WORLDWEAVE: JOINTLY LEARNING WORLD DYNAMICS ACROSS EMBODIED NAVIGATION TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Mobile embodied agents continuously alter their observations through their own motion, while external dynamics further shape future interactions. Different mobile tasks therefore capture complementary patterns of state evolution, yet jointly learning world dynamics across such interactions remains less explored. We investigate this problem through Vision-and-Language Navigation (VLN) and Embodied Visual Tracking (EVT), which respectively involve navigation toward a stationary goal and interaction with a moving target. We propose WORLDWEAVE, a cross-task world model that couples shared predictive dynamics with task-specific adaptation. Instead of conditioning state transitions on heterogeneous physical actions, WORLDWEAVE learns latent actions from observed state changes via inverse–forward dynamics, providing a common transition representation across heterogeneous action spaces. The resulting future-state predictions are adaptively integrated into each task's representation for downstream decision making. Experiments across multiple VLN and EVT benchmarks demonstrate consistent improvements over specialized baselines.

1 INTRODUCTION

A mobile robot's observations are inherently shaped by its own actions: each movement changes what the agent perceives and alters its spatial relationship with the environment and task goal. This creates a continuous perception–action loop, in which current decisions shape future observations, which in turn inform subsequent decisions. Effective mobile decision making therefore requires not only understanding the current interaction state, but also anticipating how that state may evolve as the agent acts. The dynamics underlying this evolution can vary substantially with the task goal and interaction setting. When navigating toward a goal that remains fixed within an episode, state changes are primarily induced by the agent's own motion as it progresses through the environment. When interacting with a moving goal, state evolution additionally depends on external motion, requiring the agent to account for changes jointly induced by itself and the target. Vision-and-Language Navigation (VLN) (Anderson et al., 2018a; Krantz et al., 2020a). and Embodied Visual Tracking (EVT) (Zhong et al., 2019a; Wang et al., 2025c) instantiate these two settings with distinct decision requirements. VLN requires an agent to ground natural-language instructions in visual observations and navigate toward an instruction-specified goal over extended trajectories. EVT, in contrast, requires a mobile agent to actively follow a moving target through continuous perception and control, forming a closed-loop interaction jointly shaped by the motions of both the agent and target.

Despite these differences, VLN and EVT share a common predictive requirement: both must anticipate how the current embodied state may evolve to support subsequent decisions. In VLN, this involves the agent's progression through environmental structure over extended trajectories (Yao et al., 2025; Zhang et al., 2026e), whereas in EVT, it involves the continuously changing agent–target relationship (Xie et al., 2026). As illustrated in Fig. 1 (a), the two tasks thus provide complementary views of embodied state evolution: VLN emphasizes state changes induced by the agent's navigation through the environment, while EVT additionally captures changes jointly induced by the agent and a moving target. These different forms of state evolution provide complementary dynamics, while their common predictive requirement suggests that such dynamics may be shared across the two tasks rather than learned independently. This motivates our central hypothesis: *shared predictive dynamics can benefit both VLN and EVT despite their different goals and action spaces.*

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

A natural framework for modeling such predictive dynamics is a world model, which learns how latent states evolve through interaction (Ha & Schmidhuber, 2018). Conventional world models typically model state transitions conditioned on predefined actions, *i.e.*, $p(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \boldsymbol{a}_t)$ (Hafner et al., 2019b;a). While this formulation is well suited to individual tasks, extending it across heterogeneous tasks introduces additional challenges, as the actions conditioning state transitions may differ across task settings (Zhang et al., 2026a). In our setting, directly conditioning a shared world model on the heterogeneous actions of VLN and EVT would tie its learned dynamics to task-specific action definitions. Latent-action learning provides an alternative by inferring latent transition variables from state changes encoded in consecutive observations, rather than relying on predefined action labels (Bruce et al., 2024; Gao et al., 2026). This motivates a shared transition representation across heterogeneous embodied tasks.

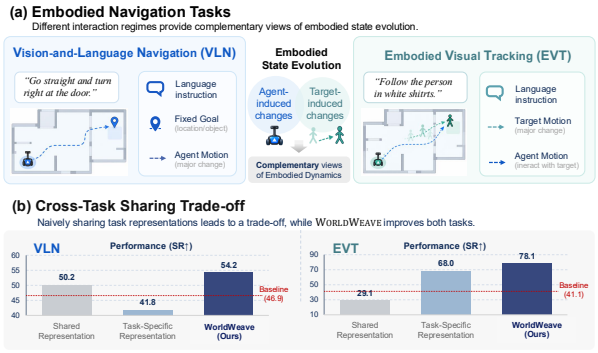

Figure 1: **Motivation of WORLDWEAVE.** VLN and EVT tasks exhibit complementary dynamics, while representation sharing leads to trade-offs.

Establishing such a transition representation, however, does not determine how the shared dynamics should support each task. VLN and EVT rely on predictive dynamics differently for downstream decision making, while naive sharing across heterogeneous tasks can introduce interference and negative transfer (Jeong & Yoon, 2024; 2025). Our preliminary experiments make this issue explicit. As shown in Fig. 1 (b), using a single shared adapter preserves VLN performance but yields little effective tracking capability for EVT. Introducing task-specific adapters substantially recovers EVT performance, whereas directly incorporating the shared world representation into VLN's decision pathway degrades navigation performance. These observations suggest that effective dynamics sharing requires not only a common transition representation, but also task-specific adaptation of the shared predictions to the distinct representations and decision processes of each task.

Motivated by these observations, we propose WORLDWEAVE, a cross-task world-modeling framework that couples shared predictive dynamics with task-specific adaptation for VLN and EVT. During pretraining, WORLDWEAVE learns state transitions through inverse–forward dynamics: latent actions are inferred from state changes encoded in consecutive observations and used to predict subsequent states. Because these latent actions characterize observed state transitions rather than task-specific physical actions, the learned dynamics are not explicitly tied to either task's action definition and can be pretrained from videos without action annotations. To enable prediction without access to future observations at inference time, the model further learns to infer these latent actions from current interaction context. The resulting future-state predictions are then adaptively integrated into each task's representation, allowing shared predictive dynamics to support distinct downstream decision processes. In this way, WORLDWEAVE shares how state transitions are modeled while preserving how predictive information is used for task-specific decision making.

Our contributions are summarized as follows:

- We investigate cross-task dynamics sharing in embodied navigation, focusing on Vision-and-Language Navigation (VLN) and Embodied Visual Tracking (EVT) as two complementary interaction settings with distinct forms of state evolution and decision making.
- We propose WORLDWEAVE, which couples shared latent dynamics with task-specific predictive adaptation. It learns latent transitions from state changes in consecutive observations without relying on task-specific action definitions, while adapting shared future-state predictions to distinct task representations and decision processes.
- Extensive experiments across multiple VLN and EVT benchmarks show consistent improvements over task-specific baselines, supporting the benefit of jointly learning world dynamics across complementary embodied interactions.

2 Related Work

Vision-and-Language Navigation. Vision-and-Language Navigation (VLN) requires agents to follow natural-language instructions through visual environments, evolving from discrete navigation graphs to continuous environments (Anderson et al., 2018a; Krantz et al., 2020a). Subsequent methods incorporate spatial representations and observation history for navigation reasoning (Chen et al., 2022b; Wang et al., 2023b; Liu et al., 2024; Chen et al., 2021), while recent LLM/VLM-based approaches improve instruction understanding and action prediction (Zhou et al., 2024b;a; Chen et al., 2024b; Chu et al., 2026). Predictive modeling has also been explored, e.g., NavMorph (Yao et al., 2025) learns latent state transitions to anticipate future navigation states.

Embodied Visual Tracking. Embodied Visual Tracking (EVT) extends active visual tracking to embodied agents that actively follow moving targets through egocentric perception and closed-loop control (Zhong et al., 2019a;b; Wang et al., 2025c). Recent methods leverage vision–language–action models for target understanding and motion decision making (Luo et al., 2018; Wang et al., 2025c; Liu et al., 2025), while USS (Xie et al., 2026) introduces latent predictive dynamics for temporal modeling. Despite predictive modeling in both VLN and EVT, existing approaches model their dynamics separately, leaving cross-task dynamics sharing largely unexplored.

World Models. World models learn predictive state dynamics for sequential decision making, traditionally through action-conditioned latent transitions (Ha & Schmidhuber, 2018; Hafner et al., 2019b;a). Recent latent-action approaches instead infer transition variables from temporal visual changes, enabling dynamics learning without predefined action labels (Bruce et al., 2024; Ye et al., 2025; Liang et al., 2025; Gao et al., 2026). World models have also been applied to embodied tasks such as navigation and tracking (Chen et al., 2025; Xie et al., 2026), but predominantly within individual tasks or control settings. In contrast, we investigate whether predictive dynamics can be shared across heterogeneous embodied tasks.

Unified Embodied Models. Recent generalist embodied models aim to accommodate diverse tasks and embodiments within shared architectures or control interfaces (Team et al., 2024; Doshi et al., 2024; O'Neill et al., 2024; Intelligence et al., 2025; Bjorck et al., 2025). In navigation, NavFoM jointly addresses multiple navigation and tracking tasks, while Qwen-RobotNav unifies navigation settings through a shared waypoint-prediction interface (Zhang et al., 2026b;d). Complementary to these efforts, WorldWeave focuses on sharing predictive dynamics while retaining task-specific adaptation to different decision requirements.

3 Our Approach

We present WorldWeave, a framework that jointly learns predictive world dynamics across Vision-and-Language Navigation (VLN) and Embodied Visual Tracking (EVT) while preserving their task-specific state representations and decision processes.

Problem Formulation. We index the two tasks by $\tau \in \{\mathrm{VLN}, \mathrm{EVT}\}$ and formulate each task as a partially observable Markov decision process (POMDP) $\mathcal{M}^\tau = (\mathcal{S}^\tau, \mathcal{A}^\tau, \mathcal{O}^\tau, \mathcal{T}^\tau, \Omega^\tau)$, where $\mathcal{S}^\tau$, $\mathcal{A}^\tau$, and $\mathcal{O}^\tau$ denote the state, action, and observation spaces, respectively, $\mathcal{T}^\tau$ and Ω^τ denote the transition and observation processes. At time step t, the agent receives an observation $\boldsymbol{o}_t^\tau \in \mathcal{O}^\tau$ and makes a decision based on its observation history $\boldsymbol{o}_{\le t}^\tau$ and natural-language instruction $\boldsymbol{l}^\tau$. The instruction provides navigation guidance in VLN and specifies the target to be followed in EVT. The corresponding policy is denoted by $\pi^\tau(\boldsymbol{a}_t^\tau \mid \boldsymbol{o}_{\le t}^\tau, \boldsymbol{l}^\tau)$, with $\boldsymbol{a}_t^\tau \in \mathcal{A}^\tau$.

A shared vision-language backbone (Bai et al., 2025a) encodes the observation history and instruction into contextual representations $\boldsymbol{h}_t^\tau = \mathrm{VLM}(\boldsymbol{o}_{\le t}^\tau, l^\tau)$. We derive the world state $\boldsymbol{s}_t^\tau = Q_\tau(\boldsymbol{h}_t^\tau)$ for dynamics modeling, where Q_τ employs learnable world-state queries to extract task-relevant state information from $\boldsymbol{h}_t^\tau$ via cross-attention. We denote $\boldsymbol{h}_t^\tau \in \mathbb{R}^{N_h^\tau \times d}$ and $\boldsymbol{s}_t^\tau \in \mathbb{R}^{N_s^\tau \times d}$. Although state representations may differ in number and semantics across tasks, they lie in a common feature space, allowing state evolution to be modeled through shared predictive dynamics.

Framework Overview. VLN and EVT exhibit complementary patterns of state evolution, but their different action semantics and parameterizations make conventional action-conditioned dynamics $p(\boldsymbol{s}_{t+1}^\tau \mid \boldsymbol{s}_t^\tau, \boldsymbol{a}_t^\tau)$ (Ha & Schmidhuber, 2018; Hafner et al., 2019b;a) difficult to share directly across tasks. Rather than aligning their heterogeneous action spaces, WorldWeave represents

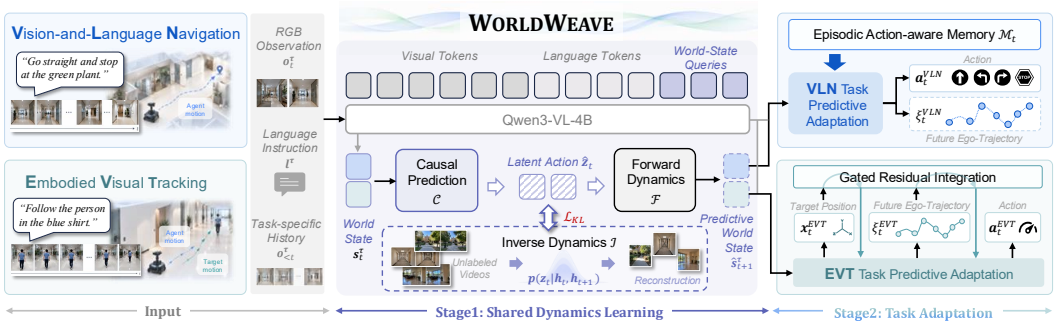

Figure 2: **Overview of WORLDWEAVE.** Shared predictive dynamics are learned from state changes in consecutive observations through inverse–forward dynamics and transferred across VLN and EVT without relying on task-specific action definitions. Task-specific adaptation then incorporates the predicted future states into each task's decision process.

state changes through a latent action $\boldsymbol{z}_t$, providing a shared transition space for modeling state evolution across tasks without directly conditioning on task-specific physical actions.

As illustrated in Fig. 2, we first learn the shared dynamics from diverse temporally ordered videos without requiring task-specific action annotations. Despite their heterogeneous motions, these videos share a common learning signal: how the represented world state changes over time. WORLDWEAVE encodes these changes into a shared latent transition space, allowing heterogeneous video transitions to jointly supervise the same transition model. Given consecutive visual representations, the model infers a latent action from their temporal change and learns how the represented state evolves under this latent action. Since future observations are unavailable during online decision making, we further learn a causal latent-action predictor that estimates $\boldsymbol{z}_t$ using only information available up to the current step. The resulting predictive dynamics can therefore anticipate future states without accessing future observations at decision time. We detail the learning of these shared dynamics in Sec. 3.1. The predicted future state is then incorporated differently into each task. For VLN, WORLDWEAVE combines the predicted state with historical context to support navigation, whereas for EVT, it supports structured estimation of target position, ego-trajectory, and continuous control. These task-specific integration mechanisms are described in Sec. 3.2, followed by the learning objectives and training procedure in Sec. 3.3.

3.1 LEARNING SHARED WORLD DYNAMICS

Building on the formulation above, we use $\boldsymbol{h}_t^\tau$ and $\boldsymbol{s}_t^\tau$ to denote the contextual visual representation and task-specific world-state representation at time t, respectively. While latent actions have been used to learn dynamics from videos without action annotations, we leverage them here to enable dynamics sharing across tasks with heterogeneous action spaces. Rather than requiring a unified state or action representation, WORLDWEAVE retains task-specific world states while sharing their transition modeling. To this end, we introduce a shared latent transition space $\mathcal{Z}$ and formulate predictive state evolution:

$$p(\boldsymbol{s}_{t+1}^\tau \mid \boldsymbol{s}_t^\tau, \boldsymbol{h}_{\le t}^\tau) = \int_{\mathcal{Z}} p\left(\boldsymbol{s}_{t+1}^\tau \mid \boldsymbol{s}_t^\tau, \boldsymbol{z}_t\right) p\left(\boldsymbol{z}_t \mid \boldsymbol{h}_{\le t}^\tau, \boldsymbol{s}_t^\tau\right) \, d\boldsymbol{z}_t. \tag{1}$$

Here, $\boldsymbol{z}_t \in \mathcal{Z}$ serves as a shared latent action that characterizes state changes without relying on task-specific physical action definitions. The transition model $\mathcal{F}$ is shared across tasks, while $\boldsymbol{s}_t^\tau$ retains task-specific state information; the causal predictor $\mathcal{C}$ estimates the latent action from the interaction history available up to the current step. Since the following dynamics-learning formulation applies identically to both tasks, we omit the task superscript τ for clarity.

Learning Future-Informed Latent Actions. During dynamics learning, consecutive observations provide direct evidence of the transition that occurred. We leverage this information through an inverse dynamics model $\mathcal{I}$, which takes the contextual representations before and after the transition to parameterize a future-informed posterior over the latent action: $\boldsymbol{z}_t \sim q\left(\boldsymbol{z}_t \mid \boldsymbol{h}_t, \boldsymbol{h}_{t+1}\right), \boldsymbol{z}_t \in \mathbb{R}^{N_z \times d_z}$. We refer to $\boldsymbol{z}_t$ as a *latent action*, which characterizes the observed transition between consecutive states. Here, q denotes the posterior over the latent action, parameterized by the inverse

dynamics model $\mathcal{I}$. We then use this latent action to model how the current state evolves into the subsequent state via the forward dynamics model $\mathcal{F}$, which predicts $\boldsymbol{s}_{t+1} \sim p_{\mathcal{F}}\left(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \boldsymbol{z}_t\right)$.

The inverse and forward dynamics therefore form a coupled transition-learning process: $\mathcal{I}$ identifies the transition from consecutive observations, while $\mathcal{F}$ models how that transition evolves the current state. Importantly, $\boldsymbol{z}_t$ is not required to recover the task-specific physical action executed between the two observations. Across tasks, these latent actions lie in a shared transition space $\mathcal{Z}$, allowing $\mathcal{F}$ to be shared without aligning the heterogeneous physical action spaces of VLN and EVT. Meanwhile, $\boldsymbol{s}_t^\tau$ remains task-specific, so sharing $\mathcal{F}$ does not require a unified state representation or assume identical state evolution across tasks. To learn this coupled dynamics, we jointly optimize the inverse and forward models via a state prediction objective:

$$\mathcal{L}_{\text{state}} = -\mathbb{E}_{\boldsymbol{z}_t \sim q(\boldsymbol{z}_t \mid \boldsymbol{h}_t, \boldsymbol{h}_{t+1})}\left[\log p_{\mathcal{F}}\left(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \boldsymbol{z}_t\right)\right]. \tag{2}$$

This objective encourages the inferred latent action to capture the observed state change, so that $\boldsymbol{z}_t$ encodes transition information directly useful for predicting $\boldsymbol{s}_{t+1}$.

Causal Latent-Action Prediction. The inverse dynamics model $\mathcal{I}$ infers latent action from consecutive observations, providing the latent transition used to learn the forward dynamics. However, this inference requires $\boldsymbol{h}_{t+1} = \text{VLM}(\boldsymbol{o}_{\leq t+1}, \boldsymbol{l})$, which is unavailable during inference. To enable prediction without future observations, we learn a causal prior over the latent action, parameterized by $\mathcal{C}$, using only the observed history and current world state: $\hat{\boldsymbol{z}}_t \sim p\left(\hat{\boldsymbol{z}}_t \mid \boldsymbol{h}_{\leq t}, \boldsymbol{s}_t\right)$. Here, $\hat{\boldsymbol{z}}_t \in \mathbb{R}^{N_z \times d_z}$ denotes a causal latent-action sample obtained without future observations. We then align the causal prior with the posterior via KL divergence (Kullback & Leibler, 1951):

$$\mathcal{L}_{\text{KL}} = D_{\text{KL}}\left(q\left(\boldsymbol{z}_t \mid \boldsymbol{h}_t, \boldsymbol{h}_{t+1}\right) \parallel p\left(\boldsymbol{z}_t \mid \boldsymbol{h}_{\leq t}, \boldsymbol{s}_t\right)\right), \tag{3}$$

which transfers the transition information from the posterior to the causal prior, allowing the latent transition to be predicted without access to future observations. Note that both distributions are defined over the random variable $\boldsymbol{z}_t$, while the hat notation $\hat{\boldsymbol{z}}_t$ is used specifically to distinguish the causal latent-action sample from the future-informed one.

The KL objective constrains the causal prediction in latent-action space, but does not directly supervise the resulting state prediction. We therefore feed $\hat{\boldsymbol{z}}_t$ into the shared forward dynamics model $\mathcal{F}$, yielding $\hat{\boldsymbol{s}}_{t+1} \sim p_{\mathcal{F}}\left(\hat{\boldsymbol{s}}_{t+1} \mid \boldsymbol{s}_t, \hat{\boldsymbol{z}}_t\right)$, a causal estimate of the future state usable at decision time. Since $\boldsymbol{s}_{t+1}$ is a low-dimensional projection of $\boldsymbol{h}_{t+1}$, state-space supervision alone may be insufficient. To address this, we introduce a decoder $d(\cdot)$ that reconstructs $\boldsymbol{h}_{t+1}$ from $\hat{\boldsymbol{s}}_{t+1}$:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{\hat{\boldsymbol{z}}_t \sim p\left(\boldsymbol{z}_t \mid \boldsymbol{h}_{\leq t}, \boldsymbol{s}_t\right)}\left[\left\|d\left(\hat{\boldsymbol{s}}_{t+1}\right) - \boldsymbol{h}_{t+1}\right\|_2^2\right]. \tag{4}$$

Together, these objectives supervise the dynamics learning at complementary levels: $\mathcal{L}_{\text{state}}$ learns the forward dynamics, $\mathcal{L}_{\text{KL}}$ aligns the causal prior with the posterior, and $\mathcal{L}_{\text{rec}}$ further constrains the resulting future-state prediction through the future observation representation. Then the overall dynamics-learning objective is:

$$\mathcal{L}_{\text{Dyn}} = \mathcal{L}_{\text{state}} + \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{KL}}, \text{ where } \lambda \text{ balances the KL term.} \tag{5}$$

In practice, $\mathcal{I}$, $\mathcal{C}$, and $\mathcal{F}$ are implemented as Transformer-based recurrent modules. Both $\mathcal{I}$ and $\mathcal{C}$ model probabilistic latent transitions as the future-informed posterior and causal prior, respectively, and are aligned via $\mathcal{L}_{\text{KL}}$. At inference, $\mathcal{I}$ and the decoder $d(\cdot)$ are discarded; only $\mathcal{C}$ and $\mathcal{F}$ are retained to predict $\hat{\boldsymbol{s}}_{t+1}$ without future observations. Implementation details appear in Appendix B.1.

3.2 Adapting Shared Dynamics Across Tasks

While the shared world dynamics predict how the current state may evolve, VLN and EVT retain distinct task representations and decision processes. We thus use a common context construction to incorporate the shared prediction, with task-specific readouts for adaptation. Specifically, we construct a decision context $\boldsymbol{c}_t^\tau$ by concatenating the current observation representation, world state, and predicted future state. For each task, a learnable query representation $\boldsymbol{q}_t^\tau$ is extracted from the VLM features and used to retrieve relevant information from this predictive context for downstream decision making. The task representation $\boldsymbol{q}_t^\tau$ then selectively retrieves relevant information from this context via a task-specific readout:

$$\boldsymbol{c}_t^\tau = \text{Concat}\left(\boldsymbol{h}_t^\tau,\, \boldsymbol{s}_t^\tau,\, \hat{\boldsymbol{s}}_{t+1}^\tau\right), \quad \boldsymbol{q}_t^{\tau,*} = \boldsymbol{q}_t^\tau + \mathcal{R}_\tau\left(\boldsymbol{q}_t^\tau,\, \boldsymbol{c}_t^\tau\right), \qquad \tau \in \{\text{VLN}, \text{EVT}\}. \tag{6}$$

Here, $\mathcal{R}_\tau$ denotes a task-specific adaptation function, which we instantiate with cross-attention between $\boldsymbol{q}_t^\tau$ and $\boldsymbol{c}_t^\tau$. The shared context exposes both tasks to the predicted state evolution, while the residual connection preserves task-specific information. From $\boldsymbol{q}_t^{\tau,*}$, each task branch further incorporates the signals required for its own decision making.

For VLN, $\boldsymbol{q}_t^{\mathrm{VLN},*}$ incorporates the state evolution predicted by the shared world dynamics into the instruction-conditioned task representation. As navigation unfolds over an extended instruction-guided trajectory, the current decision can further benefit from recent navigation experience. We therefore augment the predictive representation with an action-aware episodic memory $\mathcal{M}_t = \{\boldsymbol{m}_i\}_{i<t}$, where each entry $\boldsymbol{m}_i$ encodes the previous task representation $\boldsymbol{q}_i^{\mathrm{VLN}}$, the executed action $\boldsymbol{a}_i^{\mathrm{VLN}}$, and its temporal distance to the current step. By associating previous task representations with the actions executed from them, the memory provides historical state–action context for the current navigation decision rather than merely caching past observations.

While the episodic memory complements the predictive representation with past experience, action supervision alone focuses on the immediate decision and offers limited constraint on how the agent's motion evolves over subsequent steps. We therefore introduce an auxiliary future-motion objective that predicts the agent's K-step ego-trajectory, providing explicit multi-step supervision that encourages the predictive representation to retain information beyond that required for the current action. Accordingly, the VLN predictions are formulated as

$$p\big(\boldsymbol{a}_t^{\mathrm{VLN}}, \boldsymbol{\xi}_t^{\mathrm{VLN}} \,|\, \boldsymbol{q}_t^{\mathrm{VLN},*}, \mathcal{M}_t\big) = p\big(\boldsymbol{a}_t^{\mathrm{VLN}} \,|\, \boldsymbol{q}_t^{\mathrm{VLN},*}, \mathcal{M}_t\big)\, p\big(\boldsymbol{\xi}_t^{\mathrm{VLN}} \,|\, \boldsymbol{q}_t^{\mathrm{VLN},*}\big), \tag{7}$$

where $\boldsymbol{a}_t^{\mathrm{VLN}} = (u_t, r_t)$, with $u_t \in \{\textsc{Stop}, \textsc{Forward}, \textsc{Left}, \textsc{Right}\}$ denoting the action type and r_t its magnitude—movement distance for FORWARD and turning angle for LEFT / RIGHT. $\boldsymbol{\xi}_t^{\mathrm{VLN}} \in \mathbb{R}^{K\times 3}$ denotes the K-step future ego-trajectory predicted at time t, with each ego-pose represented by two-dimensional planar position and heading in the current agent-centric coordinate frame. The two factors serve distinct roles: episodic memory directly informs current action prediction, whereas trajectory prediction provides task-specific multi-step motion supervision. Notably, the trajectory does not condition the navigation action and serves only as an auxiliary training objective.

For EVT, translating the predicted future state into a tracking action requires reasoning about its implications for agent–target interaction. Unlike VLN, which benefits from episodic memory to retain navigation experience along an extended instruction-guided trajectory, EVT primarily requires responsive control with respect to a continuously moving target. Recent motion cues are already captured by its consecutive visual observations; we therefore do not introduce additional episodic memory, but instead focus the task adaptation on the current agent–target relation and anticipated ego-motion. Specifically, whereas VLN uses future ego-trajectory prediction as auxiliary supervision, EVT models a progressive dependency among target localization, future ego-motion, and action prediction. We formulate their joint predictive distribution as:

$$\begin{aligned} &p\left(\boldsymbol{a}_t^{\mathrm{EVT}}, \boldsymbol{\xi}_t^{\mathrm{EVT}}, \boldsymbol{x}_t^{\mathrm{EVT}} \,|\, \boldsymbol{q}_t^{\mathrm{EVT},*}\right) \\ &= p\big(\boldsymbol{x}_t^{\mathrm{EVT}} \,|\, \boldsymbol{q}_t^{\mathrm{EVT},*}\big)\; p\big(\boldsymbol{\xi}_t^{\mathrm{EVT}} \,|\, \boldsymbol{x}_t^{\mathrm{EVT}}, \boldsymbol{q}_t^{\mathrm{EVT},*}\big)\, p\big(\boldsymbol{a}_t^{\mathrm{EVT}} \,|\, \boldsymbol{\xi}_t^{\mathrm{EVT}}, \boldsymbol{q}_t^{\mathrm{EVT},*}\big), \end{aligned} \tag{8}$$

where $\boldsymbol{x}_t^{\mathrm{EVT}} \in \mathbb{R}^2$ denotes the target position in the agent-centric frame, $\boldsymbol{\xi}_t^{\mathrm{EVT}} \in \mathbb{R}^{K\times 2}$ the predicted K-step ego-trajectory (each entry a planar position), and $\boldsymbol{a}_t^{\mathrm{EVT}} = (v_t, \omega_t) \in \mathbb{R}^2$ the linear and angular velocities. We implement these dependencies through gated residual integration, where each intermediate prediction is selectively incorporated into the representation for the subsequent stage. Detailed formulations are provided in Appendix B.2.

3.3 TWO-STAGE JOINT LEARNING

We train WORLDWEAVE in two stages: shared dynamics learning and task adaptation. In the first stage, we learn shared predictive dynamics from heterogeneous real-world and simulated videos without requiring physical-action annotations, enabling the model to capture state-transition patterns across diverse embodied interactions. In the second stage, the learned dynamics are transferred to VLN and EVT through lightweight task-specific adaptation using embodied task data. This two-stage design separates general dynamics learning from downstream task adaptation, allowing the shared dynamics to be reused while preserving the distinct state representations and decision processes of each task. Details of data composition and training configurations are given in Appendix A.

Stage I: Shared Dynamics Learning. We first learn the shared world dynamics introduced in Sec. 3.1 from heterogeneous real-world and simulated videos, without requiring physical-action annotations.This stage jointly adapts the pretrained vision-language backbone (Bai et al., 2025a) and learns the dynamics modules using $\mathcal{L}_{\mathrm{Dyn}}$ in Eq.(5), which optimizes state prediction, representation reconstruction, and latent-action alignment. During training, latent actions are sampled via reparameterization from the posterior and causal prior distributions, while the KL divergence aligns the causal prior with the posterior. The posterior latent action is used to learn the shared forward dynamics $\mathcal{F}$, while the causal latent action is used to reconstruct the future observation representation.

After Stage I, the inverse dynamics model $\mathcal{I}$ and representation decoder d are discarded. The causal latent-action predictor $\mathcal{C}$ and shared forward dynamics model $\mathcal{F}$ are retained to predict future states using only information available at the current step.

Stage II: Task Adaptation. We incorporate the pretrained world dynamics into VLN and EVT through the task-specific adaptation described in Sec. 3.2. The dynamics modules are frozen, while the vision-language backbone is adapted via separate LoRA adapters for each task, along with their respective action heads. Although both tasks share the same predictive dynamics, their learning objectives reflect their distinct decision processes.

For VLN, we jointly supervise the navigation action $\boldsymbol{a}_t^{\mathrm{VLN}}$ and the future ego-trajectory $\boldsymbol{\xi}_t^{\mathrm{VLN}}$:

$$\mathcal{L}_{\mathrm{VLN}} = \mathbb{E}\big[-\log p\big(\boldsymbol{a}_t^{\mathrm{VLN,gt}} \mid \boldsymbol{q}_t^{\mathrm{VLN},*}, \mathcal{M}_t\big) + \ell\big(\boldsymbol{\xi}_t^{\mathrm{VLN}}, \boldsymbol{\xi}_t^{\mathrm{VLN,gt}}\big)\big], \tag{9}$$

where the superscript gt denotes ground-truth supervision. We use ℓ_1 loss for trajectory regression $\ell(\cdot,\cdot)$ and categorical cross-entropy for both action type and motion parameters. The trajectory loss is computed over valid future steps and serves only as auxiliary supervision.

For EVT, the task adaptation successively predicts the agent-centric target position $\boldsymbol{x}_t^{\mathrm{EVT}}$, future ego-trajectory $\boldsymbol{\xi}_t^{\mathrm{EVT}}$, and action $\boldsymbol{a}_t^{\mathrm{EVT}} = (v_t, \omega_t)$. We jointly supervise these predictions as

$$\mathcal{L}_{\mathrm{EVT}} = \mathbb{E}\big[\ell\big(\boldsymbol{x}_t^{\mathrm{EVT}}, \boldsymbol{x}_t^{\mathrm{EVT,gt}}\big) + \ell\big(\boldsymbol{\xi}_t^{\mathrm{EVT}}, \boldsymbol{\xi}_t^{\mathrm{EVT,gt}}\big) + \ell\big(\boldsymbol{a}_t^{\mathrm{EVT}}, \boldsymbol{a}_t^{\mathrm{EVT,gt}}\big)\big]. \tag{10}$$

4 EXPERIMENTS

We evaluate WORLDWEAVE on VLN and EVT through comparisons with existing methods, ablations of cross-task dynamics sharing and task-specific adaptation, and real-world robot deployment. **More details, experiments, real-world evaluation, and further analyses are given in Appendix.**

Datasets. We evaluate on R2R-CE (Anderson et al., 2018b; Krantz et al., 2020a), RxR-CE (Ku et al., 2020), VLN-PE (Wang et al., 2025a), and VLNVerse (Lin et al., 2025) for VLN, and EVT-Bench (Wang et al., 2025c), UnrealZoo (Zhong et al., 2025), and Gym-UnrealCV (Qiu et al., 2017) for EVT. Training and preprocessing details are provided in Appendix A.

Evaluation Metrics. For VLN, we report Success Rate (SR), Success weighted by Path Length (SPL), Navigation Error (NE), Normalized Dynamic Time Warping (NDTW), and Success weighted by NDTW (SDTW) (Anderson et al., 2018a). For EVT, we report Success Rate (SR), Tracking Rate (TR), and Collision Rate (CR). Detailed definitions are provided in Appendix B.3.

Table 1: Cross-task performance overview.

	R2R-CE		RxR-CE		EVTBench-STT	
Method	**SR↑**	**SPL↑**	**SR↑**	**SPL↑**	**SR↑**	**TR↑**
Baseline (VLN)	46.9	41.9	45.8	38.1	–	–
Baseline (EVT)	–	–	–	–	41.1	71.2
Uni-NaVid	47.0	42.7	48.7	40.9	25.7	39.5
WORLDWEAVE	**54.2**	**49.4**	**52.7**	**45.9**	**78.1**	**88.0**

Cross-Task Overview. Before examining each task individually, we evaluate WORLDWEAVE from a cross-task perspective. We report the task-specific baselines trained independently on VLN and EVT, and include Uni-NaVid (Zhang et al., 2025) as a publicly available generalist reference covering both instruction-guided navigation and human following. Other recent generalist models spanning navigation and target tracking do not provide comparable public releases; their reported results are therefore included in the task-specific comparisons below. As shown in Table 1, WORLDWEAVE improves over the corresponding task-specific baselines on both VLN and EVT, while also outperforming Uni-NaVid across all evaluated settings. These results provide an overall view of cross-task performance before we examine each benchmark in detail.

Table 2: Comparison on R2R-CE and RxR-CE Val-Unseen splits under monocular settings. Results outperforming the baseline are **bold**; best results among publicly available methods are underlined. *Gray rows indicate recent VLM-based methods whose code or training data are not publicly released; we include their officially reported results for reference only.*

Methods	R2R-CE Val-Unseen					RxR-CE Val-Unseen					
	TL ↓	NE ↓	OSR ↑	SR ↑	SPL ↑	TL ↓	NE ↓	OSR ↑	SR ↑	SPL ↑	NDTW ↑
monoVLN (Lu et al., 2025)	-	4.61	62.4	54.8	44.4	-	8.29	37.7	31.8	26.8	-
SC^2-WM (Yao et al., 2026)	17.65	5.37	58.8	50.9	37.2	20.87	8.36	48.4	35.8	27.2	44.9
UniNaVid (Vicuna-7B) (Zhang et al., 2025)	-	5.58	53.3	47.0	42.7	-	6.24	-	48.7	40.9	-
NaVILA (VILA-1.5-3B) (Cheng et al., 2024)	-	5.43	62.5	54.0	49.0	-	6.77	-	49.3	44.0	58.8
StreamVLN (Qwen2-7B) (Wei et al., 2025b)	-	4.98	64.2	56.9	51.9	-	6.22	-	52.9	46.0	61.9
InternVLA-N1 (Qwen2.5-7B) (Team, 2025)	-	4.83	63.3	58.2	54.0	-	5.91	-	53.5	46.1	65.3
QwenRobotNav (Qwen3-VL-4B) (Zhang et al., 2026d)	-	4.22	73.6	66.9	60.5	-	4.15	-	71.3	61.5	68.6
Baseline (Qwen3-VL-4B) (Bai et al., 2025b)	9.00	6.20	52.1	46.9	41.9	11.29	7.24	41.2	45.8	38.1	55.2
WORLDWEAVE (Qwen3-VL-4B)	11.3	**5.78**	**61.5**	**54.2**	**49.4**	13.89	**6.31**	**50.1**	**52.7**	**45.9**	**62.9**

Table 3: Comparison on EVT-Bench. Results outperforming the baseline are **bold**.

Methods	*Single-Target Tracking (STT)*			*Distracted Tracking (DT)*			*Ambiguity Tracking (AT)*		
	SR↑	TR↑	CR↓	SR↑	TR↑	CR↓	SR↑	TR↑	CR↓
EVT (Zhong et al., 2024)	32.5	49.9	40.5	15.7	35.7	53.3	18.3	21.0	44.9
Uni-NaVid (Vicuna-7B) (Zhang et al., 2025)	25.7	39.5	41.9	11.3	27.4	43.5	8.26	28.6	43.7
OpenTrackVLA (Qwen-0.6B) (Lee & Zhao, 2025)	47.2	58.7	5.72	22.5	42.8	8.93	28.6	51.8	6.37
Qwen-RobotNav (Qwen3-VL-4B) (Zhang et al., 2026d)	77.4	90.0	6.40	-	-	-	-	-	-
NavFoM(Qwen2-7B) (Zhang et al., 2026c)	85.0	80.5	-	61.4	68.2	-	-	-	-
TrackVLA++ (Qwen2-7B)(Liu et al. (2025))	86.0	81.0	2.10	66.5	68.8	4.71	51.2	63.4	15.9
Baseline (Qwen3-VL-4B)(Bai et al., 2025b)	41.1	71.2	6.50	16.3	45.7	19.4	14.8	47.1	8.70
WORLDWEAVE (Qwen3-VL-4B)	**78.1**	**88.0**	**4.89**	**41.7**	**75.2**	**14.24**	**51.2**	**74.5**	18.62

Note: Evaluation metrics are defined as follows: **Success Rate (SR)**, proportion of episodes that the agent ends correctly oriented within 1–3m of the target; **Tracking Rate (TR)**, proportion of timesteps with successful target tracking; and **Collision Rate (CR)**, proportion of episodes terminated due to collisions.

4.1 COMPARISON WITH STATE-OF-THE-ART METHODS

Vision-and-Language Navigation. Table 2 compares WORLDWEAVE with existing methods on R2R-CE and RxR-CE under monocular settings. On R2R-CE, WORLDWEAVE improves the matched Qwen3-VL baseline from 46.9 to 54.2 in SR and from 41.9 to 49.4 in SPL, while reducing NE from 6.20 to 5.78. The gains transfer to the more instruction-diverse RxR-CE, where SR / SPL increase from 45.8 / 38.1 to 52.7 / 45.9, with improved NDTW further indicating better trajectory-level navigation quality. These improvements are consistent with our VLN-specific adaptation, where action-aware episodic memory retains navigation experience and future ego-trajectory prediction provides multi-step motion supervision. While some recent VLM-based methods achieve higher absolute performance, WORLDWEAVE consistently improves the matched task-specific baseline and remains competitive with publicly available VLN methods. Consistent gains on VLN-PE and VLNVerse are further reported in the Appendix C.1.

Embodied Visual Tracking. On EVTBench, WORLDWEAVE substantially improves over the matched Qwen3-VL baseline in SR and TR across STT, DT, and AT (Table 3) TR rises from 71.2 / 45.7 / 47.1 to 88.0 / 75.2 / 74.5, with larger gains under the more challenging DT and AT settings involving distractors or target ambiguity. These improvements align with our EVT-specific adaptation, which progressively integrates predicted target position and ego-motion to tailor shared predictions to agent–target interactions. CR decreases under STT and DT yet increases under AT, suggesting that improved target retention does not always reduce collisions when ambiguity is present. While some recent methods shown in gray achieve higher absolute performance, WORLDWEAVE consistently improves the matched baseline and remains competitive among publicly available methods. Additional evaluations on UnrealZoo and Gym-UnrealCV further show that tracking gains extend to changing targets and zero-shot conditions (please refer to Appendix C.2).

4.2 ABLATION STUDIES

What Should Be Shared Across Tasks? A central design choice in WORLDWEAVE is to share predictive dynamics while preserving task-specific representations and adaptation. Beyond independently trained VLN and EVT baselines, we compare four joint-training variants with different sharing boundaries: **(a)** a shared LoRA adapter, **(b)** task-specific adapters, **(c)** (b) augmented with the shared causal latent-action predictor $\mathcal{C}$ and forward dynamics model $\mathcal{F}$, and **(d) WORLDWEAVE**, which further introduces task-specific predictive adaptation $\mathcal{R}_\tau$. As shown in Table 4, (a) favors VLN but severely degrades EVT (*e.g.*, SR/CR to 29.1/25.1), while (b) restores EVT at the

Table 4: Comparison of cross-task sharing strategies on R2R Val-Unseen and EVTBench STT sets. Variants (a)–(d) progressively examine the sharing of state representations, predictive dynamics, and task-specific predictive adaptation. Best results are in **bold**.

	Model Design			R2R Val-Unseen			EVTBench STT		
Model Design	**State Representation**	**Dynamics**	**Predictive Adaptation**	**OSR ↑**	**SR ↑**	**SPL ↑**	**SR ↑**	**TR ↑**	**CR ↓**
Single-Task Training									
Baseline (VLN Only)	Task-specific	–	–	52.1	46.9	41.9	–	–	–
Baseline (EVT Only)	Task-specific	–	–	–	–	–	41.1	71.2	6.50
Cross-Task Joint Training									
(a)	Shared	–	–	57.7	50.2	46.3	29.1	58.6	25.1
(b)	Task-specific	–	–	46.3	41.8	39.1	68.0	77.1	**3.41**
(c)	Task-specific	Shared $\mathcal{C}, \mathcal{F}$	Shared	55.2	48.5	44.3	71.2	72.7	4.82
(d) WORLDWEAVE	Task-specific	Shared $\mathcal{C}, \mathcal{F}$	Task-specific $\mathcal{R}_\tau$	**61.5**	**54.2**	**49.4**	**78.1**	**88.0**	4.89

cost of VLN performance. Thus, changing representation sharing alone merely shifts the cross-task trade-off. In contrast, adding shared $\mathcal{C}$ and $\mathcal{F}$ in (c), while keeping task-specific representations, improves both tasks, showing that predictive dynamics can be productively shared. Finally, (d) further improves upon (c) with task-specific $\mathcal{R}_\tau$, indicating that each task benefits from adapting shared predictions to its own decision process. These results support the design of WORLDWEAVE: shared predictive dynamics, and task-specific predictive adaptation.

4.3 DISCUSSION

Recent methods in the gray rows achieve higher absolute performance on some metrics, but at substantially larger training scales (∼8M–15.6M samples versus 500 hours of video and 2.5M task samples). We therefore focus on the controlled question of *what should be shared across heterogeneous embodied tasks*. Our results suggest that sharing representations induces cross-task trade-offs, whereas sharing predictive dynamics with task-specific adaptation improves both VLN and EVT over their single-task baselines. This provides evidence that predictive dynamics can be shared without requiring unified task representations or decision processes. More importantly, the design naturally lends itself to scaling: Stage-I can exploit heterogeneous videos without unified action annotations, while Stage-II accommodates task-specific adaptation. Due to computational constraints, we leave larger-scale training to future work, which will investigate whether such scaling further improves transferability, particularly to distinct embodied interactions such as manipulation.

4.4 REAL-WORLD ROBOT EXPERIMENTS

We further deploy WORLDWEAVE on a TITA wheel-legged robot for both VLN and EVT in previously unseen indoor environments. As shown in Fig. 3, the robot performs closed-loop navigation and tracking under progressively challenging instructions and target-motion patterns, achieves 66.7% and 66.0% SR on VLN and EVT, respectively, compared with 46.0% and 48.7% for the matched baselines. Detailed task protocols, and results are provided in Appendix D.

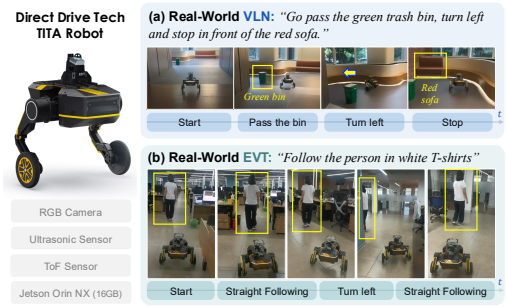

Figure 3: Real-world deployment on the TITA robot for VLN and EVT tasks.

5 CONCLUSION

We introduced WORLDWEAVE to investigate shared predictive dynamics across heterogeneous embodied tasks, using VLN and EVT as complementary testbeds. By coupling shared predictive dynamics with task-specific adaptation, our approach achieves consistent improvements over single-task baselines on both tasks. These results support shared predictive dynamics with task-specific adaptation as a promising paradigm for cross-task embodied learning. As embodied models scale to diverse tasks, key questions arise: which components can be shared, and how should shared predictions be adapted to different interaction patterns? Our study suggests that predictive world dynamics provide a reusable foundation—additional tasks can leverage shared dynamics while introducing task-specific adaptation modules. While our current study focuses on mobile embodied tasks, extending to broader scenarios such as manipulation remains an important future direction. Upon acceptance, we will release our code and models to facilitate future research.

AI USE STATEMENT

In this work, we used generative AI tools for polishing and improving the clarity of writing. We have not used generative AI tools for research ideation, experimental design, generating datasets, or proving mathematical claims. We have reviewed all AI-assisted text and take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide detailed information to facilitate the reproduction of WORLDWEAVE. The model architecture and learning objectives are described in Sec. 3, including shared dynamics learning, task-specific adaptation, and the two-stage training procedure. The Appendix provides complete details of the datasets and preprocessing procedures, evaluation metrics and protocols, model implementation and training hyperparameters, as well as the setup and evaluation protocol for the real-world robot experiments. We will also release the training and evaluation code, model configurations, and data-processing scripts to support reproduction of the reported results.

REFERENCES

Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Topo-metric map pre-training for language-guided navigation. *arXiv preprint arXiv:2212.04385*, 2022a.

Dong An, Zun Wang, Yangguang Li, Yi Wang, Yicong Hong, Yan Huang, Liang Wang, and Jing Shao. 1st place solutions for rxr-habitat vision-and-language navigation competition. *arXiv preprint arXiv:2206.11610*, 2022b.

Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE TPAMI*, 2024.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pp. 3674–3683. IEEE, 2018a.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pp. 3674–3683. IEEE, 2018b.

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *CVPR*, pp. 19107–19117. IEEE, 2022.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025b.

Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, pp. 6182–6191, 2019.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *ICML*, 2024.

Jiaqi Chen, Bingqian Lin, Xinmin Liu, Xiaodan Liang, and Kwan-Yee K Wong. Affordances-oriented planning using foundation models for continuous vision-language navigation. *arXiv preprint arXiv:2407.05890*, 2024a.

Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee Wong. Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9796–9810, 2024b.

Jintao Chen, Junjun Hu, Haochen Bai, Minghua Luo, Xinda Xue, Botao Ren, Chengyu Bai, Shichao Xie, Ziyi Chen, Fei Liu, et al. Astranav-world: World model for foresight control and consistency. *arXiv preprint arXiv:2512.21714*, 2025.

Peihao Chen, Dongyu Ji, Kun-Li Channing Lin, Runhao Zeng, Thomas H. Li, Mingkui Tan, and Chuang Gan. Weakly-supervised multi-granularity map learning for vision-and-language navigation. In *NeurIPS*, pp. 38149–38161, 2022a.

Peihao Chen, Xinyu Sun, Hongyan Zhi, Runhao Zeng, Thomas H Li, Gaowen Liu, Mingkui Tan, and Chuang Gan. A^2 nav: Action-aware zero-shot robot navigation by exploiting vision-and-language ability of foundation models. *arXiv preprint arXiv:2308.07997*, 2023.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *NeurIPS*, 34:5834–5847, 2021.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *CVPR*, pp. 16516–16526. IEEE, 2022b.

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. *arXiv preprint arXiv:2412.04453*, 2024.

Zedong Chu, Shichao Xie, Xiaolong Wu, Yanfen Shen, Minghua Luo, Zhengbo Wang, Fei Liu, Xiaoxu Leng, Junjun Hu, Mingyang Yin, et al. Abot-n0: Technical report on the vla foundation model for versatile embodied navigation. *arXiv preprint arXiv:2602.11598*, 2026.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Guangzhao Dai, Jian Zhao, Yuantao Chen, Yusen Qin, Hao Zhao, Guosen Xie, Yazhou Yao, Xiangbo Shu, and Xuelong Li. Unitedvln: Generalizable gaussian splatting for continuous vision-language navigation. *arXiv preprint arXiv:2411.16053*, 2024.

Ria Doshi, Homer Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. *arXiv preprint arXiv:2408.11812*, 2024.

Junyu Gao, Xuan Yao, and Changsheng Xu. Fast-slow test-time adaptation for online vision-and-language navigation. In *ICML*, pp. 14902–14919, 2024.

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.

Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In *CVPR*, pp. 15439–15449, 2022.

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 *CVPR*, pp. 18995–19012, 2022.

Meenakshi Gupta, Swagat Kumar, Laxmidhar Behera, and Venkatesh K Subramanian. A novel vision-based tracking algorithm for a human-following mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(7):1415–1427, 2016.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *ICML*, pp. 2555–2565. PMLR, 2019b.

Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *CVPR*, pp. 15418–15428, 2022.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

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

Wooseong Jeong and Kuk-Jin Yoon. Quantifying task priority for multi-task optimization. In *CVPR*, pp. 363–372, 2024.

Wooseong Jeong and Kuk-Jin Yoon. Selective task group updates for multi-task optimization. In *ICLR*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Jacob Krantz and Stefan Lee. Sim-2-sim transfer for vision-and-language navigation in continuous environments. In *ECCV*, pp. 588–603, 2022.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *ECCV*, pp. 104–120. Springer, 2020a.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *ECCV*, pp. 104–120. Springer, 2020b.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4392–4412, 2020.

Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

Duy Tho Le, Chenhui Gou, Stavya Datta, Hengcan Shi, Ian Reid, Jianfei Cai, and Hamid Rezatofighi. Jrdb-panotrack: An open-world panoptic segmentation and tracking robotic dataset in crowded human environments. In *CVPR*, pp. 22325–22334. IEEE, 2024.

Kyusong Lee and Tiancheng Zhao. Opentrackvla: Open-source visual language action model for visual navigation and following. https://github.com/om-ai-lab/OpenTrackVLA, 2025.

Anthony Liang, Pavel Czempin, Matthew M Hong, Yutai Zhou, Jingzhen Wang, Erdem Biyik, and Stephen Tu. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025.

Kunyang Lin, Peihao Chen, Diwei Huang, Thomas H Li, Mingkui Tan, and Chuang Gan. Learning vision-and-language navigation from youtube videos. In *ICCV*, pp. 8283–8292. IEEE, 2023.

Sihao Lin, Zerui Li, Xunyi Zhao, Gengze Zhou, Liuyi Wang, Rong Wei, Rui Tang, Juncheng Li, Hanqing Wang, Jiangmiao Pang, et al. Vlnverse: A benchmark for vision-language navigation with versatile, embodied, realistic simulation and evaluation. *arXiv preprint arXiv:2512.19021*, 2025.

Jiahang Liu, Yunpeng Qi, Jiazhao Zhang, Minghan Li, Shaoan Wang, Kui Wu, Hanjing Ye, Hong Zhang, Zhibo Chen, Fangwei Zhong, et al. Trackvla++: Unleashing reasoning and memory capabilities in vla models for embodied visual tracking. *arXiv preprint arXiv:2510.07134*, 2025.

Rui Liu, Wenguan Wang, and Yi Yang. Volumetric environment representation for vision-language navigation. In *CVPR*, pp. 16317–16328. IEEE, 2024.

Youzhi Liu, Li Gao, Liu Liu, Mingyang Lv, and Yang Cai. Comatrack: Competitive multi-agent game-theoretic tracking with vision-language-action models. *arXiv preprint arXiv:2603.22846*, 2026.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024.

Renjie Lu, Yu Zhou, Hao Cheng, Jingke Meng, and Wei-Shi Zheng. monovln: Bridging the observation gap between monocular and panoramic vision and language navigation. In *ICCV*, pp. 9477–9486, October 2025.

Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking via reinforcement learning. In *ICML*, pp. 3286–3295. PMLR, 2018.

Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking and its real-world deployment via reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1317–1332, 2019.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *ICLR*, volume 2024, pp. 15306–15336, 2024.

Weichao Qiu, Fangwei Zhong, Yi Zhang, Siyuan Qiao, Zihao Xiao, Tae Soo Kim, and Yizhou Wang. Unrealcv: Virtual worlds for computer vision. In *ACM MM*, pp. 1221–1224, 2017.

Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel X. Chang. Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. In *EMNLP*, pp. 4018–4028, 2021.

Xiangyu Shi, Zerui Li, Wenqi Lyu, Jiatong Xia, Feras Dayoub, Yanyuan Qiao, and Qi Wu. Smartway: Enhanced waypoint prediction and backtracking for zero-shot vision-and-language navigation. *arXiv preprint arXiv:2503.10069*, 2025.

IN Team. Internvla-n1: An open dual-system vision-language navigation foundation model with learned latent plans. Technical report, Technical report, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Hanqing Wang, Wei Liang, Luc Van Gool, and Wenguan Wang. Dreamwalker: Mental planning for continuous vision-language navigation. In *ICCV*, pp. 10873–10883, 2023a.

Liuyi Wang, Xinyuan Xia, Hui Zhao, Hanqing Wang, Tai Wang, Yilun Chen, Chengju Liu, Qijun Chen, and Jiangmiao Pang. Rethinking the embodied gap in vision-and-language navigation: A holistic study of physical and visual disparities. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9455–9465. IEEE, 2025a.

Liuyi Wang, Xinyuan Xia, Hui Zhao, Hanqing Wang, Tai Wang, Yilun Chen, Chengju Liu, Qijun Chen, and Jiangmiao Pang. Rethinking the embodied gap in vision-and-language navigation: A holistic study of physical and visual disparities. In *ICCV*, pp. 9455–9465. IEEE, 2025b.

Shaoan Wang, Jiazhao Zhang, Minghan Li, Jiahang Liu, Anqi Li, Kui Wu, Fangwei Zhong, Junzhi Yu, Zhizheng Zhang, and He Wang. Trackvla: Embodied visual tracking in the wild. *arXiv preprint arXiv:2505.23189*, 2025c.

Shaoan Wang, Yuanfei Luo, Xingyu Chen, Aocheng Luo, Dongyue Li, Chang Liu, Sheng Chen, Yangang Zhang, and Junzhi Yu. Vlingnav: Embodied navigation with adaptive reasoning and visual-assisted linguistic memory. *arXiv preprint arXiv:2601.08665*, 2026.

Shuo Wang, Yongcai Wang, Wanting Li, Xudong Cai, Yucheng Wang, Maiyue Chen, Kaihui Wang, Zhizhong Su, Deying Li, and Zhaoxin Fan. Aux-think: Exploring reasoning strategies for data-efficient vision-language navigation. *arXiv preprint arXiv:2505.11886*, 2025d.

Zihan Wang and Gim Hee Lee. g3d-lf: Generalizable 3d-language feature fields for embodied tasks. In *CVPR*, pp. 14191–14202, June 2025.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation. In *ICCV*, pp. 15579–15590. IEEE, 2023b.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation. In *ICCV*, pp. 15625–15636, 2023c.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, Junjie Hu, Ming Jiang, and Shuqiang Jiang. Lookahead exploration with neural radiance representation for continuous vision-language navigation. In *CVPR*, pp. 13753–13762, 2024.

Zihan Wang, Seungjun Lee, and Gim Hee Lee. Dynam3d: Dynamic layered 3d tokens empower vlm for vision-and-language navigation. In *NeurIPS*, 2025e.

Zihan Wang, Xiangyang Li, Jiahao Yang, and Shuqiang Jiang. Sim-to-real transfer via 3d feature fields for vision-and-language navigation. In *CoRL*, 2025f.

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, et al. Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. *arXiv preprint arXiv:2512.08186*, 2025a.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025b.

Meng Wei, Chenyang Wan, Peng Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, et al. Ground slow, move fast: A dual-system foundation model for generalizable vision-language navigation. In *ICLR*, volume 2026, pp. 12380–12396, 2026.

Kui Wu, Hao Chen, Churan Wang, Fakhri Karray, Zhoujun Li, Yizhou Wang, and Fangwei Zhong. Hierarchical instruction-aware embodied visual tracking, 2025. URL https://arxiv.org/abs/2505.20710.

Yuchen Xie, Xinyu Zhou, Kuangji Zuo, Yanshuo Lu, Fengrui Huang, Boyu Ma, and Jianfei Yang. Uss: Unified spatial-semantic prompts for embodied visual tracking with latent dynamics learning. *arXiv preprint arXiv:2606.25880*, 2026.

Xinda Xue, Junjun Hu, Minghua Luo, Shichao Xie, Jintao Chen, Zixun Xie, Quan Kuichen, Wei Guo, Zedong Chu, Mu Xu, et al. Omninav: A unified framework for prospective exploration and visual-language navigation. In *ICLR*, volume 2026, pp. 106216–106231, 2026.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

Xuan Yao, Junyu Gao, and Changsheng Xu. Navmorph: A self-evolving world model for vision-and-language navigation in continuous environments. In *ICCV*, pp. 5536–5546. IEEE, 2025.

Xuan Yao, Yuze Zhu, Junyu Gao, Zongmeng Wang, and Changsheng Xu. Sc^2-wm: A self-correcting world model with closed-loop feedback for vision-and-language navigation in continuous environments. *arXiv preprint arXiv:2608.07548*, 2026.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *ICLR*, volume 2025, pp. 28213–28239, 2025.

Kuo-Hao Zeng, Zichen Zhang, Kiana Ehsani, Rose Hendrix, Jordi Salvador, Alvaro Herrasti, Ross Girshick, Aniruddha Kembhavi, and Luca Weihs. Poliformer: Scaling on-policy rl with transformers results in masterful navigators. *arXiv preprint arXiv:2406.20083*, 2024.

Boxuan Zhang, Weipu Zhang, Zhaohan Feng, Wei Xiao, Jian Sun, Jie Chen, and Gang Wang. Mixture-of-world models: Scaling multi-task reinforcement learning with modular latent dynamics. In *ICLR*, 2026a.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and Wang He. Navid: Video-based vlm plans the next step for vision-and-language navigation. In *RSS*, 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 *Robotics: Science and Systems*, 2025.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. In *ICLR*, volume 2026, pp. 127293–127322, 2026b.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. In *ICLR*, volume 2026, pp. 127293–127322, 2026c.

Jiazhao Zhang, Gengze Zhou, Hale Yin, Yiyang Huang, Zixing Lei, Qihang Peng, Haoqi Yuan, Jie Zhang, Xudong Guo, Xiaoyue Chen, et al. Qwen-robotnav technical report: A scalable navigation model designed for an agentic navigation system. *arXiv preprint arXiv:2606.18112*, 2026d.

Lingfeng Zhang, Zeying Gong, Xiaoshuai Hao, Haoxiang Fu, Qiang Zhang, Mingliang Zhou, Hangjun Ye, Xiaojun Liang, Junwei Liang, and Wenbo Ding. Futurenav: Unified world-action modeling for vision-and-language navigation. *arXiv preprint arXiv:2606.30367*, 2026e.

Fangwei Zhong, Peng Sun, Wenhan Luo, Tingyun Yan, and Yizhou Wang. Ad-vat: An asymmetric dueling mechanism for learning visual active tracking. In *ICLR*, 2019a.

Fangwei Zhong, Peng Sun, Wenhan Luo, Tingyun Yan, and Yizhou Wang. Ad-vat+: An asymmetric dueling mechanism for learning and understanding visual active tracking. *IEEE TPAMI*, 43(5): 1467–1482, 2019b.

Fangwei Zhong, Peng Sun, Wenhan Luo, Tingyun Yan, and Yizhou Wang. Towards distraction-robust active visual tracking. In *ICML*, pp. 12782–12792. PMLR, 2021.

Fangwei Zhong, Kui Wu, Hai Ci, Churan Wang, and Hao Chen. Empowering embodied visual tracking with visual foundation models and offline rl. In *ECCV*, pp. 139–155. Springer, 2024.

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

Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In *ECCV*, pp. 260–278. Springer, 2024a.

Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *AAAI*, pp. 7641–7649, 2024b.

Jialong Zuo, Jiahao Hong, Feng Zhang, Changqian Yu, Hanyu Zhou, Changxin Gao, Nong Sang, and Jingdong Wang. Plip: Language-image pre-training for person representation learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

APPENDIX OVERVIEW

We provide additional details, experiments, and analyses to complement the main paper. For ease of reference, the supplementary material is organized as follows:

- **Appendix A: Dataset Details.** Details of the Stage-I dynamics pretraining data and Stage-II task data.
- **Appendix B: Experimental Details.** Further details on the probabilistic latent-transition implementation, EVT gated integration, evaluation metrics, baselines, and training configurations.
- **Appendix C: Additional Experimental Results.** Additional benchmark evaluations, full comparisons, and ablation studies.
- **Appendix D: Real-World Robot Experiments.** Robot and sensor setup, evaluation protocols, quantitative results, and representative real-world rollouts.
- **Appendix E: Qualitative Analysis.** Additional qualitative results for VLN and EVT.

A DATASET DETAILS

A.1 STAGE-I DYNAMICS PRETRAINING DATA

Stage I aims to learn general embodied dynamics from large-scale, heterogeneous egocentric video before task-specific adaptation. We construct an approximately **500-hour** video corpus by mixing the following sources.

Ego4D. We use Ego4D (Grauman et al., 2022) as the primary source of egocentric videos for world-model pretraining. Ego4D is a large-scale first-person video dataset containing approximately 3,670 hours of unscripted daily activities collected from 931 camera wearers across 74 locations in 9 countries. The dataset covers diverse human activities in household, workplace, outdoor, and leisure environments, providing rich temporal observations of human-object and human-environment interactions. Compared with conventional third-person web videos, its egocentric viewpoint more closely resembles the observation stream encountered by embodied agents, making it particularly suitable for learning transferable visual representations and environment dynamics.

Real-world VLN videos. We further incorporate YouTube-VLN(Lin et al., 2023), a large-scale VLN-oriented dataset constructed from real-world house-tour videos. The dataset is collected from 4,078 YouTube videos with a total duration of 433 hours, from which approximately 587K indoor frames are extracted after filtering redundant and irrelevant content. Different from navigation data synthesized from isolated web images, YouTube-VLN preserves the temporal continuity and spatial transitions of real house tours, providing diverse indoor environments, realistic layouts, and naturally occurring navigation motions. We leverage these real-world navigation sequences to enrich the model with indoor layout priors and temporally coherent navigation dynamics.

JRDB-PanoTrack. We further incorporate JRDB-PanoTrack (Le et al., 2024), a dataset of real-world videos captured by a mobile robot in crowded indoor and outdoor environments. It contains 20,000 images sampled at 1 Hz from 54 videos; images from five camera views can be combined into 4,000 360° panoramic frames. The dataset provides approximately 428K panoptic masks and 27K tracking annotations across 72 object classes, capturing both scene structure and object motion over time. We use these temporally ordered observations to expose the model to crowded scenes, changing viewpoints, and persistent objects in real-world robot navigation.

TrackingVideo. We extract egocentric videos from simulated trajectories in the EVT-Bench (Wang et al., 2025c) training set. The videos show moving targets as the agent follows them, capturing changes in relative position, viewpoint, and appearance. We use these sequences to strengthen short-term target-motion modeling for embodied visual tracking.

VLNVideo. We extract egocentric videos from the R2R and RxR training trajectories in VLN-CE (Krantz et al., 2020a). Paired with their navigation instructions, these simulated sequences capture how the agent's view changes as it moves through indoor environments. They complement Ego4D with visual transitions more closely aligned with instruction-following navigation.

Table 5: Stage I dynamics pretraining corpus (∼500 h).

Source	Hours	Domain	Role
Ego4D	*210*	egocentric daily	general dynamics
Real-world VLN	*137*	indoor nav.	navigation dynamics
JRDB-PanoTrack	*70*	crowded outdoor	distractor robustness
TrackingVideo	*70*	visual tracking	target-relative motion
VLNVideo	*10*	VLN-oriented	instruction-aligned nav.
Total	∼500	heterogeneous	Stage I pretraining

A.2 Stage-II Task Data

Stage II adapts the pretrained dynamics model to VLN and EVT with task-labeled trajectories.

A.2.1 VLN Training Data

R2R / RxR (VLN-CE). We train on R2R and RxR under the VLN-CE continuous-control setting. Expert trajectories are converted into temporally aligned samples containing language instructions, egocentric RGB frames, agent states, and navigation actions. The `hybrid-v1` annotations additionally contain the trajectory information extracted from these expert demonstrations. We use them to supervise a structured discrete VLN action head with a prediction horizon of $H = 8$. To mitigate action-type imbalance, we resample training examples with target ratios of $[3.5, , 2.0, , 2.0, , 1.5]$. Each example uses eight historical frames, with a token budget corresponding to at most 313,600 pixels per frame.

VLNVerse. We incorporate the fine-grained and coarse-grained training splits of VLNVerse (Lin et al., 2025), which contain 3,963 and 11,895 instruction–trajectory pairs, respectively, across 177 training scenes. The fine-grained task provides step-by-step navigation instructions, whereas the coarse-grained task specifies goals through higher-level descriptions. We convert these trajectories into the same unified sequential format used for R2R/RxR, aligning instructions with temporally ordered egocentric observations, agent states, and navigation actions. This preprocessing yields approximately 0.6 million step-level training samples, expanding the diversity of instruction styles and scenes used for VLN training.

VLN-PE. VLN-PE (Wang et al., 2025a) extends R2R navigation to a robot-oriented simulation platform with realistic visual scenes and support for physical locomotion. Its R2R split contains 8,679 training, 658 validation-seen, and 1,347 validation-unseen episodes after trajectories involving stairs are filtered out. We evaluate WorldWeave on R2R Val-Unseen using the Flash controller to examine how the navigation policy transfers from VLN-CE to the VLN-PE environment. Because this evaluation uses Flash control, it measures navigation under the changed simulation and visual conditions without testing low-level physical locomotion.

A.2.2 EVT Training Data

EVT-Bench. EVT-Bench is built on Habitat (Puig et al., 2024) with 804 scenes and 100 humanoid avatars, comprising 25,986 episodes across three subtasks: Single-Target Tracking(ST), Distracted Tracking(DT), and Ambiguity Tracking(AT), each with 1,405 test episodes. Since prior works do not release training datasets, we follow TrackVLA (Wang et al., 2025c) to collect about 1.2M interaction steps. Each trajectory provides structured supervision, including target-relative positions and shortest-path waypoints. STT/DT/AT are mixed at a ratio of 2:1:1. We will release our dataset upon acceptance.

UnrealZoo. We use the single-target tracking dataset provided by UnrealZoo (Zhong et al., 2025), comprising approximately 200k interaction steps from eight training environments disjoint from the ten environments used for evaluation. Each trajectory contains egocentric RGB observations, agent actions, and target annotations, including relative position and visibility. Occlusions, viewpoint changes, and background distractions make this benchmark suitable for evaluating long-horizon tracking and cross-scene generalization.

A.2.3 VLM Training Data

ScanQA. We incorporate ScanQA (Azuma et al., 2022) to enhance the model's language-grounded spatial understanding of indoor environments. Built upon 800 indoor scenes from ScanNet, ScanQA contains over 41K question-answer pairs that associate natural-language queries with objects and spatial configurations in 3D scenes. Unlike conventional image-based visual question answering, ScanQA requires reasoning over the geometry, object relationships, and spatial layout of complete 3D environments. We use these question-answer pairs as auxiliary vision-language supervision to strengthen the model's object-centric and spatial reasoning capabilities, complementing the temporal and interaction-rich experience provided by egocentric video data.

VLM Co-training Data. To preserve the general vision-language capabilities of the pretrained VLM backbone during embodied training, we additionally incorporate the LLaVA-OneVision-COCO dataset following the multimodal co-training strategy of StarVLA(Community, 2026). The dataset provides general-purpose image-language supervision and is jointly optimized with embodied task data through a language modeling objective. This auxiliary supervision mitigates over-specialization to task-specific embodied data and helps retain general visual understanding and language grounding capabilities.

SYNTH-PEDES. SYNTH-PEDES (Zuo et al., 2024) is a large-scale person image–text dataset containing approximately 4.79 million images of 312,321 identities and 12.14 million automatically generated textual descriptions. Each image is associated with multiple descriptions of pedestrian appearance and attribute annotations. We use this dataset to provide diverse person appearances and language descriptions for learning target recognition and identity-sensitive visual representations in EVT.

A.3 Evaluation Benchmarks

VLN. We evaluate WorldWeave on R2R-CE, RxR-CE, VLNVerse, and VLN-PE. For R2R-CE and RxR-CE, we report results on the Val-Seen and Val-Unseen splits under the VLN-CE continuous-navigation setting. On VLNVerse, we evaluate both the fine-grained and coarse-grained navigation tasks. For VLN-PE, we report results on R2R Val-Unseen using the Flash controller, which assesses transfer to a more realistic visual environment without evaluating low-level physical locomotion.

- **R2R (VLN-CE).** Room-to-Room (R2R) (Krantz et al., 2020a) is a vision-and-language navigation benchmark in which an agent follows a natural-language instruction to reach a destination in an indoor environment. We evaluate R2R in the VLN-CE setting, which transfers R2R instructions and routes to continuous Matterport3D environments in Habitat. The agent uses egocentric visual observations to select low-level navigation actions, evaluating its ability to ground instructions in visual landmarks, maintain spatial context over long trajectories, and navigate through closed-loop control. We report results on both *val-seen* and *val-unseen*: the former evaluates navigation in buildings encountered during training, while the latter measures generalization to new buildings.
- **RxR (VLN-CE).** Room-Across-Room (RxR) (Krantz et al., 2020a) is a vision-and-language navigation benchmark with detailed route instructions. Compared with R2R, its instructions and paths place greater demands on grounding landmarks and following the described route over a longer navigation horizon. We evaluate RxR in the VLN-CE setting, where the agent follows an instruction through a continuous Matterport3D environment using egocentric observations and low-level actions. This benchmark assesses language grounding, temporal and spatial reasoning, route following, and closed-loop navigation. We report results on *val-unseen* to examine generalization to new buildings.

- **VLNVerse.** VLNVerse (Lin et al., 2025) is a vision-language navigation benchmark built in NVIDIA Isaac Sim, comprising 263 diverse 3D scenes with physics-aware movement and collision constraints. We evaluate both its *fine-grained* and *coarse-grained* navigation tasks. Fine-grained navigation requires the agent to follow step-by-step route instructions, testing landmark grounding, sequential instruction execution, and long-term spatial reasoning. Coarse-grained navigation instead specifies a goal through a higher-level description, testing goal identification, visual search, and route planning. Across both tasks, the agent must repeatedly update its actions from visual observations, allowing VLNVerse to assess closed-loop navigation under physical constraints. We report Success Rate (SR) and Success weighted by Path Length (SPL) to measure task completion and navigation efficiency, respectively.
- **VLN-PE.** VLN-PE (Wang et al., 2025a) is a vision-and-language navigation benchmark built on GRUTopia and NVIDIA Isaac Sim to study how visual conditions and robot embodiment affect navigation performance. It transfers R2R episodes to its simulation platform and supports different robot embodiments and control modes. We evaluate on *R2R val-unseen* using the *flash controller*, which moves the agent directly to the target position specified by each action. This setting tests whether the model can ground instructions in egocentric observations, select navigation actions over a sequence of steps, stop near the goal, and generalize to unseen buildings under the VLN-PE visual and simulation setting.

EVT. We evaluate on EVT-Bench (STT/DT/AT) (Wang et al., 2025c), UnrealZoo (Zhong et al., 2025), and Gym-UnrealCV (Qiu et al., 2017).

- **EVT-Bench.** EVT-Bench evaluates an agent's ability to follow a moving humanoid target using visual observations and a language instruction. Its test split contains 4,215 episodes across 101 unseen indoor scenes, with no overlap in scenes or avatars between training and testing. The benchmark comprises three tasks of increasing difficulty: Single-Target Tracking (STT) tests basic target following; Distracted Tracking (DT) tests whether the agent can identify and follow a person specified by a detailed appearance description; and Ambiguity Tracking (AT) tests whether it can maintain the correct target identity when visually identical distractors are present and the instruction is ambiguous. Together, these tasks assess continuous tracking, language-grounded target identification, resistance to distractors, and generalization to unseen scenes and people. We report success rate (SR), tracking rate (TR), and collision rate (CR) to measure final tracking success, tracking quality throughout an episode, and safety, respectively.
- **Gym-UnrealCV.** We evaluate WORLDWEAVE on Gym-UnrealCV under a zero-shot scene transfer setting: none of its evaluation scenes is used during training. The benchmark covers single-target tracking, tracking amid visually similar distractors, and tracking unseen target categories. These settings test whether the model can sustain pursuit over time, retain the correct target when distractors appear, and generalize to new objects and environments. Following the evaluation protocol, each episode lasts up to 500 steps; we report episode length (EL) and success rate (SR) over 100 episodes.
- **UnrealZoo.** UnrealZoo provides photorealistic virtual environments for evaluating embodied agents. For active visual tracking, the original paper tests agents in 16 unseen environments spanning indoor, palace, wilderness, and modern scenes. It primarily assesses whether an agent can maintain long-term tracking across unfamiliar scene layouts and obstacles; the paper also uses additional distractor settings to test target identification under visual distractions and occlusion. In our evaluation, we select ten unseen environments and report average accumulated reward (AR), episode length (EL), and success rate (SR) to measure tracking performance, duration, and reliability. Because our instruction changes four times within each episode, this setting additionally tests whether WORLDWEAVE can respond to changing goals during closed-loop tracking.

B EXPERIMENTAL DETAILS

B.1 PROBABILISTIC LATENT TRANSITION IMPLEMENTATION.

$\mathcal{I}$, $\mathcal{C}$, and $\mathcal{F}$ are implemented as Transformer-based recurrent modules. To model latent transitions probabilistically, $\mathcal{I}$ and $\mathcal{C}$ parameterize the future-informed posterior and causal prior, respectively,

as diagonal Gaussians. During training, latent actions are sampled via reparameterization:

$$\boldsymbol{z}_t = \boldsymbol{\mu}_t + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{11}$$

This formulation allows the causal prior to match the future-informed transition distribution through $\mathcal{L}_{\mathrm{KL}}$, rather than regressing to a single deterministic latent action.

At inference, $\mathcal{I}$ and the representation decoder $d(\cdot)$ are discarded. Only $\mathcal{C}$ and $\mathcal{F}$ are retained, with the mean of the causal prior used as $\hat{\boldsymbol{z}}_t$ for deterministic prediction of $\hat{\boldsymbol{s}}_{t+1}$.

B.2 Gated Progressive Integration for EVT

As described in Sec. 3.2, EVT progressively uses target-relative position and future ego-motion to inform subsequent predictions. We implement these conditional dependencies through gated residual integration between consecutive prediction stages.

Let $\boldsymbol{q}_t^{\mathrm{pos}}$, $\boldsymbol{q}_t^{\mathrm{traj}}$, and $\boldsymbol{q}_t^{\mathrm{act}}$ denote the representations used for target-position, ego-trajectory, and action prediction, respectively. The target-position predictor first estimates $\boldsymbol{x}_t^{\mathrm{EVT}} \in \mathbb{R}^2$. To condition trajectory prediction on this estimate, we map the predicted position into the representation space and integrate it into the trajectory representation through a learned gate:

$$\tilde{\boldsymbol{q}}_t^{\mathrm{traj}} = \boldsymbol{q}_t^{\mathrm{traj}} + \boldsymbol{g}_t^{\mathrm{pos}} \odot \phi_{\mathrm{pos}}\left(\boldsymbol{x}_t^{\mathrm{EVT}}\right), \tag{12}$$

where $\phi_{\mathrm{pos}}(\cdot)$ projects the two-dimensional target position into the representation space and $\boldsymbol{g}_t^{\mathrm{pos}}$ is a learned gate controlling its contribution. The future ego-trajectory $\boldsymbol{\xi}_t^{\mathrm{EVT}} \in \mathbb{R}^{K\times 2}$ is then predicted from $\tilde{\boldsymbol{q}}_t^{\mathrm{traj}}$.

Similarly, the predicted trajectory is mapped into the representation space and integrated into the action representation:

$$\tilde{\boldsymbol{q}}_t^{\mathrm{act}} = \boldsymbol{q}_t^{\mathrm{act}} + \boldsymbol{g}_t^{\mathrm{traj}} \odot \phi_{\mathrm{traj}}\left(\boldsymbol{\xi}_t^{\mathrm{EVT}}\right), \tag{13}$$

where $\phi_{\mathrm{traj}}(\cdot)$ encodes the predicted future ego-trajectory and $\boldsymbol{g}_t^{\mathrm{traj}}$ controls its residual contribution. The continuous tracking action $\boldsymbol{a}_t^{\mathrm{EVT}} = (v_t, \omega_t)$ is predicted from $\tilde{\boldsymbol{q}}_t^{\mathrm{act}}$.

For both stages, the gate is computed from the current representation and the projected intermediate prediction:

$$\boldsymbol{g} = \sigma(f_{\mathrm{gate}}\left([\boldsymbol{q}; \phi(\boldsymbol{z})]\right)), \tag{14}$$

where $[\cdot;\cdot]$ denotes feature concatenation, f_{gate} is a lightweight multilayer perceptron, and $\sigma(\cdot)$ is the sigmoid function. This design allows intermediate predictions to influence subsequent stages through residual updates while preserving the original task representation.

B.3 Evaluation Metrics

Vision-and-Language Navigation. We report Success Rate (SR), Success weighted by Path Length (SPL), Navigation Error (NE), normalized Dynamic Time Warping (nDTW), and Success weighted by Dynamic Time Warping (SDTW).

- **TL (Trajectory length)** measures the average length of the predicted navigation trajectories.
- **NE (Navigation Error)** measures the average distance (in meter) between the agent's final position in the predicted trajectory and the target in the ground truth.
- **SR (Success Rate)** is the proportion of the agent stopping in the predicted route within a threshold distance (set as 3 meters) of the goal in the reference route.
- **OSR (Oracle Success Rate)** is the proportion of the closest point in the predicted trajectory to the target in the reference trajectory within a threshold distance.
- **SPL (Success weighted by Path Length)** is a comprehensive metric method integrating SR and TL that takes both effectiveness and efficiency into account.
- **NDTW (Normalized Dynamic Time Warping)** measures the normalized cumulative distance between reference path and agent position.

- **SDTW (Success weighted by normalized Dynamic Time Warping)** is a comprehensive metric method integrating NDTW and SR that takes both path efficiency and task completion into account.

Embodied Visual Tracking. We follow previous methods and adopt standard metrics for evaluating, including success rate (SR), average episode length (EL), average accumulated reward (AR), tracking rate (TR), and collision rate (CR).

For EVTBench, we report the following metrics:

- **Success Rate (SR):** The proportion of episodes in which the agent ends while facing the target and maintaining a distance of 1–3 m.
- **Tracking Rate (TR):** The proportion of time steps at which the agent successfully tracks the target. Let S be the number of successfully tracked steps and L the total number of steps; then $\mathrm{TR} = S/L$.
- **Collision Rate (CR):** The proportion of episodes terminated by a collision between the agent and the target humanoid avatar.

For UnrealZoo and Gym-UnrealCV, we report the following metrics over 100 episodes:

- **Episode Length (EL):** The average number of steps per episode under the predefined termination conditions, reflecting long-term tracking capability.
- **Success Rate (SR):** The proportion of episodes that satisfy the benchmark's success criterion.
- **Accumulated Reward (AR):** The average cumulative reward per episode, reflecting overall tracking performance.

B.4 Baselines

To ensure a fair and comprehensive comparison, we evaluate our approach against a diverse set of baselines covering classical control, reinforcement learning, and vision–language–action (VLA) paradigms.

Vision-and-Language Navigation.

- **SC²-WM** (Yao et al., 2026): A world-model-based VLN-CE method that uses predicted future states as feedback to refine navigation decisions and conditionally adapts the world model at test time when further correction is needed.
- **NaVILA** (Cheng et al., 2024): A two-level framework for legged robot navigation that uses a VLA model to predict spatially grounded, language-formatted mid-level actions, which a learned locomotion policy executes as low-level movements.
- **Uni-NaVid** (Zhang et al., 2025): A unified VLA model designed for general navigation tasks, including human following.
- **StreamVLN** (Wei et al., 2025b): A streaming VLN framework that combines a fast dialogue context for responsive action generation with a slowly updated visual memory for long-horizon navigation, using 3D-aware token pruning and KV-cache reuse to limit inference cost.
- **DualVLN** (Wei et al., 2026): A dual-system VLN model in which a VLM reasons over the scene and predicts mid-term waypoint goals, while a fast diffusion policy converts those goals into smooth trajectories for real-time navigation.
- **InternVLA-N1** (Team, 2025): An open dual-system VLN model that uses a vision-language planner to set navigation goals and a fast diffusion policy to execute them, with learned latent plans connecting the two systems for asynchronous, real-time control.
- **Qwen-RobotNav** (Zhang et al., 2026d): A Qwen3-VL-based navigation model with configurable task modes and visual-history encoding, allowing one backbone to support instruction following, target tracking, and other navigation tasks within an agentic system.

Embodied Visual Tracking.

- **DiMP** (Bhat et al., 2019): Utilizes a pre-trained video tracker to generate target bounding boxes as scene representations and applies a PID controller for motion control.
- **SARL** (Luo et al., 2019): An online reinforcement learning (RL) approach that encodes RGB observations into latent visual features and trains an end-to-end Conv-LSTM policy via RL.
- **AD-VAT** (Zhong et al., 2019a): Introduces an asymmetric dueling mechanism and trains an RL-based tracker with a learnable adversarial target to improve robustness.
- **AD-VAT+** (Zhong et al., 2019b): An enhanced version of AD-VAT that incorporates a two-stage training scheme, aiming to improve performance in cluttered and obstacle-rich environments.
- **TS** (Zhong et al., 2021): A teacher-student framework that extends behavior cloning by leveraging a pose-based teacher to provide real-time supervision for the vision-based student policy during interaction.
- **EVT** (Zhong et al., 2024): An offline RL-based framework designed for dynamic target following, which integrates vision foundation models to enhance perception and robustness.
- **IBVS** (Gupta et al., 2016): A model-based method that takes the target bounding box as input and applies a Kalman filter-based visual servoing algorithm to follow the target.
- **PoliFormer** (Zeng et al., 2024): A reinforcement learning-based navigation framework that explicitly encodes target bounding boxes into the observation space to enhance tracking accuracy.
- **Uni-NaVid** (Zhang et al., 2025). : A unified VLA model designed for general navigation tasks, including human following.
- **TrackVLA** (Wang et al., 2025c): A unified VLA framework for embodied visual tracking that tightly couples target recognition and trajectory planning within a single model.
- **NavFoM** (Zhang et al., 2026b): A cross-task and cross-embodiment navigation foundation model that unifies vision-and-language navigation, object searching, target tracking, and autonomous driving within a single architecture.
- **TrackVLA++** (Liu et al., 2025). : An enhanced VLA framework for embodied visual tracking that extends TrackVLA with explicit spatial reasoning and long-horizon target identification memory.
- **USS** (Xie et al., 2026): An end-to-end embodied visual tracking framework that accepts text, point, box, and mask prompts to specify the target, and uses a latent dynamics model to improve temporal tracking robustness.
- **VLingNav** (Wang et al., 2026): A VLA model for embodied navigation that combines adaptive reasoning with visual-assisted linguistic memory to support long-horizon planning and reduce repeated exploration.
- **CoMaTrack** (Liu et al., 2026): A VLA framework for embodied visual tracking that uses competitive multi-agent reinforcement learning, training the tracker against adaptive opponents to improve planning and robustness to interference.

B.5 Implementation Details

Stage I: Shared World-Dynamics Learning. We train the shared world-dynamics model on temporally ordered observations without requiring physical action annotations. Each training sample contains the observation history up to time t and the subsequent observation at $t+1$. The vision encoder remains frozen, while a shared LoRA adapter adapts the VLM backbone to produce contextual representations h_t and h_{t+1}. Task-specific world-state representations s_t and s_{t+1} are obtained from these contextual features.

The inverse dynamics module $\mathcal{I}$ takes (h_t, h_{t+1}) as input and parameterizes a diagonal Gaussian posterior $q(z_t \mid h_t, h_{t+1})$. We sample z_t using the reparameterization trick and pass it, together

with s_t, to the shared forward dynamics module $\mathcal{F}$ to predict s_{t+1}. This future-informed pathway is supervised by the state prediction loss $\mathcal{L}_{\text{state}}$. In parallel, the causal predictor $\mathcal{C}$ uses only $(h_{\leq t}, s_t)$ to parameterize a diagonal Gaussian prior $p(z_t \mid h_{\leq t}, s_t)$. We align the prior with the posterior using $\mathcal{L}_{\text{KL}}$. A latent action sampled from the causal prior is passed through $\mathcal{F}$, and the resulting predicted state is decoded to reconstruct h_{t+1} under $\mathcal{L}_{\text{rec}}$.

We jointly optimize the shared VLM LoRA adapter, $\mathcal{I}$, $\mathcal{C}$, $\mathcal{F}$, and the representation decoder for 80,000 steps using AdamW with a weight decay of 1×10^{-8} and a batch size of 4. The training objective is

$$\mathcal{L}_{\text{Dyn}} = \mathcal{L}_{\text{state}} + \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{KL}}. \tag{15}$$

We apply this objective to the heterogeneous video sources while retaining task-specific world-state representations and sharing the transition model. After Stage I, we discard $\mathcal{I}$ and the representation decoder. At inference time, we use the mean of the causal prior as the latent action and predict the next world state with $\mathcal{C}$ and $\mathcal{F}$.

Stage II: Task Adaptation. We adapt the Stage I model to VLN and EVT with separate task-specific LoRA adapters and prediction heads. The pretrained dynamics modules and the VLM vision encoder remain frozen. The LoRA rank, scaling factor, and dropout rate are 32, 64, and 0.05, respectively.

Visual input configuration. For EVT, each timestep contains 28 temporally ordered RGB frames of resolution $224 \times 224 \times 3$: the current observation and 27 historical frames. Each frame is processed independently by the frozen vision encoder $\psi^{(v)}_{\text{VLM}}(\cdot)$, and the resulting features are concatenated along the temporal dimension. GridPool aggregates these features into 64 visual tokens of dimension 2560. This input provides a short temporal history while keeping the visual representation compact. For VLN, each timestep contains eight temporally ordered RGB frames. Their features are concatenated without GridPool, retaining $8 \times 64 = 512$ visual tokens of dimension 2560. For each task, the visual tokens, encoded task instruction, and learnable tokens are jointly fed into the Qwen backbone.

Optimization and supervision. Training consists of 40,000 steps of Dual-LoRA adaptation with 4,000 warm-up steps, followed by residual continuation from step 40,000 to step 80,000 with a 2,000-step warm-up. The residual gate is initialized to $\alpha_0 = 0.05$. We use AdamW with a weight decay of 1×10^{-8}. The learning rates are 1×10^{-5} for the trainable VLM parameters and 1×10^{-4} for the task heads and residual modules. A cosine schedule decreases the learning rate to a minimum of 1×10^{-6}. VLN and EVT are jointly optimized with balanced task sampling.

For VLN, we jointly supervise the navigation action and an eight-step future ego-trajectory. Categorical cross-entropy is used for the action type and motion parameters, while an ℓ_1 loss is computed over valid future trajectory steps. For EVT, we jointly supervise the agent-centric target position, the eight-step future ego-trajectory, and the linear and angular velocities, as specified by the EVT objective in the main text. The frozen dynamics modules provide predicted future-state features to both task branches.

We train on eight NVIDIA H200 GPUs using DeepSpeed ZeRO-2 and bfloat16 precision. The per-device batch size is 4, yielding a global batch size of 32. Gradients are clipped to a maximum norm of 1.0. The complete Stage II schedule takes approximately 18 hours.

C Additional Experimental Results

C.1 VLN-PE and VLNVerse.

We further examine whether these improvements persist under different navigation settings. On VLN-PE, WorldWeave attains 55.62 SR and 50.12 SPL with the flash controller, compared with 36.12 and 34.07 for the matched baseline. On VLNVerse, gains appear under both instruction granularities: SR climbs from 25.12 to 48.92 with fine-grained instructions and from 18.24 to 38.23 with coarse-grained ones. These results suggest that task-specific predictive adaptation is not restricted to a particular instruction granularity. Meanwhile, the SPL improvement is more modest, especially in the fine-grained setting (21.82→28.93), indicating that the benefit is more pronounced in goal completion than in path efficiency. This distinction reflects our design focus: WorldWeave supplies

Table 6: Stage II training hyperparameters.

Hyperparameter	Value
Backbone	Qwen3-VL-4B
GPUs	$8\times$ NVIDIA H200
Wall-clock (full Stage II schedule)	$\sim$18 hours
Optimizer / weight decay	AdamW / 1×10^{-8}
LR schedule	cosine, min LR 1×10^{-6}
LR (backbone / shared WM)	1×10^{-5}
LR (task heads / residual modules)	1×10^{-4}
Per-device / global batch	4 / 32
Parallelism	DeepSpeed ZeRO-2
Precision / grad clip	bfloat16 / 1.0
LoRA r / α / dropout	32 / 64 / 0.05
Dual-LoRA steps / warm-up	40k / 4k
Residual continuation	40k$\rightarrow$80k / 2k warm-up
λ_{WM} (default)	0
Residual gate init α_0	0.05
K	8
λ	10^{-3}
d	2560
d_z	640
N_h^τ	64
N_s^τ	64
N_z	2

Table 7: **Comparison on VLN-PE with flash controller** (R2R Val-Unseen).

Method	OS↑	SR↑	SPL↑
Seq2Seq (Krantz et al., 2020b)	43.05	15.74	9.70
CMA (Krantz et al., 2020b)	45.06	20.94	14.06
RDP (Wang et al., 2025b)	42.54	24.94	17.54
InternVLA-N1 (Team, 2025)	67.63	60.36	54.93
QwenRobotNav (Zhang et al., 2026d)	72.70	60.28	55.24
Baseline (Qwen3-VL-4B)	50.87	36.12	34.07
WORLDWEAVE (Qwen3-VL-4B)	**62.54**	**55.62**	**50.12**

Table 8: **Comparison on VLNVerse test split under fine-grained and coarse-grained instruction settings.**

Method	Fine-grained			Coarse-grained		
	OSR↑	SR↑	SPL↑	OSR↑	SR↑	SPL↑
InternVLA-N1 (Team, 2025)	38.69	28.95	25.00	23.32	17.51	16.54
Uni-NaVid (Zhang et al., 2025)	67.02	45.74	26.91	42.70	29.68	13.47
NavFoM (Zhang et al., 2026b)	69.68	51.59	32.40	45.93	38.02	23.15
QwenRobotNav (Zhang et al., 2026d)	70.36	62.61	56.22	51.03	41.25	37.37
Baseline (Qwen3-VL-4B)	33.52	25.12	21.82	21.66	18.24	16.62
WORLDWEAVE (Qwen3-VL-4B)	**65.81**	**48.92**	**28.93**	**47.18**	**38.23**	**23.05**

predictive state and motion context rather than explicitly optimizing path length. Overall, improvements across these settings demonstrate that the VLN gains of WORLDWEAVE generalize beyond a single benchmark or evaluation protocol.

C.2 UNREALZOO AND GYM-UNREALCV.

The complementary evaluations in Tables 9 and 10 demonstrate that the improvements extend beyond EVTBench. On UnrealZoo, WORLDWEAVE surpasses the matched baseline under dynamic instruction changes across the representative environments reported here; results on additional environments appear in the Appendix. On Gym-UnrealCV, gains are most pronounced under distractors and unseen objects, where SR climbs from 0.61 to 0.84 and from 0.76 to 1.00, respectively. These results provide complementary evidence that tracking improvements persist under changing targets and zero-shot conditions.

C.3 FULL COMPARISON ON VLN AND EVT BENCHMARKS

In the appendix, we provide detailed VLN results on R2R-CE (Table 11), RxR-CE (Table 12). For EVT, results on EVT-Bench is reported in Tables 13 respectively.

Table 9: **Results on UnrealZoo (Zhong et al., 2025).** We compare performance across ten environments with dynamic instruction changes. Each cell reports Average Accumulated Reward (AR), Average Episode Length (EL), and Success Rate (SR) in the format AR/EL/SR. The user instruction changes four times per episode. Results exceeding the base model are shown in **bold**, and the best results are underlined. * indicates results evaluated using official checkpoints.

Method	FPS	Environment Name				
		FlexibleRoom	Suburb	Supermarket	Parking Lot	Old Factory
Bbox-based PID(Wu et al. (2025))	50	154 / 365 / 0.50	124 / 386 / 0.36	175 / 298 / 0.26	112 / 356 / 0.38	80 / 309 / 0.30
Ensembled RL(Zhong et al. (2024))	50	183 / 330 / 0.36	126 / 294 / 0.22	114 / 393 / 0.38	169 / 301 / 0.42	64 / 334 / 0.36
OpenVLA(Kim et al. (2024))	8	123 / 253 / 0.25	63 / 311 / 0.18	68 / 233 / 0.10	12 / 205 / 0.08	5 / 226 / 0.12
GPT-4o(Hurst et al. (2024))	0.38	15 / 194 / 0.10	14 / 241 / 0.20	54 / 286 / 0.16	23 / 286 / 0.30	5 / 297 / 0.32
VLM-Assistant (GPT-4o)(Hurst et al. (2024))	-	-	-	325 / 493 / 0.94	313 / 470 / 0.88	315 / 484 / 0.92
HIEVT(Wu et al. (2025))	50	278 / 500 / 1.00	166 / 445 / 0.72	212 / 422 / 0.64	169 / 491 / 0.93	128 / 469 / 0.84
OpenTrackVLA*(Lee & Zhao (2025))	5.6	285 / 475 / 0.89	202 / 341 / 0.33	293 / 430 / 0.64	313 / 435 / 0.69	288 / 422 / 0.61
Baseline(Qwen3-VL-4B)(Bai et al., 2025b)	10	223 / 500 / 1.00	231 / 436 / 0.71	192 / 410 / 0.65	210 / 465 / 0.87	227 / 470 / 0.81
Ours(Qwen3-VL-4B)	10	**362 / 500 / 1.00**	**372 / 473 / 0.86**	**248 / 457 / 0.85**	**326 / 478 / 0.92**	**301 / 476 / 0.89**
Method	FPS	Environment Name				
		Container Yard	Desert Ruins	Brass Garden	Modular Old Town	Roof City
Bbox-based PID(Wu et al. (2025))	50	140 / 369 / 0.42	128 / 297 / 0.26	120 / 302 / 0.38	73 / 305 / 0.32	64 / 289 / 0.34
Ensembled RL(Zhong et al. (2024))	50	27 / 327 / 0.24	56 / 368 / 0.34	-8 / 348 / 0.34	14 / 356 / 0.32	19 / 322 / 0.38
OpenVLA(Kim et al. (2024))	8	43 / 232 / 0.08	22 / 219 / 0.12	30 / 236 / 0.08	-9 / 213 / 0.16	96 / 303 / 0.22
GPT-4o(Hurst et al. (2024))	0.38	-43 / 149 / 0.00	-5 / 249 / 0.26	3 / 243 / 0.18	29 / 304 / 0.28	18 / 256 / 0.20
VLM-Assistant (GPT-4o)(Hurst et al. (2024))	-	-	-	-	-	-
HIEVT(Wu et al. (2025))	50	156 / 469 / 0.76	148 / 434 / 0.74	126 / 425 / 0.72	85 / 433 / 0.64	86 / 400 / 0.58
OpenTrackVLA*(Lee & Zhao (2025))	5.6	297 / 432 / 0.70	296 / 427 / 0.74	206 / 390 / 0.43	290 / 436 / 0.72	283 / 414 / 0.66
Baseline(Qwen3-VL-4B)(Bai et al., 2025b)	10	351 / 490 / 0.96	328 / 486 / 0.93	207 / 425 / 0.70	204 / 469 / 0.87	186 / 453 / 0.84
Ours(Qwen3-VL-4B)	10	**370 / 500 / 1.00**	**330 / 487 / 0.93**	**290 / 465 / 0.86**	**355 / 486 / 0.94**	**264 / 470 / 0.87**

Table 10: **Zero-shot results on Gym-UnrealCV.** Episode Length (EL) is the average number of steps before termination, with a maximum of 500 steps; Success Rate (SR) is the proportion of episodes lasting the full 500 steps. Results exceeding the base model are shown in **bold**, the best results are underlined, and gray text indicates methods whose code is not publicly available.

Methods	*Single Target*		*Distractor*		*Unseen Objects*	
	EL↑	SR↑	EL↑	SR↑	EL↑	SR↑
DiMP(Bhat et al. (2019))	367	0.58	309	0.27	-	-
SARL(Luo et al. (2019))	394	0.57	240	0.14	-	-
AD-VAT(Zhong et al. (2019a))	416	0.62	220	0.12	-	-
AD-VAT+(Zhong et al. (2019b))	454	0.76	224	0.12	-	-
TS(Zhong et al. (2021))	474	0.86	371	0.48	-	-
EVT(Zhong et al. (2024))	490	0.95	459	0.81	480	0.96
OpenTrackVLA*(Lee & Zhao (2025))	465	0.82	436	0.76	232	0.35
TrackVLA(Wang et al. (2025c))	500	1.00	474	0.91	500	1.00
TrackVLA++(Liu et al. (2025))	500	1.00	484	0.92	500	1.00
Baseline(Qwen3-VL-4B)(Bai et al., 2025b)	495	0.97	381	0.61	465	0.76
Ours(Qwen3-VL-4B)	**500**	**1.00**	**470**	**0.84**	**500**	**1.00**

Table 11: **Results on R2R-CE.** Results exceeding the base model are shown in **bold**, and the best results within each observation setting (panoramic or monocular) are underlined. Gray text indicates methods whose code is not publicly available. † indicates methods based on large language or vision-language models.

Methods	Val Seen					Val Unseen				
	TL ↓	NE ↓	OSR	SR	SPL	TL ↓	NE ↓	OSR	SR	SPL
Monocular										
LAW (Raychaudhuri et al., 2021) [EMNLP21]	9.34	6.35	49	40	37	8.89	6.83	44	35	31
CM^2 (Georgakis et al., 2022) [CVPR22]	12.05	6.10	50.7	42.9	34.8	11.54	7.02	41.5	34.3	27.6
WS-MGMap (Chen et al., 2022a) [NeurIPS22]	10.12	5.65	51.7	46.9	43.4	10.00	6.28	47.6	38.9	34.3
NaVid (Zhang et al., 2024) [RSS24]	-	-	-	-	-	-	5.47	49.1	37.4	35.9
ETPNav/p (Wang et al., 2025f) [CoRL24]	-	-	-	-	-	-	6.81	42.4	32.9	23.1
VLN-3DFF (Wang et al., 2025f) [CoRL24]	-	-	-	-	-	-	5.95	55.8	44.9	30.4
NavMorph (Yao et al., 2025) [ICCV2025]	20.03	4.58	62.7	55.8	38.9	22.54	5.75	56.9	47.9	33.2
g3D-LF (Wang & Lee, 2025) [CVPR2025]	-	-	-	-	-	-	5.70	59.5	47.2	34.6
monoVLN (Lu et al., 2025) [CVPR2025]	-	-	-	-	-	-	4.61	62.4	54.8	44.4
SC^2-WM (Yao et al., 2026)[ICML2026]	17.26	4.53	64.3	56.0	41.9	17.65	5.37	58.8	50.9	37.2
NaVILA† (Cheng et al., 2024) [arxiv2025]	-	-	-	-	-	-	5.43	62.5	54.0	49.0
UniNaVid† (Zhang et al., 2024) [arxiv2025]	-	-	-	-	-	-	5.58	53.3	47.0	42.7
StreamVLN† (Wei et al., 2025b) [arxiv2025]	-	-	-	-	-	-	4.98	64.2	56.9	51.9
DualVLN† (Wei et al., 2025a)[arxiv2025]	-	-	-	-	-	-	4.05	70.7	64.3	58.5
InternVLA-N1† (Team, 2025)[arxiv2025]	-	-	-	-	-	-	4.83	63.3	58.2	54.0
QwenRobotNav† (Zhang et al., 2026d)[arxiv2026]	-	-	-	-	-	-	4.22	73.6	66.9	60.5
Baseline(Qwen3-VL-4B(Bai et al., 2025b))	10.31	5.64	64.1	57.2	44.2	9.00	6.20	52.1	46.9	41.9
Ours(Qwen3-VL-4B)	10.89	**4.72**	**70.0**	**62.4**	**50.2**	11.3	**5.78**	**61.5**	**54.2**	**49.4**
Panoramic										
Seq2Seq (Anderson et al., 2018a) [CVPR18]	9.26	7.12	46	37	35	8.64	7.37	40	32	30
Sim2Sim (Krantz & Lee, 2022) [ECCV22]	11.18	4.67	61	52	44	10.69	6.07	52	43	36
CWP-CMA (Hong et al., 2022) [CVPR22]	11.47	5.20	61	51	45	10.90	6.20	52	41	36
CWP-BERT (Hong et al., 2022)[CVPR22]	12.50	5.02	59	50	44	12.23	5.74	53	44	39
DREAMW (Wang et al., 2023a) [ICCV23]	11.60	4.09	59	66	48	11.30	5.53	49	59	44
GridMM (Wang et al., 2023c) [ICCV23]	12.69	4.21	69	59	51	13.36	5.11	61	49	41
BEVBert (An et al., 2022a) [ICCV23]	13.98	3.77	73	68	60	13.27	4.57	67	59	50
InstructNav† (Long et al., 2024) [arxiv2024]	-	-	-	-	-	7.74	6.89	-	31	24
FSTTA (Gao et al., 2024) [ICML24]	12.39	4.25	69	58	50	11.58	5.27	58	48	42
SmartWay† (Shi et al., 2025) [arxiv2025]	-	-	-	-	-	13.09	7.01	51	29	22
Aux-Think† (Wang et al., 2025d) [arxiv2025]	-	-	-	-	-	-	6.01	52	46	41
Dynam3D† (Wang et al., 2025e) [ICCV2025]	-	-	-	-	-	-	5.34	62	53	46
ETPNav (An et al., 2024) [TPAMI24]	11.78	3.95	72	66	59	11.99	4.71	65	57	49
HNR (Wang et al., 2024) [CVPR24]	11.79	3.67	76	69	61	12.64	4.42	67	61	51
NavMorph (Yao et al., 2025) [ICCV2025]	11.76	3.66	78	70	62	12.68	4.37	68	64	53
g3D-LF (Wang & Lee, 2025) [CVPR2025]	-	-	-	-	-	-	4.53	68	61	52
SC^2-WM (Yao et al., 2026)	12.09	3.28	80	71	64	12.89	4.25	70	66	54
NavFoM† (Zhang et al., 2026b) [ICLR2026]	-	-	-	-	-	-	4.61	72.1	61.7	55.3
ABot-N0† (Chu et al., 2026) [arxiv2026]	-	-	-	-	-	-	3.78	70.8	66.4	63.9
OmniNav† (Xue et al., 2026) [ICLR2026]	-	-	-	-	-	-	3.74	74.6	69.5	66.1
AstraNav-World† (Chen et al., 2025) [arxiv2025]	-	-	-	-	-	-	3.86	73.9	67.9	65.4
QwenRobotNav† (Zhang et al., 2026d) [arxiv2026]	-	-	-	-	-	-	3.80	77.2	69.5	63.6
baseline(Qwen3-VL-4B(Bai et al., 2025b))	10.92	4.15	70.9	64.3	62.0	10.70	5.09	62.7	57.8	52.4
Ours(Qwen3-VL-4B)	11.76	**3.78**	**78.2**	**72.1**	**66.5**	11.22	**4.48**	**69.3**	**63.1**	**57.2**

Note: Following prior work, we report the results with different precision formats across camera configurations——integers for panoramic settings and two decimal places for monocular settings.

Table 12: **Results on the RxR-CE Val-Unseen split.** Results exceeding the base model are shown in **bold**, and the best results within each observation setting (panoramic or monocular) are underlined. Gray text indicates methods whose code is not publicly available.

Methods	TL ↓	NE ↓	OSR ↑	SR ↑	SPL ↑	NDTW ↑	SDTW ↑
Monocular							
LAW (Raychaudhuri et al., 2021)	4.01	10.87	21.0	8.0	8.0	-	-
CM^2 (Georgakis et al., 2022)	12.29	8.98	25.3	14.4	9.2	-	-
WS-MGMap (Chen et al., 2022a)	10.80	9.83	29.8	15.0	12.1	-	-
NaVid (Zhang et al., 2024)	10.59	8.41	34.5	23.8	32.2	-	-
A^2-Nav (Chen et al., 2023)	-	-	-	16.8	6.3	-	-
NavMorph (Yao et al., 2025)	20.28	8.85	43.05	30.76	22.84	44.19	23.30
monoVLN (Lu et al., 2025)	-	8.29	37.7	31.8	26.8	-	25.2
VLN-3DFF (Wang et al., 2025f)	-	8.79	36.7	25.5	18.1	-	-
SC^2-WM (Yao et al., 2026)	20.87	8.36	48.38	35.77	27.20	44.93	26.47
Uni-NaVid (Zhang et al., 2025)	-	6.24	-	48.7	40.9	-	-
NaVILA (Cheng et al., 2024)	-	6.77	-	49.3	44.0	58.8	-
StreamVLN (Wei et al., 2025b)	-	6.22	-	52.9	46.0	61.9	-
DualVLN (Wei et al., 2025a)	-	4.58	-	61.4	51.8	70.0	-
InternVLA-N1 (Team, 2025)	-	5.91	-	53.5	46.1	65.3	-
Qwen-RobotNav (Zhang et al., 2026d)	-	4.15	-	71.3	61.5	68.6	-
Baseline (Qwen3-VL-4B) (Bai et al., 2025b)	11.29	7.24	41.20	45.81	38.12	55.20	40.11
Ours (Qwen3-VL-4B)	13.89	**6.31**	**50.13**	**52.67**	**45.91**	**62.87**	**45.77**
Panoramic							
Seq2Seq (Anderson et al., 2018a)	7.33	12.1	-	13.93	11.96	30.86	11.01
Reborn (An et al., 2022b)	-	5.98	-	48.60	42.05	63.35	41.82
CWP-CMA (Hong et al., 2022)	-	8.76	-	26.59	22.16	47.05	-
CWP-RecBERT (Hong et al., 2022)	-	8.98	-	27.08	22.65	46.71	-
AO-Planner (Chen et al., 2024a)	-	7.06	-	43.3	30.5	50.1	-
LAW-Pano (Raychaudhuri et al., 2021)	4.62	11.04	16.0	10.0	9.0	-	-
UnitedVLN (Dai et al., 2024)	-	5.48	-	57.9	45.9	63.9	48.1
ETPNav (An et al., 2024)	-	5.64	-	54.79	44.89	61.90	45.33
HNR (Wang et al., 2024)	-	5.51	-	56.39	46.73	63.56	47.24
NavMorph (Yao et al., 2025)	21.33	5.67	66.02	58.02	48.98	64.77	48.85
SC^2-WM (Yao et al., 2026)	21.01	5.72	66.77	60.02	49.79	66.77	49.50
NavFoM (Zhang et al., 2026b)	-	4.74	-	64.4	56.2	65.8	-
ABot-N0 (Chu et al., 2026)	-	3.83	-	69.3	60.0	-	-
OmniNav (Xue et al., 2026)	-	3.77	-	73.6	62.0	-	-
AstraNav-World (Chen et al., 2025)	-	3.82	-	72.9	61.5	-	-
Qwen-RobotNav (Zhang et al., 2026d)	-	3.80	-	75.2	65.0	71.9	-
Baseline (Qwen3-VL-4B) (Bai et al., 2025b)	12.61	4.97	60.81	60.21	52.14	57.12	42.40
Ours (Qwen3-VL-4B)	13.88	**4.82**	**67.32**	**68.17**	**61.25**	**63.42**	**47.15**

Table 13: **Results on EVT-Bench.** Success Rate (SR) is the proportion of episodes ending with the agent oriented toward the target and within 1–3 m of it; Tracking Rate (TR) is the proportion of steps with successful target tracking; Collision Rate (CR) is the proportion of episodes terminated by a collision. Results exceeding the base model are shown in **bold**, the best results are underlined, and gray text indicates methods whose code is not publicly available. † uses GroundingDINO as the detector. ‡ uses SoM (Yang et al., 2023) and GPT-4o (Hurst et al., 2024) as the visual foundation model. * denotes results evaluated using official checkpoints.

Methods	*Single-Target Tracking (STT)*			*Distracted Tracking (DT)*			*Ambiguity Tracking (AT)*		
	SR↑	TR↑	CR↓	SR↑	TR↑	CR↓	SR↑	TR↑	CR↓
IBVS†(Gupta et al. (2016))	42.9	56.2	3.75	10.6	28.4	6.14	15.2	39.5	4.90
PoliFormer†(Zeng et al. (2024))	4.67	15.5	40.1	2.62	13.2	44.5	3.04	15.4	41.5
EVT(Zhong et al. (2024))	24.4	39.1	42.5	3.23	11.2	47.9	17.4	21.1	45.6
EVT‡(Zhong et al. (2024))	32.5	49.9	40.5	15.7	35.7	53.3	18.3	21.0	44.9
Uni-NaVid(Vicuna-7B)(Zhang et al. (2025))	25.7	39.5	41.9	11.3	27.4	43.5	8.26	28.6	43.7
OpenTrackVLA(Qwen-0.6B)(Lee & Zhao (2025))	64.8	84.4	5.00	33.6	66.3	8.84	39.6	76.7	6.38
OpenTrackVLA(Qwen-0.6B)*(Lee & Zhao (2025))	47.2	58.7	5.72	22.5	42.8	8.93	28.6	51.8	6.37
USS(Xie et al. (2026))	70.8	72.1	3.13	49.8	54.6	9.89	34.2	35.8	28.2
Qwen-RobotNav(Qwen-4B)(Zhang et al. (2026d))	77.4	90.0	6.40	-	-	-	-	-	-
TrackVLA(Vicuna-7B)(Wang et al. (2025c))	85.1	78.6	1.65	57.6	63.2	5.80	50.2	63.7	17.1
NavFoM(Qwen2-7B)(Zhang et al. (2026c))	85.0	80.5	-	61.4	68.2	-	-	-	-
TrackVLA++(Qwen-7B)(Liu et al. (2025))	86.0	81.0	2.10	66.5	68.8	4.71	51.2	63.4	15.9
VLingNav(Wang et al. (2026))	88.4	81.2	2.07	67.6	73.5	5.51	-	-	-
CoMaTrack(Liu et al. (2026))	92.1	90.3	0.9	74.2	80.5	2.1	57.5	73.4	12.0
Baseline(Qwen3-VL-4B)(Bai et al., 2025b)	41.1	71.2	6.50	16.3	45.7	19.4	14.8	47.1	8.70
Ours(Qwen3-VL-4B)	**78.1**	**88.0**	**4.89**	**41.7**	**75.2**	**14.24**	**51.2**	**74.5**	18.62

C.4 Additional Ablation Studies

How Should Shared Predictions Be Adapted to Each Task? Table 14 examines the task-specific components used together with shared predictive dynamics. Removing either episodic memory or ego-trajectory supervision consistently degrades VLN performance, indicating complementary benefits from historical navigation experience and future-motion supervision. For EVT, removing target-position prediction causes a pronounced drop in SR (78.1 to 68.4) together with a substantial increase in CR (4.89 to 13.07), highlighting the utility of explicitly modeling the agent–target relation before action prediction. Replacing gated integration with direct residual fusion also degrades performance, suggesting that selectively incorporating intermediate predictions benefits the progressive tracking pipeline. Ego-trajectory supervision is retained across all EVT variants, as preliminary experiments showed that the EVT baseline was difficult to optimize without this auxiliary objective.

Table 14: Ablation of task-specific adaptation in WorldWeave on R2R-CE Val-Unseen and EVT-Bench STT. Best results are in **bold**.

VLN Setting	OSR ↑	SR ↑	SPL ↑
Baseline	52.1	46.9	41.9
WorldWeave	**61.5**	**54.2**	**49.4**
w/o Episodic Memory	58.1	51.0	46.4
w/o Ego-Trajectory Supervision	56.2	49.4	46.9
EVT Setting	**SR ↑**	**TR ↑**	**CR ↓**
Baseline	41.1	71.2	6.50
WorldWeave	**78.1**	**88.0**	**4.89**
w/o Target-Position Prediction	68.4	82.9	13.07
w/o Gated Integration	75.0	82.2	7.29

D REAL-WORLD ROBOT EXPERIMENTS

We further evaluate WORLDWEAVE on a physical mobile robot to examine its applicability beyond simulation. We consider both VLN and EVT in real indoor environments and evaluate the model under progressively more challenging navigation instructions and target-motion patterns.

D.1 ROBOT AND SENSOR SETUP

We conduct the real-world experiments on a TITA wheel-legged robot (Direct Drive Tech). The robot is equipped with an NVIDIA Jetson Orin NX 16GB computing platform, providing up to 100 TOPS of onboard computation. Visual observations are captured using a LI-AR0234CS-STEREO-GMSL2-30 stereo camera, while a CH201 ultrasonic sensor and an ST VL53L5CX SPAD ToF sensor provide complementary short-range obstacle information. For VLN, RGB observations are captured at a resolution of 640×480 and JPEG-compressed with a quality factor of 75 before being processed by the model. For EVT, the target region is cropped and resized to 224×224 to provide target-centric visual observations.

The predicted actions are converted into low-level motion commands and executed by the robot in a closed-loop manner. For safe real-world deployment, we employ a lightweight obstacle-aware control layer based on the onboard range sensors. The robot immediately stops when an obstacle is detected within 0.55 m, while its motion is progressively constrained when the obstacle distance falls below 1.0 m. This safety layer operates independently of the learned model and does not alter its predicted task objective.

D.2 EVALUATION SETUP

We evaluate VLN and EVT in previously unseen indoor environments using the same sensing stack, action interface, and safety layer described above. For each task, we construct three settings with progressively increasing interaction complexity.

VLN. The VLN evaluation examines instruction-following navigation at three levels. **V1** considers single-landmark navigation, where the robot navigates toward and stops near a specified landmark. **V2** introduces two-stage instructions that combine landmark navigation with a directional change, such as turning left or right before reaching the final goal. **V3** further considers longer compositional instructions involving multiple landmarks, directional relations, and intermediate navigation constraints. Representative instructions include "Go to the chair and stop beside it" for V1, "Move toward the chair, then turn left and stop beside the box" for V2, and "Pass the chair, turn right at the box, and stop near the door" for V3. For each setting, we construct ten distinct instruction–route pairs, each evaluated over five independent trials, resulting in 50 trials per setting. Each trial is limited to a maximum of 500 steps and is considered successful when the robot issues STOP within $d_{\text{stop}} = 1.0$ m of the instructed goal.

EVT. The EVT evaluation examines closed-loop target following under increasingly dynamic target motion. **V1** considers straight following in both open and constrained paths with obstacles. **V2** introduces target trajectories containing left or right turns. **V3** further introduces changes in target speed and stop-and-go motion, including slowing down, accelerating, stopping, and restarting. For each setting, we construct ten distinct instruction–route pairs, each evaluated over five independent trials, resulting in 50 trials per setting. In each trial, the target follows a predefined route with a maximum horizon of 500 steps. A trial is considered successful if the robot follows the designated target until it completes the predefined route within this horizon.

Evaluation Metrics. For VLN, we report Success Rate (SR), Final Distance (FD), and the number of collision events (Coll.). SR denotes the percentage of trials in which the robot successfully issues STOP within 1.0 m of the instructed goal. FD is the Euclidean distance between the robot's final position and the goal, averaged over all trials.

For EVT, we report Success Rate (SR), Target Retention Rate (TR), and the number of collision events (Coll.). SR denotes the percentage of trials in which the robot successfully follows the designated target until completion of the predefined route within the 500-step horizon. TR is computed as the fraction of successfully tracked time steps over all time steps across the evaluated trials, reflect-

Table 15: Real-world VLN results across progressively challenging instruction settings.

Setting	Method	SR (%) ↑	FD (m) ↓	#Coll. ↓
V1	Baseline	64.0	1.10	12
	WORLDWEAVE	**80.0**	**0.86**	**8**
V2	Baseline	48.0	1.56	21
	WORLDWEAVE	**70.0**	**1.12**	**12**
V3	Baseline	26.0	1.81	27
	WORLDWEAVE	**50.0**	**1.42**	**19**
Overall	Baseline	46.0	1.49	60
	WORLDWEAVE	**66.7**	**1.13**	**39**

Table 16: Real-world EVT results across progressively challenging target-motion settings.

Setting	Method	SR (%) ↑	TR (%) ↑	#Coll. ↓
V1	Baseline	70.0	88	14
	WORLDWEAVE	**78.0**	**92**	**7**
V2	Baseline	54.0	53	**27**
	WORLDWEAVE	**72.0**	**72**	28
V3	Baseline	22.0	57	37
	WORLDWEAVE	**48.0**	**66**	**15**
Overall	Baseline	48.7	-	78
	WORLDWEAVE	**66.0**	-	**50**

ing the ability to maintain the designated target throughout tracking. Coll. denotes the total number of collision events.

D.3 EXPERIMENTAL RESULTS

VLN. As shown in Table 15, WORLDWEAVE consistently outperforms the baseline across all three instruction settings. Overall, SR increases from 46.0% to 66.7%, while FD decreases from 1.49 m to 1.13 m and the number of collisions drops from 60 to 39. The SR improvement becomes more pronounced from V1 to V3, suggesting that the predictive context is particularly beneficial as instructions involve longer and more compositional navigation behaviors. These results demonstrate that the gains observed in simulation transfer to closed-loop navigation on a physical robot.

EVT. Table 16 shows that WORLDWEAVE also improves real-world target following across increasingly dynamic motion settings. Overall SR increases from 48.7% to 66.0%, while the total number of collisions decreases from 78 to 50. The improvement is particularly pronounced under V2 and V3, where target motion involves turning and varying speeds, respectively. WORLDWEAVE also achieves higher TR across all three settings, indicating more persistent target retention during tracking. Although V2 exhibits a slight increase in collisions (27 to 28), the overall results show improved target following under dynamic real-world interactions.

Qualitative Results. Figures 4 and 5 visualize representative closed-loop rollouts across the three difficulty levels. In VLN, WORLDWEAVE executes increasingly compositional behaviors from direct landmark approach to multi-stage navigation involving intermediate landmarks and turns. In EVT, the robot maintains target following from straight paths to turning and more dynamic interactions. Together with the quantitative results, these examples illustrate the model's ability to execute both instruction-guided navigation and target-conditioned tracking on a physical robot.

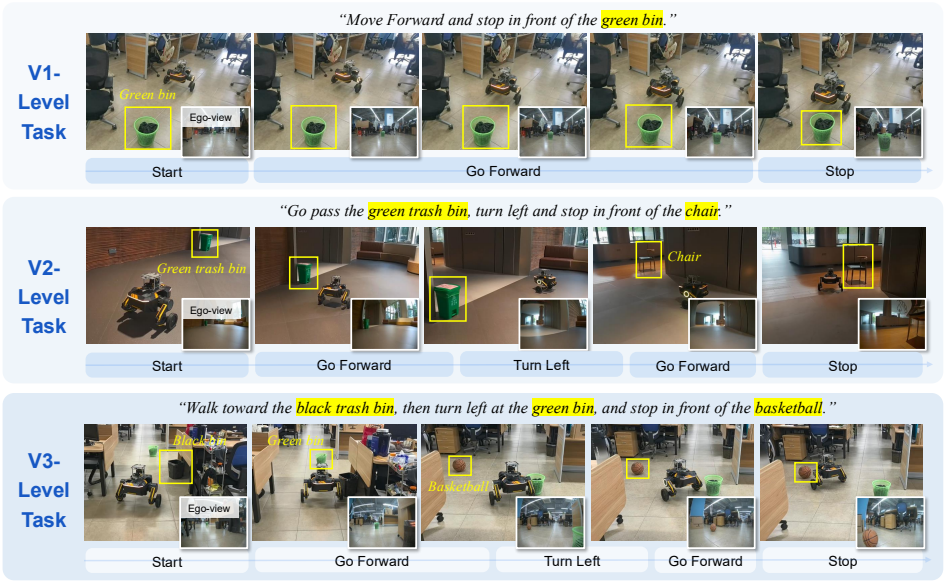

Figure 4: Representative real-world VLN rollouts under the V1–V3 instruction settings. Each sequence shows the robot's third-person execution together with the corresponding ego-view observations.

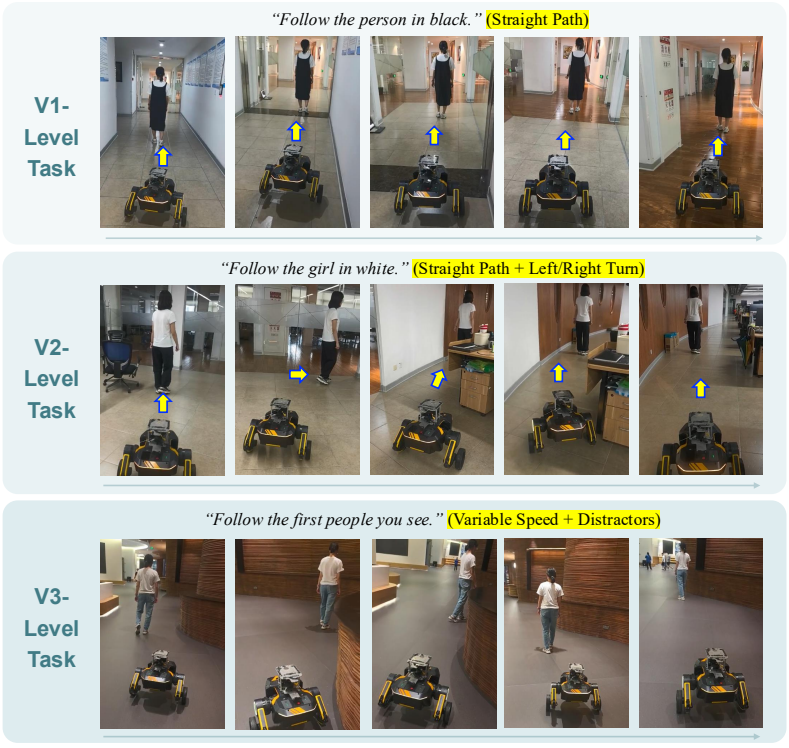

Figure 5: Representative real-world EVT rollouts under the V1–V3 target-motion settings, illustrating straight following, turning, and more dynamic target-following interactions.

E Qualitative Analysis

E.1 Vision-and-Language Navigation

Figure 6 presents qualitative comparisons between the task-specific baseline and WorldWeave on representative VLN episodes. Across diverse instructions involving multiple sequential navigation steps, the baseline deviates from the intended route or fails to reach the specified destination, whereas WorldWeave successfully follows the instruction to completion. These examples complement the quantitative results by showing how shared predictive dynamics, when adapted to VLN, can support long-horizon instruction following.

E.2 Embodied Visual Tracking

Figure 7 presents qualitative comparisons between the task-specific baseline and WorldWeave on representative EVT episodes. Under challenging target motions and interactions, the baseline may lose the designated target or fail to maintain stable tracking, whereas WorldWeave maintains target following over longer sequences. These examples complement the quantitative results by showing how shared predictive dynamics, when adapted to EVT, can support continuous agent–target interaction.



Figure 6: **Qualitative visualization of VLN.** Representative navigation trajectories of WORLD-WEAVE under language instructions.



Figure 7: **Qualitative EVT comparison.** Representative tracking trajectories of the task-specific baseline and WORLDWEAVE. WORLDWEAVE maintains more stable target following across challenging interactions.