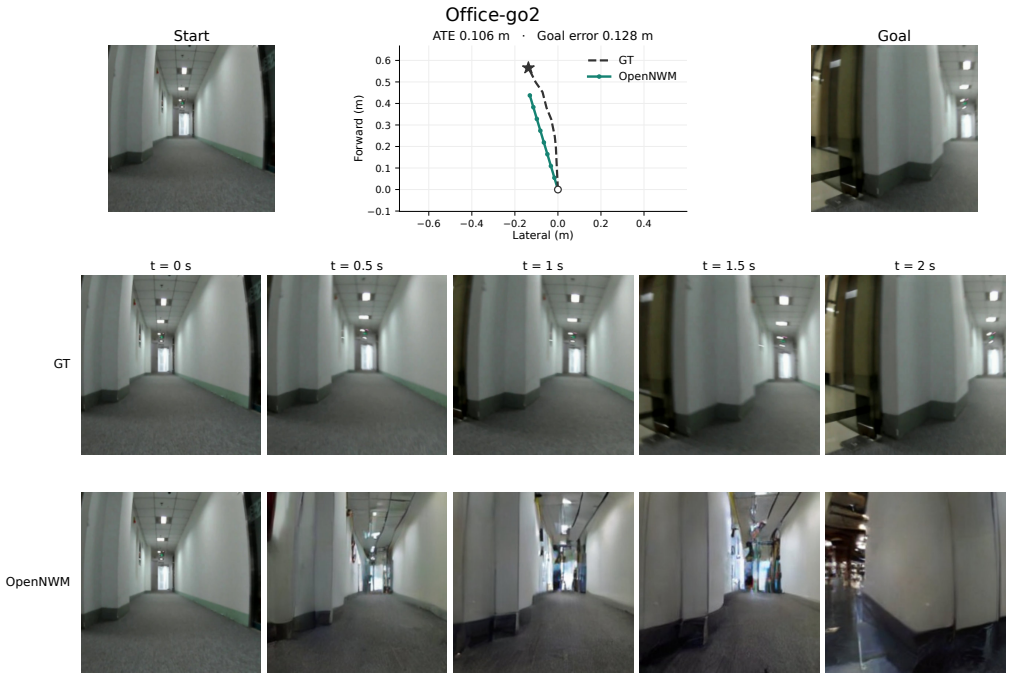

Figure 22: Qualitative navigation results of OpenNWM on Office-go2 (OOD). The top row shows the start and goal images alongside ground-truth and predicted trajectories, with ATE and goal error reported in meters. The bottom rows compare ground-truth and predicted observations at 0.5-second intervals over a 2-second horizon.

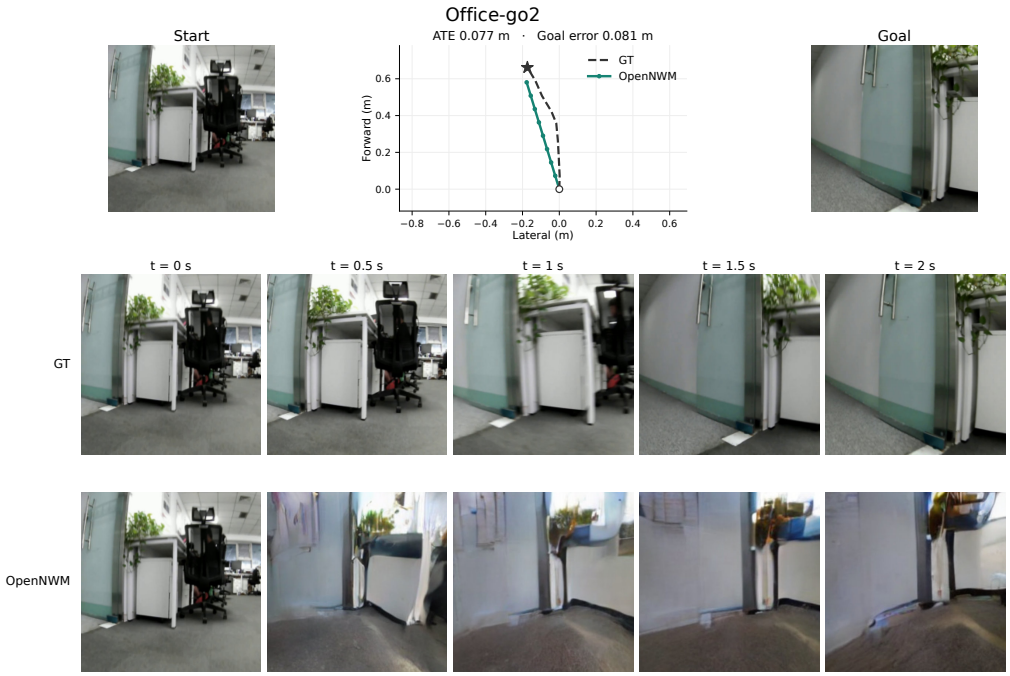

Figure 23: Qualitative navigation results of OpenNWM on Office-go2 (OOD). The top row shows the start and goal images alongside ground-truth and predicted trajectories, with ATE and goal error reported in meters. The bottom rows compare ground-truth and predicted observations at 0.5-second intervals over a 2-second horizon.

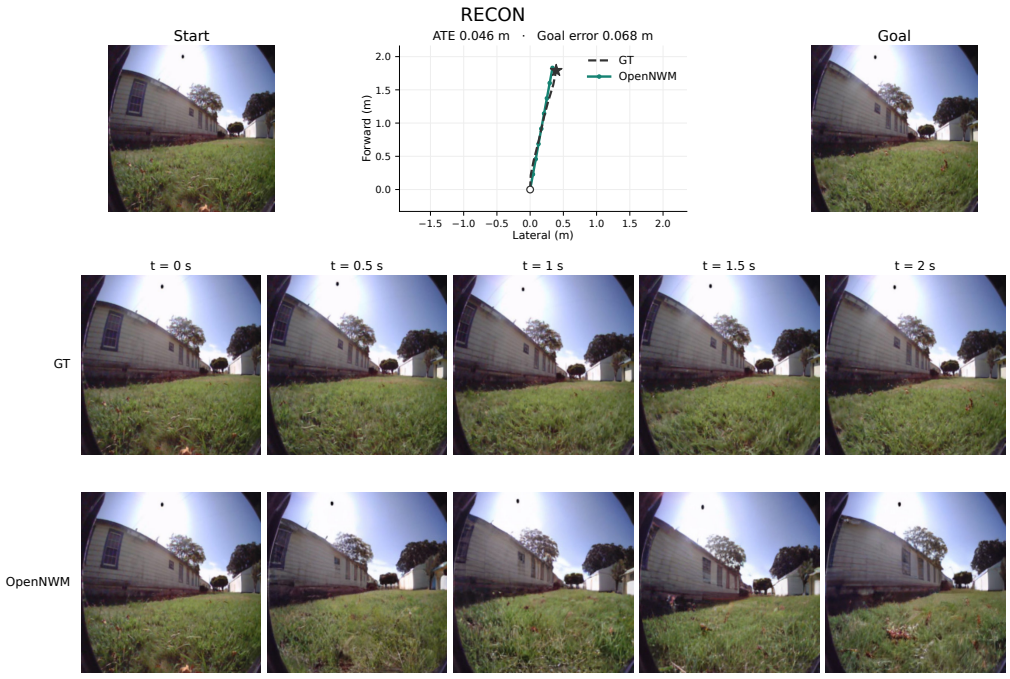

Figure 24: Qualitative navigation results of OpenNWM on RECON. The top row shows the start and goal images alongside ground-truth and predicted trajectories, with ATE and goal error reported in meters. The bottom rows compare ground-truth and predicted observations at 0.5-second intervals over a 2-second horizon.

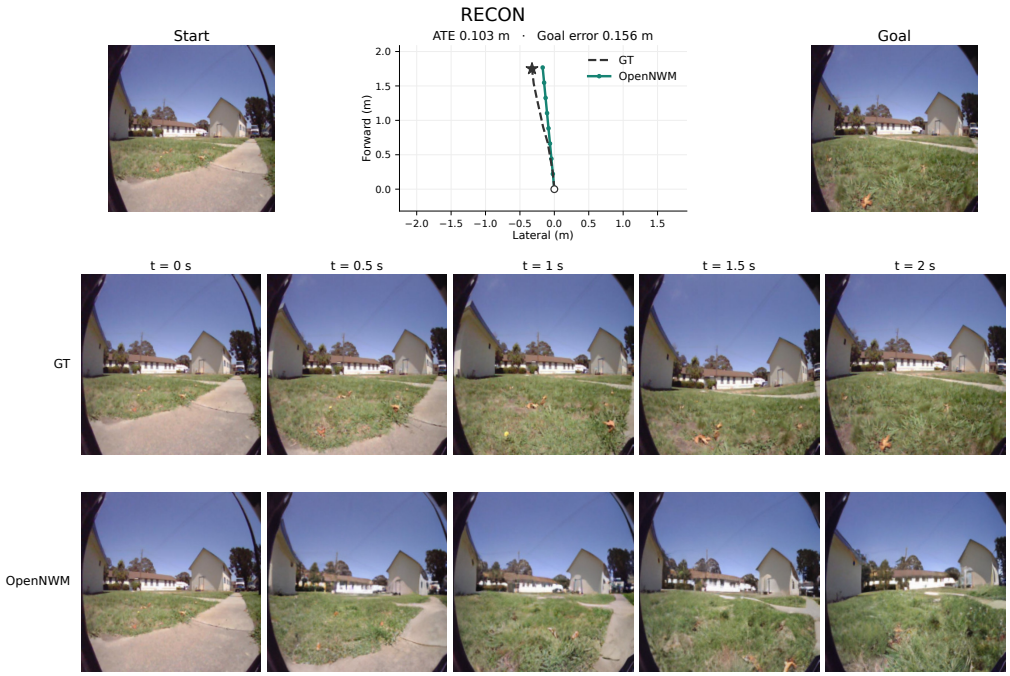

Figure 25: Qualitative navigation results of OpenNWM on RECON. The top row shows the start and goal images alongside ground-truth and predicted trajectories, with ATE and goal error reported in meters. The bottom rows compare ground-truth and predicted observations at 0.5-second intervals over a 2-second horizon.

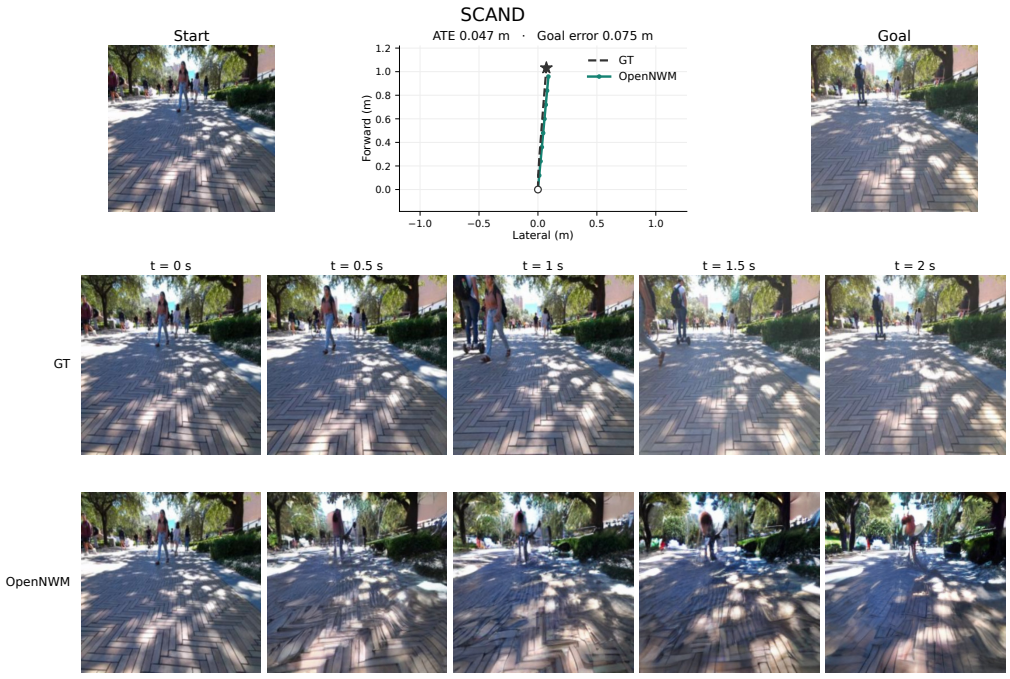

Figure 26: Qualitative navigation results of OpenNWM on SCAND. The top row shows the start and goal images alongside ground-truth and predicted trajectories, with ATE and goal error reported in meters. The bottom rows compare ground-truth and predicted observations at 0.5-second intervals over a 2-second horizon.

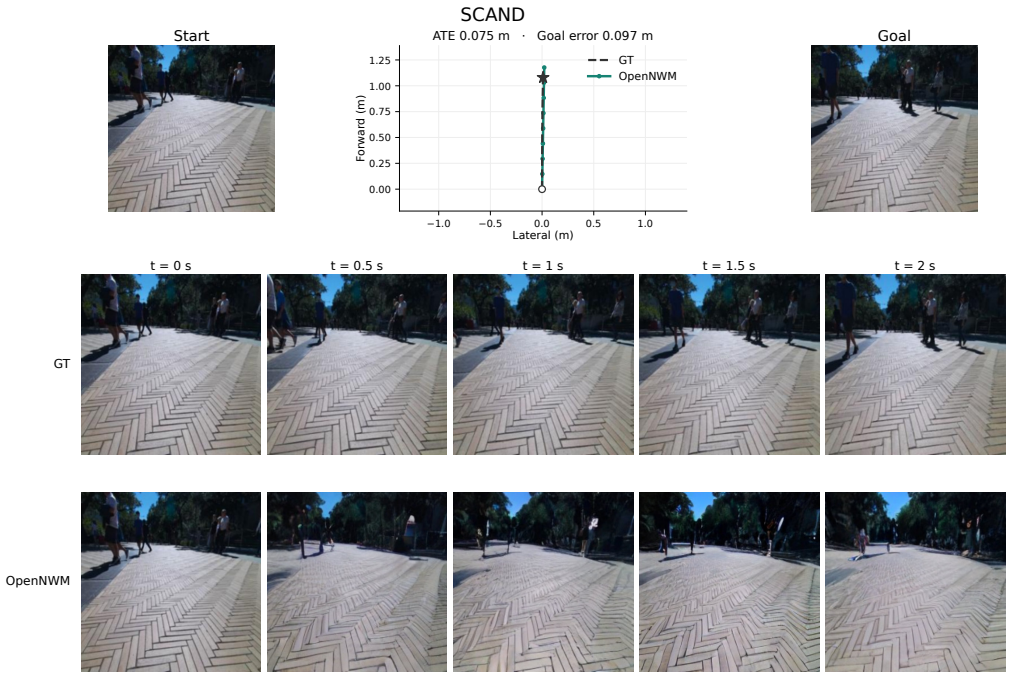

Figure 27: Qualitative navigation results of OpenNWM on SCAND. The top row shows the start and goal images alongside ground-truth and predicted trajectories, with ATE and goal error reported in meters. The bottom rows compare ground-truth and predicted observations at 0.5-second intervals over a 2-second horizon.

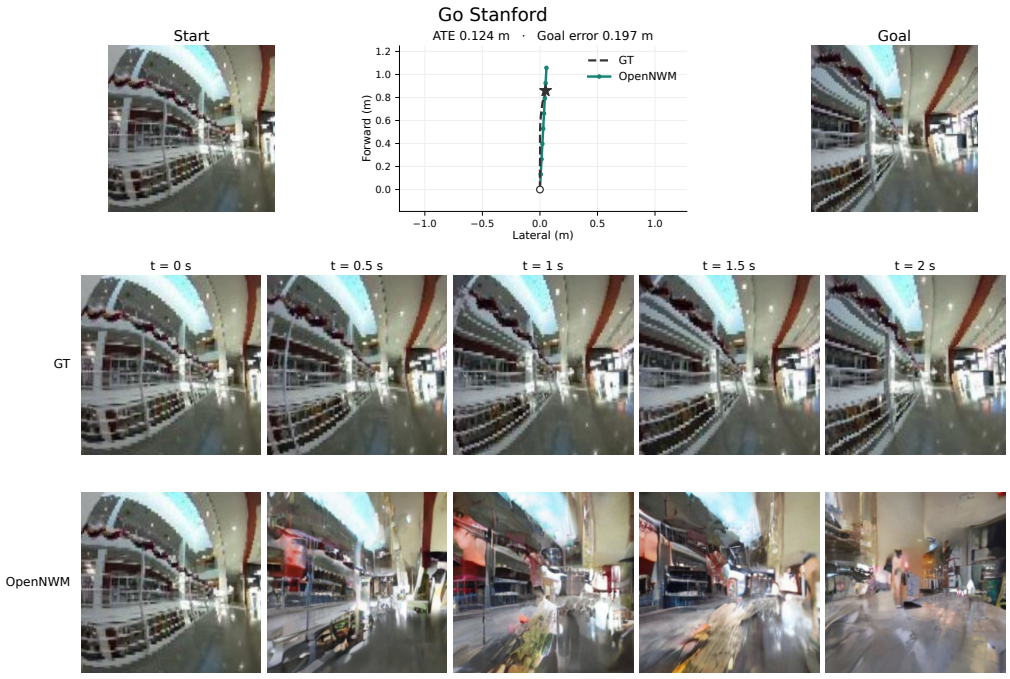

Figure 28: Qualitative navigation results of OpenNWM on Go Stanford (OOD). The top row shows the start and goal images alongside ground-truth and predicted trajectories, with ATE and goal error reported in meters. The bottom rows compare ground-truth and predicted observations at 0.5-second intervals over a 2-second horizon.

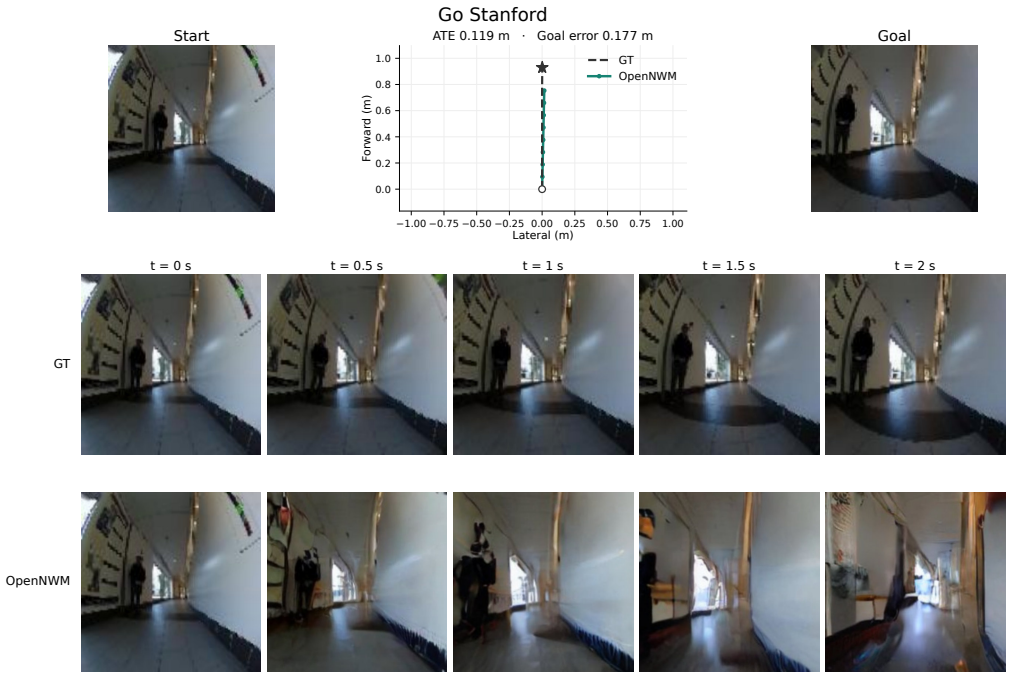

Figure 29: Qualitative navigation results of OpenNWM on Go Stanford (OOD). The top row shows the start and goal images alongside ground-truth and predicted trajectories, with ATE and goal error reported in meters. The bottom rows compare ground-truth and predicted observations at 0.5-second intervals over a 2-second horizon.

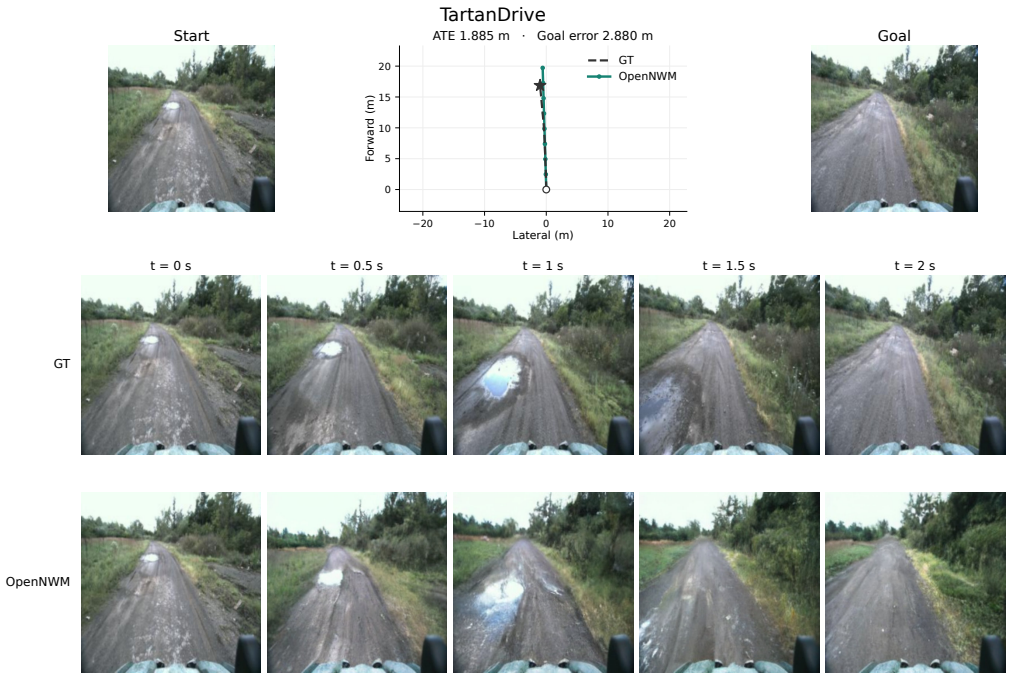

Figure 30: Qualitative navigation results of OpenNWM on TartanDrive. The top row shows the start and goal images alongside ground-truth and predicted trajectories, with ATE and goal error reported in meters. The bottom rows compare ground-truth and predicted observations at 0.5-second intervals over a 2-second horizon.

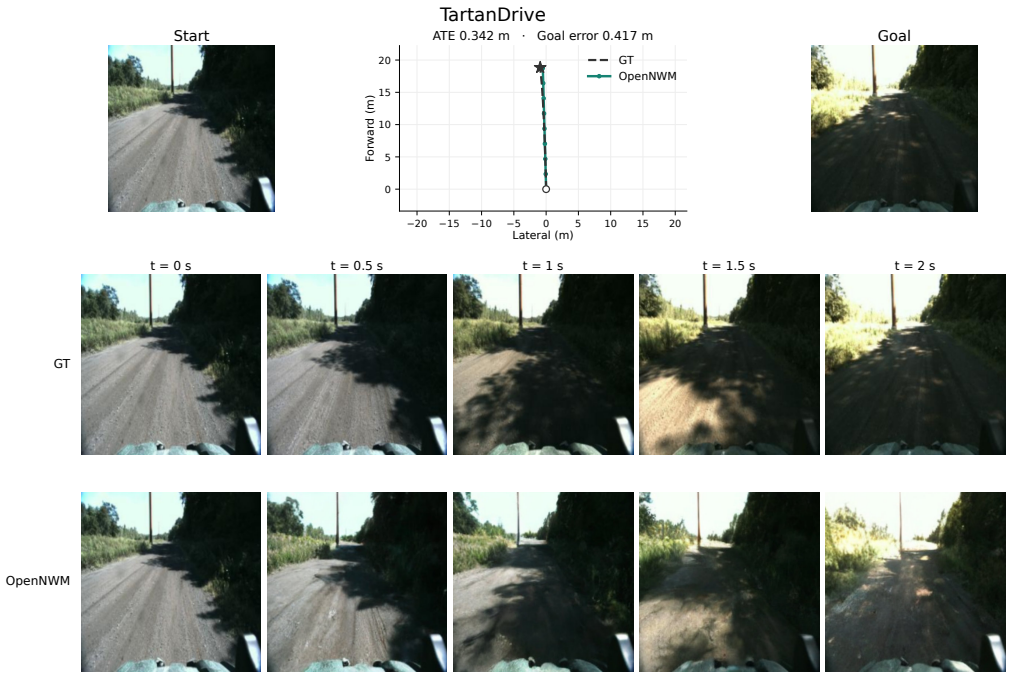

Figure 31: Qualitative navigation results of OpenNWM on TartanDrive. The top row shows the start and goal images alongside ground-truth and predicted trajectories, with ATE and goal error reported in meters. The bottom rows compare ground-truth and predicted observations at 0.5-second intervals over a 2-second horizon.