IP-GRPO: WORLD-MODEL-GUIDED REINFORCEMENT POST-TRAINING FOR VISION-AND-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement post-training complements supervised fine-tuning with execution feedback that helps Vision-and-Language Navigation (VLN) policies correct errors after deviating from demonstrations. However, existing RL-based VLN methods often use trajectory-level rewards that provide limited guidance on whether intermediate decisions advance instruction completion. We introduce IP-GRPO, a world-model-guided reinforcement post-training framework that converts imagined futures into progress feedback at the action-chunk level. At each decision state, an action-conditioned world model predicts the short-term visual consequences of independently sampled auxiliary action chunks. An instruction-conditioned progress scorer evaluates current and imagined states using their respective textual and visual contexts. Averaging candidate-future scores yields a prospective state-progress estimate; subtracting the current-state score converts this estimate into a dense auxiliary reward. These auxiliary rewards are accumulated along execution trajectories and combined with task-success and trajectory-alignment rewards for Group Relative Policy Optimization (GRPO). All auxiliary modules operate only during post-training, preserving the policy's original inference procedure without future prediction or candidate search at deployment. Across the R2R-CE and RxR-CE Val-Unseen splits, IP-GRPO improves success rate by 4.83 percentage points on average over the Uni-NaVid and StreamVLN base policies. The post-trained Uni-NaVid also achieves a mean success rate of 76.7% across three zero-shot real-robot tasks.

1 INTRODUCTION

Vision-and-Language Navigation (VLN) requires an agent to continually perceive its environment and execute a sequence of actions following natural-language instructions. Policies built on pre-trained vision-language models typically learn from expert demonstrations through supervised fine-tuning (SFT) (Zhang et al., 2025b; Wei et al., 2026). However, fitting the conditional distribution of actions from successful demonstrations does not guarantee reliable task completion during autonomous execution. Learning from a fixed set of expert demonstrations is constrained by their quality and state coverage: prediction errors can lead the agent away from demonstrated trajectories, exposing it to out-of-distribution observations where supervision for correction and recovery is limited (Ross et al., 2011). Under partial observability, these errors can accumulate over long horizons. Reinforcement learning (RL) offers a natural route toward greater robustness by allowing agents to learn from the outcomes of their own interactions (Zhang et al., 2026c; Li et al., 2026b).

In real-world navigation, stepwise task progress is difficult to measure directly, while terminal success or failure provides limited feedback on intermediate decisions. Conducting RL directly in physical environments also requires repeated task execution, environment resets, and human supervision, incurring substantial time and labor costs. We therefore study reinforcement post-training of navigation policies by simulating real-world scenes. Terminal success rewards measure task completion, while trajectory-level path-alignment rewards measure overall route consistency; both provide limited guidance on whether intermediate decisions advance instruction completion (Li et al., 2026a). A central challenge is therefore to construct informative stepwise progress rewards and incorporate them into policy optimization.

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.

World models (WMs) (Hafner et al., 2025; Bar et al., 2025) strengthen environment-dynamics modeling and can potentially improve the sample efficiency of policy learning. Recent work applies World-Action Models (WAMs) to embodied navigation (Hu et al., 2025; Zhang et al., 2026b) by jointly learning future-state prediction and navigation-action generation. However, future prediction alone does not provide task-progress rewards for policy optimization. Moreover, VLN world-action models trained primarily through SFT still lack sufficient supervision for error correction and recovery after deviating from demonstrated trajectories. Beyond these training limitations, methods that explicitly generate future observations at deployment also incur additional inference costs. More fundamentally, prediction reliability presents another concern: robotic world models trained predominantly on successful demonstrations may exhibit success-biased future hallucination, predicting success even under incorrect actions (Peng et al., 2026; Liu et al., 2026). For language-conditioned prediction, this raises the concern that goal semantics may bias generated futures toward desired outcomes rather than the consequences of the supplied actions. Thus, converting predicted futures into useful learning feedback requires action-consistent prediction and instruction-conditioned progress evaluation.

To this end, we introduce Imagined Progress-guided Group Relative Policy Optimization (IP-GRPO), a navigation post-training framework that uses imagined progress to guide GRPO Shao et al. (2024). At each decision state, the policy samples an execution action chunk and independent auxiliary candidate chunks. A world model predicts short-term futures conditioned on visual observations and candidate actions, without directly receiving the language instruction. A progress model then evaluates the current state and candidate futures using the full instruction, visual context, and semantic descriptions. This separation aims to mitigate success-biased future hallucination. The difference between the mean candidate-future score and the current-state score forms an auxiliary reward, which complements success and path-alignment rewards from executed trajectories in GRPO updates. All auxiliary modules are used only during post-training, preserving the original policy's observation–action interface without additional future generation or candidate search at deployment. Our contributions are threefold:

- We introduce IP-GRPO, a world-model-guided reinforcement post-training framework for VLN that evaluates imagined futures to construct dense progress rewards for GRPO. The difference between mean candidate-future progress and current progress provides auxiliary feedback that complements task-level rewards.
- We demonstrate consistent improvements across Uni-NaVid Zhang et al. (2025b) and StreamVLN Wei et al. (2026) on R2R-CE and RxR-CE Val-Unseen, with an average success-rate gain of 4.8 percentage points and larger gains on long-horizon navigation tasks.
- We demonstrate zero-shot transfer to a real robot, where the post-trained Uni-NaVid achieves a mean success rate of 76.7% across three tasks. All auxiliary modules are confined to post-training, avoiding additional inference overhead from future prediction or candidate search.

2 Related Work

Vision-and-Language Navigation. Earlier VLN methods connect language understanding with navigation through cross-modal alignment, spatial maps, and explicit memory (Georgakis et al., 2022; Wang et al., 2023b). Recent pretrained vision-language models enable end-to-end policies that map instructions and egocentric video to actions, with multi-task training and temporal context modeling improving generalization and sequential decision-making (Zhang et al., 2024; 2025b; Wei et al., 2026). World-action models integrate future-state prediction with action generation for VLN (Hu et al., 2025; Zhang et al., 2026b). However, they primarily rely on SFT, with limited supervision for recovery from deviations beyond demonstration coverage, while the use of predicted futures as task-aware process rewards remains underexplored. We evaluate candidate futures for task progress and convert these evaluations into auxiliary rewards for reinforcement post-training, without modifying the policy architecture.

Reinforcement Fine-Tuning in Embodied Intelligence. Reinforcement fine-tuning (RFT) and related preference optimization methods use reward or preference feedback to improve pretrained models (Rafailov et al., 2023; Schulman et al., 2017). Group Relative Policy Optimization (GRPO) estimates advantages through within-group reward comparisons, eliminating the need for a separate critic (Shao et al., 2024). In navigation, prior work explores reinforcement fine-tuning based on

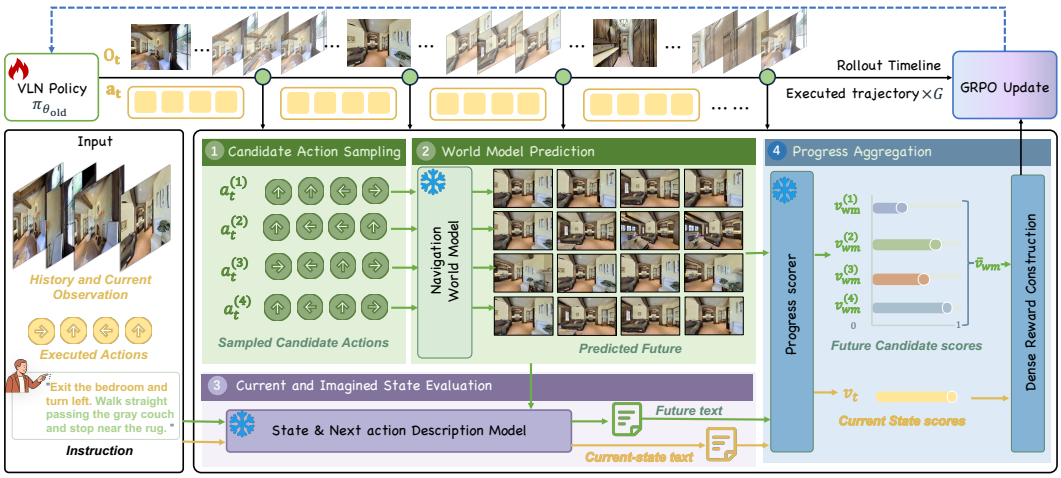

Figure 1: **Overview of IP-GRPO.** The policy executes action chunks and independently samples candidates. A frozen world model predicts candidate futures; description and progress models provide text and scores for current and imagined states. The mean imagined score minus the current score forms an auxiliary reward for GRPO alongside task rewards. Green marks imagined observations and text and scores; yellow marks observed history and current-state information.

agreement with expert actions (Qi et al., 2025; Wang et al., 2026a) and policy optimization using trajectories collected through environment interaction (Li et al., 2026b; Zhang et al., 2026c). SACA identifies valid prefixes and divergence points in executed trajectories to provide step-level process supervision (Li et al., 2026a). Beyond feedback from executed trajectories, another line of work uses world models to support policy learning through latent imagined trajectories (Hafner et al., 2025) and enable reward-driven post-training of pretrained VLA policies in video-based simulators (Zhu et al., 2025; Zhang et al., 2025a). Regulating imagined rollouts and jointly adapting world models and policies further improve training reliability (Jiang et al., 2026). Our framework instead uses imagined futures to provide auxiliary progress rewards for policy learning from navigation-simulator rollouts.

Progress Estimation and Imagined-Future Evaluation. Progress estimation aligns visual experience with instruction completion and supports action selection and backtracking (Ma et al., 2019a;b). Semantic progress reasoning further expresses completed instruction segments in language, providing context for action prediction and joint policy refinement (Wang et al., 2026a). Recent work combines future prediction with progress-value estimation or uses world-model temporal representations to learn value functions for robotic policy improvement (Yang et al., 2026; Wang et al., 2026b). We use pretrained Progress-Think components to describe current and imagined states, and a separate scorer to assess progress given these descriptions, the full instruction, and visual context.

3 METHOD

Figure. 1 summarizes IP-GRPO, which combines action-conditioned future prediction with instruction-conditioned progress evaluation for reinforcement post-training. At each decision state, the policy samples an execution action chunk and independent auxiliary chunks. The world model predicts future observations for the auxiliary chunks, semantic modules generate descriptions, and a progress scorer evaluates the current and imagined states. The mean imagined-future score minus the current-state score provides an auxiliary reward for GRPO.

3.1 TASK DEFINITION

We consider Vision-and-Language Navigation in Continuous Environments (VLN-CE). Given a natural-language instruction $\mathcal{I}$, an agent navigates from an initial pose using egocentric RGB observations, aiming to follow the described route, reach the goal, and stop correctly. At each decision

time t, a navigation policy π_θ conditions on the current observation o_t, a sampled history of past observations $\mathcal{O}_t$, and the instruction $\mathcal{I}$ to sample an action chunk of length K:

$$a_{t:t+K-1} \sim q_\theta^{\text{exec}}\left(\cdot \mid \mathcal{O}_t, o_t, \mathcal{I}\right), \tag{1}$$

where q_θ^{exec} denotes the execution sampling distribution induced by π_θ, and each primitive action belongs to $\mathcal{A} = \{\texttt{FORWARD}, \texttt{LEFT}, \texttt{RIGHT}, \texttt{STOP}\}$. Actions are executed sequentially in the continuous environment, ending early upon a `STOP` action or environment termination. If $\ell \leq K$ actions are executed and the episode remains active, the next decision occurs at $t + \ell$ using updated observations.

3.2 Action-Conditioned Future Prediction

At each decision time t, in addition to the action chunk actually executed in the environment, the policy $\pi_{\theta_{\text{old}}}$ independently samples M auxiliary candidates:

$$\widetilde{a}_{t:t+K-1}^{(m)} \overset{\text{iid}}{\sim} q_{\theta_{\text{old}}}^{\text{cand}}\left(\cdot \mid \mathcal{O}_t, o_t, \mathcal{I}\right), \quad m = 1, \ldots, M. \tag{2}$$

Candidate and execution sampling share policy parameters. Conditioned on the current visual context and instruction, the auxiliary candidates are independent of the execution sample. For each candidate, a Navigation World Model (NWM) Bar et al. (2025) predicts its short-term visual consequences:

$$\mathcal{O}_{\text{wm},t}^{(m)} = W_\psi\Big(\mathcal{O}_t, o_t, c\Big(\widetilde{a}_{t:t+K-1}^{(m)}\Big); \xi_t^{(m)}\Big). \tag{3}$$

Here, $\mathcal{O}_{\text{wm},t}^{(m)}$ contains H predicted observations spanning the candidate action chunk, $c(\cdot)$ maps navigation actions to the motion conditions required by NWM, and $\xi_t^{(m)}$ denotes generation noise. NWM takes only visual context and action-derived motion conditions as input, without directly receiving the language instruction; the instruction is instead used for subsequent progress evaluation. This separation aims to mitigate success-biased future hallucination. Future predictions are generated from actual observations at each decision step. The subscript wm denotes imagined quantities throughout. More Details are in App. A.3.

3.3 Instruction-Conditioned Progress Evaluation

We combine semantic reasoning with a scalar progress scorer to evaluate current and imagined successor states. Semantic modules generate descriptions for each state, and the scorer evaluates progress using the instruction and the corresponding visual and textual context.

Current and Imagined State Evaluation. At each decision time t, we use the progress reasoning module from Progress-Think Wang et al. (2026a) to generate a current progress description from the actual visual history and current observation. Its action module further incorporates the complete instruction to generate a next-action description. When no historical observations are available, the progress description is set to empty text.

Let $\mathcal{D}_t$ denote these textual descriptions and $\mathcal{U}(\mathcal{O}_t, o_t)$ denote the visual context obtained by uniformly sampling historical frames while retaining the current frame. We use Qwen3-VL-2B (Bai et al., 2025) with a LoRA adapter and a progress prediction head as the scorer, denoted by S. The current progress score is:

$$v_t = S(\mathcal{I}, \mathcal{D}_t, \mathcal{U}(\mathcal{O}_t, o_t)). \tag{4}$$

For each candidate action chunk m, NWM generates future observations $\mathcal{O}_{\text{wm},t}^{(m)}$, and the semantic reasoning modules produce updated textual descriptions $\mathcal{D}_{\text{wm},t+1}^{(m)}$ for the corresponding imagined successor state. The same scorer evaluates the imagined successor using these descriptions and the temporally concatenated actual and predicted visual context:

$$v_{\text{wm},t+1}^{(m)} = S\Big(\mathcal{I}, \mathcal{D}_{\text{wm},t+1}^{(m)}, \mathcal{U}(\mathcal{O}_t, o_t) \oplus \mathcal{O}_{\text{wm},t}^{(m)}\Big), \tag{5}$$

where $\oplus$ denotes temporal concatenation, $t+1$ indexes the imagined successor after the candidate action chunk, and wm marks quantities based on NWM predictions. Current and imagined progress scores lie in $[0, 1]$ and use the same scorer with their respective visual and textual contexts.

Scorer Training. We train the scorer on successful trajectories from the R2R-CE training split, using the full instruction, corresponding textual descriptions, and only actual observations available up to each labeled time step. We supervise the scorer with reference-trajectory progress, defined as the fraction of the reference path completed at the corresponding state. Normalized temporal progress t/T, where T is the total number of steps in the trajectory, increases monotonically with execution steps even when the agent fails to follow the instruction or move closer to the goal. In contrast, reference-trajectory progress measures spatial advancement along the specified route rather than elapsed execution steps. Specifically, let L_{ref} denote the total length of the reference trajectory and ℓ_t the path length along that trajectory from its start to the position corresponding to the current state. The training target is $y_t^{\mathrm{ref}} = \mathrm{clip}\left(\frac{\ell_t}{L_{\mathrm{ref}}}, 0, 1\right)$. Reference-trajectory progress labels are used only to train the scorer and are not used to directly annotate imagined candidates during reinforcement post-training. Further details are provided in App. A.3.

Imagined Progress Aggregation. During reinforcement post-training, the semantic reasoning modules and progress scorer remain frozen. The same scorer is applied to both actual observation contexts and contexts augmented with NWM-generated observations. We average the scores of the M candidate imagined successor states:

$$\bar{v}_{\mathrm{wm},t+1} = \frac{1}{M} \sum_{m=1}^{M} v_{\mathrm{wm},t+1}^{(m)}. \tag{6}$$

The candidate average estimates the expected progress after one action chunk from the current state under the auxiliary sampling distribution. Its difference from the current progress score defines the auxiliary reward in Eq. (7), which is recomputed at each subsequent real decision state.

3.4 Imagined-Progress Policy Optimization

We optimize the policy at the action-chunk level by combining imagined-progress feedback with actual task outcomes. For the same instruction, environment, and initial conditions, we collect G execution trajectories using the old policy. Trajectory indices are omitted below for brevity.

Reward Construction. Let t_j denote the starting time of action chunk j. We define the auxiliary reward as the difference between the candidate-averaged imagined progress and the current progress:

$$r_j^{\mathrm{IP}} = \beta_p \left(\bar{v}_{\mathrm{wm},t_j+1} - v_{t_j} \right). \tag{7}$$

Here, $\beta_p \geq 0$ controls the reward strength, and $t_j + 1$ indexes the imagined successor of chunk j. For each action chunk, we combine the subsequent cumulative auxiliary rewards, including the reward at that chunk, with the trajectory-level task reward. The resulting return-to-go in Eq. (9) is used for the group-relative policy update. We retain task-success and trajectory-alignment rewards:

$$R^{\mathrm{env}}(\tau) = \mathbf{1}_{\mathrm{success}}(\tau) + \lambda_{\mathrm{nDTW}}\, \mathrm{nDTW}(\tau, \tau^\star), \tag{8}$$

where τ and $\tau^\star$ are the executed and reference paths. The success reward evaluates goal arrival and correct stopping, while nDTW measures path consistency. For a trajectory containing J action chunks, the undiscounted return for chunk j is

$$\mathcal{R}_j = R^{\mathrm{env}}(\tau) + \sum_{u=j}^{J-1} r_u^{\mathrm{IP}}. \tag{9}$$

Group-Relative Policy Update. We use GRPO with process supervision Shao et al. (2024) at the action-chunk level. At each chunk index j, we normalize returns-to-go across group trajectories with a valid chunk:

$$\widehat{A}_j = \frac{\mathcal{R}_j - \mu_j}{\sigma_j + \epsilon}, \tag{10}$$

where μ_j and σ_j are the mean and population standard deviation of valid returns, and $\epsilon > 0$ ensures numerical stability. Ended trajectories are excluded from comparisons at subsequent indices; a single remaining trajectory has zero advantage. We apply an asymmetrically clipped GRPO objective to the

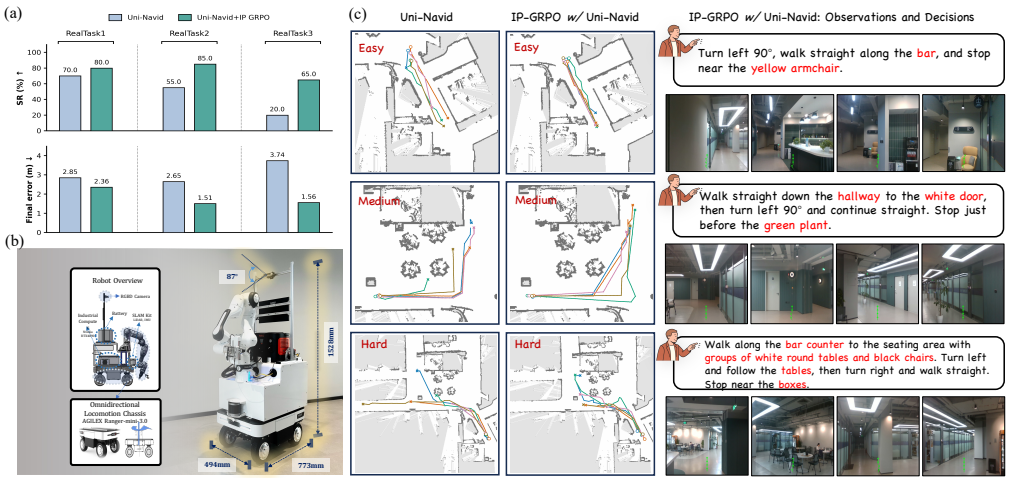

Figure 2: **Zero-shot real-world navigation with Uni-NaVid and IP-GRPO.** Left: robot platform, success rates, and mean final navigation errors over 20 trials per method on each of three tasks. Middle: executed trajectories. Right: task instructions and representative egocentric observations.

joint probability ratio of each full sampled action chunk, with KL regularization against a frozen SFT reference policy. Chunk probabilities include all K sampled actions, including any unexecuted suffix after early termination. Within each batch, old-policy probabilities, progress scores, and advantages remain fixed. Only the navigation policy is updated; all auxiliary modules remain frozen. The full optimization objective is in App. A.1.

4 Experiments

4.1 Experimental Setup

We evaluate IP-GRPO on the R2R-CE and RxR-CE benchmarks for VLN in continuous environments. Success Rate (SR) and SPL are used as the primary metrics, while Navigation Error (NE), Oracle Success Rate (OSR), and nDTW are also reported. We apply post-training to Uni-NaVid and StreamVLN, using NWM to predict candidate futures, pretrained Progress-Think components to generate semantic descriptions, and a frozen Qwen3-VL-2B-based progress scorer to evaluate current and imagined states. The action-chunk length, number of auxiliary candidates, number of predicted frames, and GRPO group size are all set to four, i.e., $K = M = H = G = 4$. Post-training updates the policy language model and multimodal projector while keeping the vision encoder and progress scorer frozen. We use AdamW with ZeRO-1 and a learning rate of 3×10^{-6}. The weights of the success, nDTW, and imagined-progress rewards are set to 1, 0.3, and 0.5, respectively, with $\gamma = 1$. The policy-ratio clipping range is $[0.8, 1.28]$, and the KL coefficient is 0.05. We use a fixed random seed of 42 for all evaluations conducted in this work. More details are provided in App. A.3.

4.2 Main Results

Table 1 reports results across two backbones and two benchmarks. IP-GRPO consistently reduces navigation error and improves success rate and path efficiency in all four settings, with an average SR gain of 4.83 percentage points, demonstrating benefits across policies and datasets. On RxR-CE, which features longer instructions and more complex routes, IP-GRPO w/ Uni-NaVid improves SPL by 8.1 percentage points. This gain may stem from supplementing trajectory-level rewards with prospective feedback at intermediate states: accumulating this feedback in trajectory returns may encourage sustained progress and reduce deviations and detours over long horizons. Without additional external post-training data, IP-GRPO outperforms methods using extra training data and representative WAM-based methods in selected settings. It uses imagined futures to construct instruction-conditioned progress rewards only during post-training, preserving the original inference procedure without auxiliary modules or candidate search at deployment.

Table 1: R2R-CE/RxR-CE Val-Unseen results. Observations: panoramic RGB (Pano.), odometry (Odo.), depth, single-view RGB (S.RGB). Units: meters for NE; percentages otherwise. Δ: absolute change from the base policy. †: reported VLN-CE expert-only setting, without extra task-specific data. "–": unreported. *: StreamVLN's external navigation data is EnvDrop only. ^{a}: trajectory counts. ^{b}: 500K DAgger samples plus unspecified ScaleVLN data. ^{c}: no extra external post-training data, excluding base-policy data.

Method	VLM Params	Observation				R2R-CE Val-Unseen				RxR-CE Val-Unseen				External Training Data
		Pano.	Odo.	Depth	S.RGB	NE↓	OSR↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑	
HPN+DN Krantz et al. (2021)	–	✓	✓	✓		6.31	40.0	36.0	34.0	–	–	–	–	–
CMA Hong et al. (2022)	–	✓	✓	✓		6.20	52.0	41.0	36.0	8.76	26.5	22.1	47.0	–
VLN↻BERT Hong et al. (2022)	–	✓	✓	✓		5.74	53.0	44.0	39.0	8.98	27.0	22.6	46.7	–
Ego^2-Map Hong et al. (2023)	–	✓	✓	✓		5.54	56.0	47.0	41.0	–	–	–	–	–
DreamWalker Wang et al. (2023a)	–	✓	✓	✓		5.53	59.0	49.0	44.0	–	–	–	–	–
GridMM Wang et al. (2023b)	–	✓	✓	✓		5.11	61.0	49.0	41.0	–	–	–	–	–
InstructNav Long et al. (2025)	–	✓	✓	✓		6.89	–	31.0	24.0	–	–	–	–	–
COSMO Zhang et al. (2025d)	–	✓				–	56.0	47.0	40.0	–	–	–	–	–
AO-Planner Chen et al. (2025)	–	✓		✓		5.55	59.0	47.0	33.0	7.06	43.3	30.5	50.1	–
LAW Raychaudhuri et al. (2021)	–		✓	✓	✓	6.83	44.0	35.0	31.0	10.90	8.0	8.0	38.0	–
MapNav Zhang et al. (2025c)	–		✓	✓	✓	4.93	53.0	39.7	37.2	–	–	–	–	–
g3D-LF Wang & Lee (2025)	–		✓	✓	✓	5.70	59.5	47.2	34.6	–	–	–	–	–
Seq2Seq Krantz et al. (2020)	–			✓	✓	7.77	37.0	25.0	22.0	12.10	13.9	11.9	30.8	–
NaVid-4D Liu et al. (2025)	–			✓	✓	5.99	55.7	43.8	37.1	–	–	–	–	–
NavMorph Yao et al. (2025)	–			✓	✓	5.75	56.9	47.9	33.2	8.85	30.8	22.8	44.2	–
NaVid Zhang et al. (2024)	7B				✓	5.47	49.1	37.4	35.9	–	–	–	–	953K
Sim2Real Wang et al. (2025)	–				✓	5.95	55.8	44.9	30.4	8.79	25.5	18.1	–	0K
InternVLA-N1 Intern Robotics (2025)	8B				✓	4.89	60.6	55.4	52.1	6.41	49.5	41.8	62.6	51,200K
NaVILA Cheng et al. (2025)	7B				✓	5.22	62.5	54.0	49.0	6.77	49.3	44.0	58.8	13,132K
FutureNav-4B† Zhang et al. (2026b)	4B				✓	5.13	61.8	55.1	50.1	5.62	54.5	46.0	61.8	0K
FutureNav-8B† Zhang et al. (2026b)	8B				✓	5.15	61.6	55.5	51.4	5.93	53.3	45.4	59.8	0K
NavFoM Zhang et al. (2026a)	7B				✓	5.01	64.9	56.2	51.2	5.51	57.4	49.4	60.2	∼10,480K
VLN-R1 Qi et al. (2025)	7B				✓	7.00	41.2	30.2	21.8	10.40	22.3	17.5	-	0K
ActiveVLN Zhang et al. (2026c)	3B				✓	5.31	58.3	50.1	43.7	5.84	50.7	41.2	58.1	146.3K^{a}
SACA Li et al. (2026a)	8B				✓	4.57	64.9	60.3	55.1	4.90	60.3	49.8	62.1	0K
Progress-Think Wang et al. (2026a)	4B				✓	4.68	63.6	60.1	53.6	–	–	–	–	≥500K^{b}
JanusVLN† Zeng et al. (2026)	7B				✓	5.17	58.0	52.8	49.2	6.46	51.4	44.3	59.1	0K
JanusVLN Zeng et al. (2026)	7B				✓	4.78	65.2	60.5	56.8	6.06	56.2	47.5	62.1	10,692K
Uni-NaVid Zhang et al. (2025b)	7B				✓	5.21	55.2	51.8	46.5	5.67	56.1	44.5	59.2	3,577K
StreamVLN* Wei et al. (2026)	7B				✓	6.05	53.8	45.5	41.6	–	–	–	–	10,033K
StreamVLN Wei et al. (2026)	7B				✓	4.98	64.7	56.3	50.2	6.22	52.9	46.0	61.9	∼26,330K
IP-GRPO *w/* Uni-NaVid	**7B**				✓	**4.78**	**60.1**	**55.8**	**51.7**	**4.85**	**60.9**	**52.6**	**62.3**	0K^{c}
Δ *vs. Uni-NaVid*						*-0.43*	*+4.9*	*+4.0*	*+5.2*	*-0.82*	*+4.8*	*+8.1*	*+3.1*	
IP-GRPO *w/* StreamVLN	**7B**				✓	**4.31**	**67.9**	**61.8**	**58.5**	**5.29**	**57.9**	**49.6**	**66.6**	0K^{c}
Δ *vs. StreamVLN*						*-0.67*	*+3.2*	*+5.5*	*+8.3*	*-0.93*	*+5.0*	*+3.6*	*+4.7*	

Table 2: Performance across navigation distances. Share denotes the episode percentage within each dataset. SR is reported as a percentage, and ΔSR denotes the gain in percentage points.

Distance	R2R-CE				RxR-CE			
	Share (%)	Uni-NaVid	IP-GRPO *w/* Uni-NaVid	ΔSR	Share (%)	Uni-NaVid	IP-GRPO *w/* Uni-NaVid	ΔSR
<5 m	3.4	76.2	77.8	+1.6	16.0	76.8	80.3	+3.5
5–10 m	66.9	55.7	60.1	+4.4	29.3	63.6	70.0	+6.4
10–15 m	26.9	41.5	44.2	+2.7	27.3	51.8	56.7	+4.9
15–20 m	2.6	29.1	37.5	+8.4	15.0	44.4	45.2	+0.8
20–30 m	0.2	0.0	33.3	+33.3	11.0	35.2	42.4	+7.2
≥ 30 m	–	–	–	–	1.3	36.8	43.9	+7.1
Overall	**100.0**	**51.8**	**55.8**	**+4.0**	**100.0**	**56.1**	**60.9**	**+4.8**

Performance across Navigation Distances. We group evaluation episodes by navigation distance on R2R-CE and RxR-CE. As shown in Table 2, IP-GRPO improves Uni-NaVid's SR across all distance intervals on both datasets, including substantial gains in the relatively well-populated 20–30 m interval of RxR-CE, supporting its effectiveness for longer-distance navigation. These gains may reflect the value of intermediate feedback on longer routes, where more sequential decisions allow errors to accumulate and trajectory-level rewards provide limited guidance for individual decisions. By providing auxiliary feedback at intermediate states, IP-GRPO may help the policy learn behaviors that sustain progress over longer trajectories.

Zero-shot Real-World Evaluation. We deploy Uni-NaVid and its counterpart post-trained with IP-GRPO on R2R-CE on a real robot for zero-shot evaluation across three navigation tasks of in-

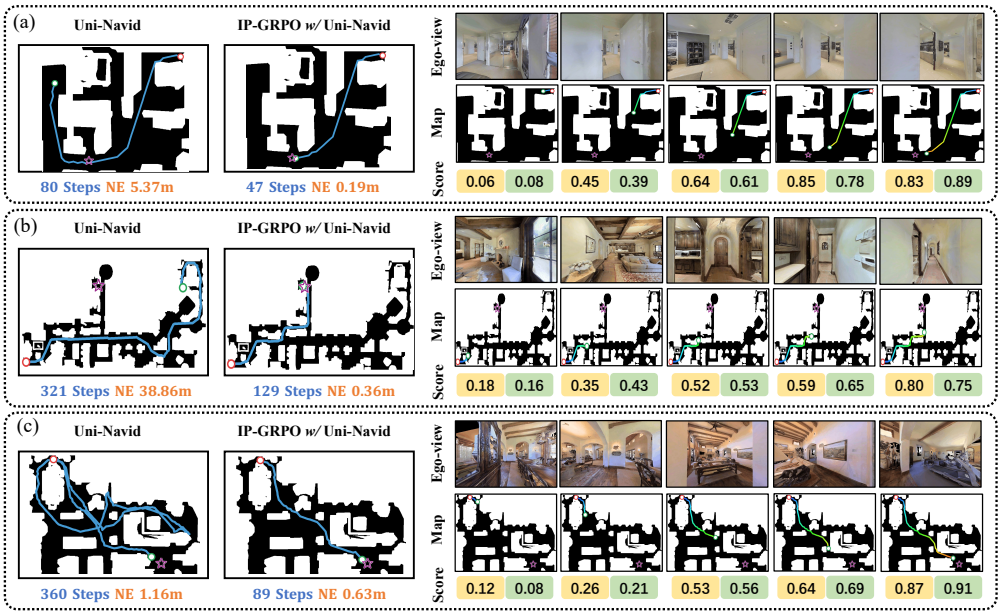

Figure 3: **Navigation examples on RxR-CE.** Red circles mark starts, green circles current or final positions, purple stars goals, and blue lines executed paths. Right: selected egocentric views, maps, and progress scores (yellow: model predictions; green: reference-trajectory progress labels).

creasing difficulty: easy, medium, and hard. Each method is evaluated over 20 trials per task, reporting success rate and mean final navigation error. As shown in Figure 2, IP-GRPO achieves success rates of 80%, 85%, and 65%, improving over Uni-NaVid by 10, 30, and 45 percentage points, respectively. On the hard task, which requires following multiple landmarks and executing successive turns, mean final navigation error decreases from 3.74 m to 1.56 m. These results indicate that gains from simulation-based post-training transfer to the evaluated real-world tasks, with larger success-rate improvements on more complex tasks. Figure 2 also presents representative trajectories and egocentric observations.

Case Study. Figure 3 presents three RxR-CE examples illustrating differences in stopping, route following, and execution efficiency. In case (a), Uni-NaVid approaches the goal but continues moving away, whereas IP-GRPO stops nearby. In case (b), Uni-NaVid deviates from the route, while IP-GRPO reaches the goal. In case (c), both methods succeed, but IP-GRPO reduces detours and the number of execution steps from 360 to 89. These examples are consistent with the lower navigation error and improved path efficiency observed in the main results.

4.3 Ablation Studies

We conduct ablations with Uni-NaVid on R2R-CE Val-Unseen. All GRPO variants retain success and nDTW rewards. *w/o RL Post-training* is the original-policy baseline, while *w/o Progress Reward* isolates the benefit of task-level RL post-training. Both world-model-free variants share the same frozen progress scorer and use the complete instruction and actual historical and current observations, without generating future images. *w/o WM (Direct Scoring)* scores one set of current-progress and next-action descriptions from the standard generation procedure, testing progress feedback without future observations. *w/o WM (Sampled Scoring)* instead samples multiple sets of these descriptions and averages their scores, with all other settings unchanged, to assess the benefit of text sampling and score averaging. Both directly use progress scores as auxiliary rewards without differencing. Full IP-GRPO predicts candidate futures and constructs the auxiliary reward by subtracting the current score from the mean predicted-future score.

As shown in Table 3, task-reward-only GRPO yields modest gains. Direct scoring provides no further improvement, while sampled scoring offers limited additional benefits. Full IP-GRPO improves SR/SPL by 2.2/1.8 percentage points over sampled scoring and 2.5/2.5 points over task-reward-only

Table 3: Framework ablations on R2R-CE Val-Unseen using Uni-NaVid.

Method	NE↓	OSR↑	SR↑	SPL↑
w/o RL Post-training	5.21	55.2	51.8	46.5
w/o Progress Reward	5.00	58.3	53.3	49.2
w/o WM (Direct Scoring)	5.06	57.6	52.9	49.0
w/o WM (Sampled Scoring)	4.95	57.4	53.6	49.9
IP-GRPO *w/* Uni-NaVid (Full)	**4.78**	**60.1**	**55.8**	**51.7**

Table 4: Progress scorer ablations. All variants are evaluated against the same reference-trajectory progress labels on successful and failed trajectories. Spearman's ρ is computed over all steps pooled within each evaluation subset, rather than averaged across trajectories. Inputs contain no future observations.

	Successful trajectories		Failed trajectories	
Variant	MAE ↓	ρ ↑	MAE ↓	ρ ↑
Full input, reference-trajectory supervision	**0.0850**	**0.9149**	**0.2106**	**0.3831**
w/o progress description	0.0924	0.8961	0.2392	0.2652
w/o next-action text	0.1003	0.8812	0.2459	0.2316
Current frame only	0.1464	0.7937	0.2798	0.3063
Text only	0.1607	0.7264	0.2650	0.2523
Full input, t/T supervision	0.1758	0.8512	0.4050	0.2113

GRPO. These results support combining future-state feedback with progress-difference rewards to improve task completion and path efficiency.

Progress Scorer Ablations. We examine the effects of semantic descriptions, visual history, and supervision targets. All variants share the same training setup and are evaluated against the same reference-trajectory progress labels on successful and failed trajectories. Full inputs comprise the complete instruction, up to nine actual RGB frames including the current frame, and progress and next-action descriptions generated during data collection, without future observations. As shown in Table 4, removing either semantic description increases error and reduces correlation, with a larger degradation from removing next-action text. Using only the current frame or text also degrades performance, supporting the joint use of visual history and semantic information. When evaluated against reference-trajectory progress, reference-trajectory supervision yields lower error and higher correlation than temporal-progress supervision, particularly on failed trajectories. The full model performs best on both subsets, although its substantially lower correlation on failed trajectories indicates limited generalization to failed executions.

World Model Prediction Quality. We evaluate world-model predictions through image quality and candidate-ranking consistency. Fine-tuning improves image quality across seen and unseen environments in both generation modes. In unseen environments, LPIPS decreases by 44.7% and 41.6%, while PSNR increases by 4.58 and 4.20 dB in Time and Rollout modes, respectively. With the policy and progress scorer frozen, candidate pairwise ranking accuracy improves from 56.02% to 64.94% on the evaluated R2R-CE val-unseen subset. These results show that fine-tuning improves visual fidelity and better preserves the scorer's relative assessments of actual candidate futures. Details are in App. A.4.

5 Conclusion

We presented IP-GRPO, a world-model-guided reinforcement post-training framework for VLN. By combining action-conditioned future prediction with instruction-conditioned progress evaluation, IP-GRPO constructs auxiliary rewards from the difference between candidate-averaged imagined progress and current progress, supplementing task-level rewards with prospective feedback. Experiments on R2R-CE and RxR-CE show consistent improvements across Uni-NaVid and StreamVLN, while real-robot experiments demonstrate zero-shot transfer benefits on the evaluated tasks. All auxiliary modules operate only during post-training, preserving the policy's original inference procedure without future prediction or candidate search at deployment.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting, language polishing, and improving clarity. We reviewed and revised all AI-assisted text and take full responsibility for the final manuscript.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation World Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15791–15801, 2025.

Jiaqi Chen, Bingqian Lin, Xinmin Liu, Lin Ma, Xiaodan Liang, and Kwan-Yee K. Wong. Affordances-oriented planning using foundation models for continuous vision-language navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23568–23576, 2025.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged Robot Vision-Language-Action Model for Navigation. In *Proceedings of Robotics: Science and Systems*, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15460–15470, 2022.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15439–15449, 2022.

Yicong Hong, Yang Zhou, Ruiyi Zhang, Franck Dernoncourt, Trung Bui, Stephen Gould, and Hao Tan. Learning Navigational Visual Representations with Semantic Map Supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3055–3067, 2023.

Junjun Hu, Jintao Chen, Haochen Bai, Minghua Luo, Shichao Xie, Ziyi Chen, Fei Liu, Zedong Chu, Xinda Xue, Botao Ren, Xiaolong Wu, Mu Xu, and Shanghang Zhang. AstraNav-World: World model for foresight control and consistency. arXiv preprint arXiv:2512.21714v1, 2025.

Intern Robotics. InternVLA-N1: An Open Dual-System Vision-Language Navigation Foundation Model with Learned Latent Plans. Technical report, Shanghai Artificial Intelligence Laboratory, 2025.

Zhennan Jiang, Shangqing Zhou, Yutong Jiang, Zefang Huang, Mingjie Wei, Yuhui Chen, Tianxing Zhou, Zhen Guo, Hao Lin, Quanlu Zhang, Yu Wang, Haoran Li, Chao Yu, and Dongbin Zhao. WoVR: World Models as Reliable Simulators for Post-Training VLA Policies with RL. arXiv preprint arXiv:2602.13977, 2026.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the Nav-Graph: Vision-and-Language Navigation in Continuous Environments. In *Computer Vision – ECCV 2020*, volume 12373 of *Lecture Notes in Computer Science*, pp. 104–120. Springer, 2020.

Jacob Krantz, Aaron Gokaslan, Dhruv Batra, Stefan Lee, and Oleksandr Maksymets. Waypoint Models for Instruction-Guided Navigation in Continuous Environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15162–15171, 2021.

Haoyuan Li, Rui Liu, Hehe Fan, and Yi Yang. Let's Reward Step-by-Step: Step-Aware Contrastive Alignment for Vision-Language Navigation in Continuous Environments. In *Computer Vision – ECCV 2026*, volume 17063 of *Lecture Notes in Computer Science*, pp. 482–500. Springer, 2026a.

Jiangyang Li, Cong Wan, SongLin Dong, Chenhao Ding, Qiang Wang, Zhiheng Ma, and Yihong Gong. Trajectory-Diversity-Driven Robust Vision-and-Language Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9143–9154, 2026b.

Haoran Liu, Weikang Wan, Xiqian Yu, Minghan Li, Jiazhao Zhang, Bo Zhao, Zhibo Chen, Zhongyuan Wang, Zhizheng Zhang, and He Wang. NaVid-4D: Unleashing spatial intelligence in egocentric RGB-D videos for vision-and-language navigation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

Xiaokang Liu, Zechen Bai, Hai Ci, Kevin Yuchen Ma, and Mike Zheng Shou. World-VLA-Loop: Closed-loop learning of video world model and VLA policy. arXiv preprint arXiv:2602.06508, 2026.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. InstructNav: Zero-shot System for Generic Instruction Navigation in Unexplored Environment. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2049–2060. PMLR, 2025.

Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-Monitoring Navigation Agent via Auxiliary Progress Estimation. In *International Conference on Learning Representations*, 2019a.

Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The Regretful Agent: Heuristic-Aided Navigation Through Progress Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6732–6740, 2019b.

Quanquan Peng, Yutong Liang, Rui Yan, Nicklas Hansen, and Xiaolong Wang. FACT: Failure-aware causal training for world-action models. arXiv preprint arXiv:2608.10232, 2026.

Zhangyang Qi, Zhixiong Zhang, Yizhou Yu, Jiaqi Wang, and Hengshuang Zhao. VLN-R1: Vision-language navigation via reinforcement fine-tuning. arXiv preprint arXiv:2506.17221, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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 *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4018–4028, 2021.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. arXiv preprint arXiv:1707.06347, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Hanqing Wang, Wei Liang, Luc Van Gool, and Wenguan Wang. DREAMWALKER: Mental planning for continuous vision-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10873–10883, 2023a.

Shuo Wang, Yucheng Wang, Guoxin Lian, Yongcai Wang, Maiyue Chen, Kaihui Wang, Bo Zhang, Zhizhong Su, Yutian Zhou, Wanting Li, Deying Li, and Zhaoxin Fan. Progress-Think: Semantic Progress Reasoning for Vision-Language Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4076–4086, 2026a.

Zhihao Wang, Jianxiong Li, Yu Cui, Yuan Gao, Xianyuan Zhan, Junzhi Yu, and Xiao Ma. World value models for robotic manipulation. arXiv preprint arXiv:2606.24742, 2026b.

Zihan Wang and Gim Hee Lee. g3D-LF: Generalizable 3D-language feature fields for embodied tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14191–14202, 2025.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. GridMM: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15625–15636, 2023b.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Sim-to-Real Transfer via 3D Feature Fields for Vision-and-Language Navigation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2982–2995. PMLR, 2025.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, Xihui Liu, and Jiangmiao Pang. StreamVLN: Streaming Vision-and-Language Navigation via SlowFast Context Modeling. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Jiazhi Yang, Kunyang Lin, Jinwei Li, Wencong Zhang, Tianwei Lin, Longyan Wu, Zhizhong Su, Hao Zhao, Ya-Qin Zhang, Li Chen, Ping Luo, Xiangyu Yue, and Hongyang Li. RISE: Self-improving robot policy with compositional world model. arXiv preprint arXiv:2602.11075, 2026.

Xuan Yao, Junyu Gao, and Changsheng Xu. NavMorph: A Self-Evolving World Model for Vision-and-Language Navigation in Continuous Environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5536–5546, 2025.

Shuang Zeng, Dekang Qi, Xinyuan Chang, Feng Xiong, Shichao Xie, Xiaolong Wu, Shiyi Liang, Mu Xu, and Xing Wei. JanusVLN: Decoupling Semantics and Spatiality with Dual Implicit Memory for Vision-Language Navigation. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

Jiahui Zhang, Ze Huang, Chun Gu, Zipei Ma, and Li Zhang. Reinforcing Action Policies by Prophesying. arXiv preprint arXiv:2511.20633, 2025a.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. NaVid: Video-based VLM Plans the Next Step for Vision-and-Language Navigation. In *Proceedings of Robotics: Science and Systems*, 2024.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-NaVid: A Video-based Vision-Language-Action Model for Unifying Embodied Navigation Tasks. In *Proceedings of Robotics: Science and Systems*, 2025b.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, Yuxin Fan, Wenjun Li, Zhibo Chen, Fei Gao, Qi Wu, Zhizheng Zhang, and He Wang. Embodied navigation foundation model. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026a.

Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. MapNav: A novel memory representation via annotated semantic maps for VLM-based vision-and-language navigation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13032–13056, 2025c.

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, 2026b.

Siqi Zhang, Yanyuan Qiao, Qunbo Wang, Zike Yan, Qi Wu, Zhihua Wei, and Jing Liu. COSMO: Combination of Selective Memorization for Low-cost Vision-and-Language Navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5511–5522, 2025d.

Zekai Zhang, Weiye Zhu, Hewei Pan, Xiangchen Wang, Rongtao Xu, Xing Sun, and Feng Zheng. ActiveVLN: Towards Active Exploration via Multi-Turn RL in Vision-and-Language Navigation. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026c.

Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. WMPO: World model-based policy optimization for vision-language-action models. arXiv preprint arXiv:2511.09515, 2025.

A Appendix

A.1 Action-Chunk-Level Policy Optimization

Let $x_j = (\mathcal{O}_{t_j}, o_{t_j}, \mathcal{I})$ denote the policy input for chunk j, and let $a_j = a_{t_j:t_j+K-1}$ denote the full sampled action chunk. Its joint probability is the product of the conditional probabilities of its constituent actions. The corresponding new-to-old policy probability ratio is

$$\rho_j(\theta) = \frac{\pi_\theta(a_j \mid x_j)}{\pi_{\theta_{\text{old}}}(a_j \mid x_j)}. \tag{11}$$

Using the group-relative advantage $\widehat{A}_j$ defined in Eq. 10, we optimize

$$\begin{aligned} \mathcal{L} = & -\mathbb{E}_j\Big[\min\Big(\rho_j(\theta)\widehat{A}_j,\ \text{clip}\left(\rho_j(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\right)\widehat{A}_j\Big)\Big] \\ & + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}. \end{aligned} \tag{12}$$

Here, the expectation is taken over valid action chunks in the collected trajectories. The parameters ϵ_{low} and ϵ_{high} control the lower and upper clipping bounds, respectively. The term $\mathcal{L}_{\text{KL}}$ regularizes the policy against a frozen SFT reference policy, with strength controlled by λ_{KL}.

A.2 World-Model Ablation Details

Table 5 summarizes the visual inputs, text generation, and auxiliary reward construction of the three variants. All variants use the same frozen progress scorer, retain success and nDTW rewards, and apply the same auxiliary reward coefficient β_p.

The two world-model-free variants differ only in text sampling and score aggregation. Both evaluate the current state using the complete instruction and actual historical and current observations. Direct Scoring uses one standard set of progress and next-action descriptions, whereas Sampled Scoring samples multiple sets and averages their scores. These descriptions are generated from the current observation context and do not provide future observations. Both variants directly reward progress scores without differencing. Full IP-GRPO instead evaluates predicted candidate futures and subtracts the current score from their mean progress score.

Table 5: Configurations of the world-model ablations. The auxiliary signals listed below are multiplied by the same coefficient β_p. All scoring inputs include the complete instruction.

Variant	Visual input	Text generation	Auxiliary signal
w/o WM (Direct Scoring)	Actual history and current frame	One standard description set	Current-state progress score
w/o WM (Sampled Scoring)	Actual history and current frame	Multiple sampled description sets	Mean current-state progress score
Full IP-GRPO	Actual context and predicted candidate futures	Descriptions for current and imagined states	Mean imagined progress minus current progress

A.3 Experiment Setup Details

Data and Evaluation. We perform post-training separately on R2R-CE and RxR-CE. R2R-CE contains 10,819 training episodes across 61 scenes and 1,839 Val-Unseen episodes across 11 scenes. Both backbones use a 0.25 m forward step, with turning angles of 30° for Uni-NaVid and 15° for StreamVLN. Episodes are capped at 150 actions during post-training and 500 during evaluation. All simulation-based navigation evaluations use a 3 m success-distance threshold and a sampling temperature of 0.2. All evaluations use a fixed random seed of 42.

World Model. NWM uses the CDiT-XL/2 architecture and is fine-tuned on Habitat-rendered data from the R2R-CE training set, comprising 10,819 navigation trajectories with an image resolution of 224×224. Since Uni-NaVid and StreamVLN use turning angles of 30° and 15°, respectively, we generate training data under their respective action configurations and fine-tune a corresponding NWM for each policy. For Uni-NaVid, NWM is initialized from pretrained weights and trained using AdamW with a learning rate of 5×10^{-5} and BF16 precision. We use the exponential moving average (EMA) weights after 90,000 optimization steps. For StreamVLN, this checkpoint is further fine-tuned on data generated under its action configuration.

During RL post-training, NWM remains frozen. It conditions on the four most recent actual observations and predicts four future frames using Time mode with 25 denoising steps. Candidate action prefixes are converted into cumulative displacements and rotations in the local coordinate frame to serve as motion conditions.

Progress Scorer. The progress scorer is built on Qwen3-VL-2B and trained on step-wise navigation records from the R2R-CE training set. Each sample contains up to nine visual observations, together with the full navigation instruction, a description of the completed instruction segments, and a description of the next action. Images are preprocessed using the native Qwen3-VL processor. We mean-pool the final-layer hidden states over valid tokens and feed the resulting representation into a two-layer MLP regression head, comprising a linear layer, GELU, dropout with probability 0.1, a second linear layer, and a sigmoid activation that constrains the output to $[0, 1]$. The supervision target is the reference-trajectory progress defined in Section 3.3. The scorer is trained on successful trajectories using Smooth-L1 loss with $\beta = 0.1$.

Training begins by freezing the backbone and optimizing only the regression head, followed by joint optimization of the LoRA parameters and the head. LoRA is applied to the attention q, k, v, and o projections, with rank $r = 16$, scaling parameter $\alpha = 32$, and dropout 0.05. We use AdamW and load the backbone in BF16 precision. To reduce reliance on the description of completed instruction segments, we replace this field with an empty string with probability 0.2 during training. Table 6 summarizes the settings for head warm-up and successful-trajectory fine-tuning.

Validation is performed on the corresponding R2R-CE val-seen data, using at most 500 samples per validation pass. The best checkpoint is selected by the lowest validation mean absolute error (MAE), and subsequent fine-tuning is initialized from the best checkpoint of the preceding stage. The scorer remains frozen throughout policy post-training.

Table 6: Settings for progress-scorer head warm-up and successful-trajectory fine-tuning. Iteration counts refer to sample iterations rather than optimizer updates.

Setting	Head warm-up	Fine-tuning
Training trajectories	Successful	Successful
Trainable parameters	Head	LoRA + head
Head learning rate	1×10^{-3}	1×10^{-4}
LoRA learning rate	—	1×10^{-4}
Sample iterations	5,000	10,000
Microbatch size per GPU	1	1
Gradient accumulation	16	4
Validation interval	500	500

Policy Optimization. Each task produces four simulator trajectories, with four actions per chunk and four independently sampled auxiliary candidates per decision state. Execution and candidate sampling use temperatures of 0.2 and 1.0, respectively. Sampling applies temperature-scaled softmax over the four valid action tokens without nucleus truncation. Old, new, and reference policy likelihoods use the same restricted action distribution and execution temperature.

We use AdamW with ZeRO-1, a learning rate of 3×10^{-6}, a clipping range of $[0.8, 1.28]$, and a KL coefficient of 0.05. Success, nDTW, and imagined-progress rewards have weights of 1, 0.3, and 0.5, respectively, with no discounting. Each rollout batch is traversed once. Samples are grouped by task and action-chunk index, with group sizes decreasing as trajectories terminate. Only the navigation policy is updated; all auxiliary modules remain frozen.

Real-Robot Evaluation. We deploy the Uni-NaVid policy post-trained on R2R-CE to a mobile robot equipped with an RTX 4090 GPU for zero-shot evaluation. The robot uses a 0.25 m forward step and 30° turns, with four actions predicted per decision. Each method is evaluated over 20 trials on each of three navigation tasks, reporting success rate and mean final navigation error. The deployed policy uses approximately 17 GB of GPU memory on an NVIDIA RTX 4090.

A.4 World Model Prediction Quality

We evaluate the quality of future observations generated by the original and fine-tuned NWM on the Habitat Seen and Unseen splits. In the **Time** mode, each future frame is predicted independently from a fixed history of real observations. In the **Rollout** mode, each predicted frame is fed back as the history for autoregressive prediction of subsequent frames.

As shown in Table 7, NWM fine-tuning consistently improves future-image quality on both splits and under both generation modes, indicating that navigation-specific adaptation improves future observation prediction and generalizes to unseen environments. The gains remain in the autoregressive Rollout setting, although its overall quality is still lower than that of the Time mode conditioned on real observation histories, suggesting that prediction error accumulation remains a challenge.

We further conduct a counterfactual evaluation with the navigation policy and progress scorer frozen to examine whether improved image prediction benefits candidate-action ranking. Under identical conditions, future observations generated by the original NWM, the fine-tuned NWM, and actual execution in Habitat are passed to the same frozen progress scorer. Candidate Spearman measures the rank correlation, within each state, between the four candidate scores and their Habitat-based reference scores. On the complete R2R-CE val-seen split, NWM fine-tuning improves Candidate Spearman from 0.150 to 0.320 and pairwise ranking accuracy from 55.17% to 65.02%. On the current val-unseen subset, the corresponding improvements are from 0.169 to 0.320 and from 56.02% to 64.94%, while top-ranked candidate agreement increases from 47.9% to 54.2%. These results show that NWM fine-tuning improves not only future-image quality, but also the relative candidate-ranking information preserved in generated observations, making them more suitable for candidate-action comparison and imagined-progress reward construction.

Table 7: Future prediction quality before and after NWM fine-tuning. LPIPS, DreamSim, and PSNR are averaged over future frames 1–4. FID@4 is computed over the complete fourth-frame image set for each split and generation mode. Bold indicates the better result within each pair.

Split	Mode	LPIPS ↓		DreamSim ↓		PSNR ↑ (dB)		FID@4 ↓	
		Original	Fine-tuned	Original	Fine-tuned	Original	Fine-tuned	Original	Fine-tuned
Seen	Time	0.4542	**0.2212**	0.3176	**0.1512**	15.69	**21.64**	27.51	**11.57**
	Rollout	0.4522	**0.2357**	0.3158	**0.1630**	15.61	**21.10**	32.13	**12.73**
Unseen	Time	0.4701	**0.2600**	0.3267	**0.1722**	14.90	**19.48**	25.08	**9.64**
	Rollout	0.4674	**0.2728**	0.3265	**0.1832**	14.87	**19.07**	30.36	**10.68**

A.5 Zero-shot Real-World Evaluation

We present qualitative comparisons between Uni-NaVid and IP-GRPO w/ Uni-NaVid on three real-world navigation tasks, with 20 trials per method for each task. Each trial includes a top-down trajectory and chronological egocentric observations from all recorded inference rounds. Circles and crosses indicate trajectory start and end points, respectively. The number of observations varies with the episode length, covering navigation until model termination or manual/system interruption. Trials are interrupted only for safety, such as imminent collisions, and are counted as failures. Success requires the policy to issue STOP and satisfy the task success criteria.

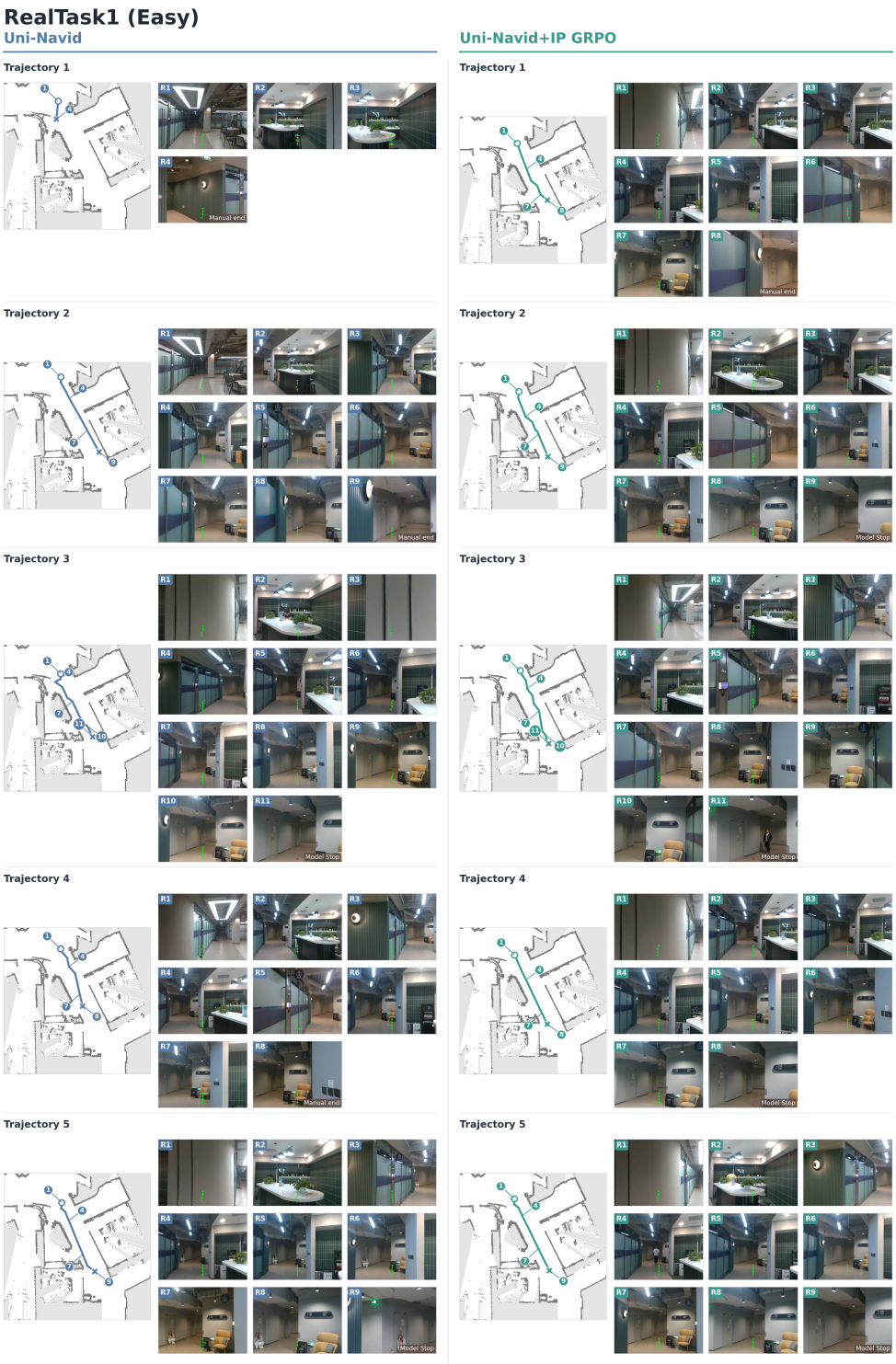

Figure 4: **RealTask1.**

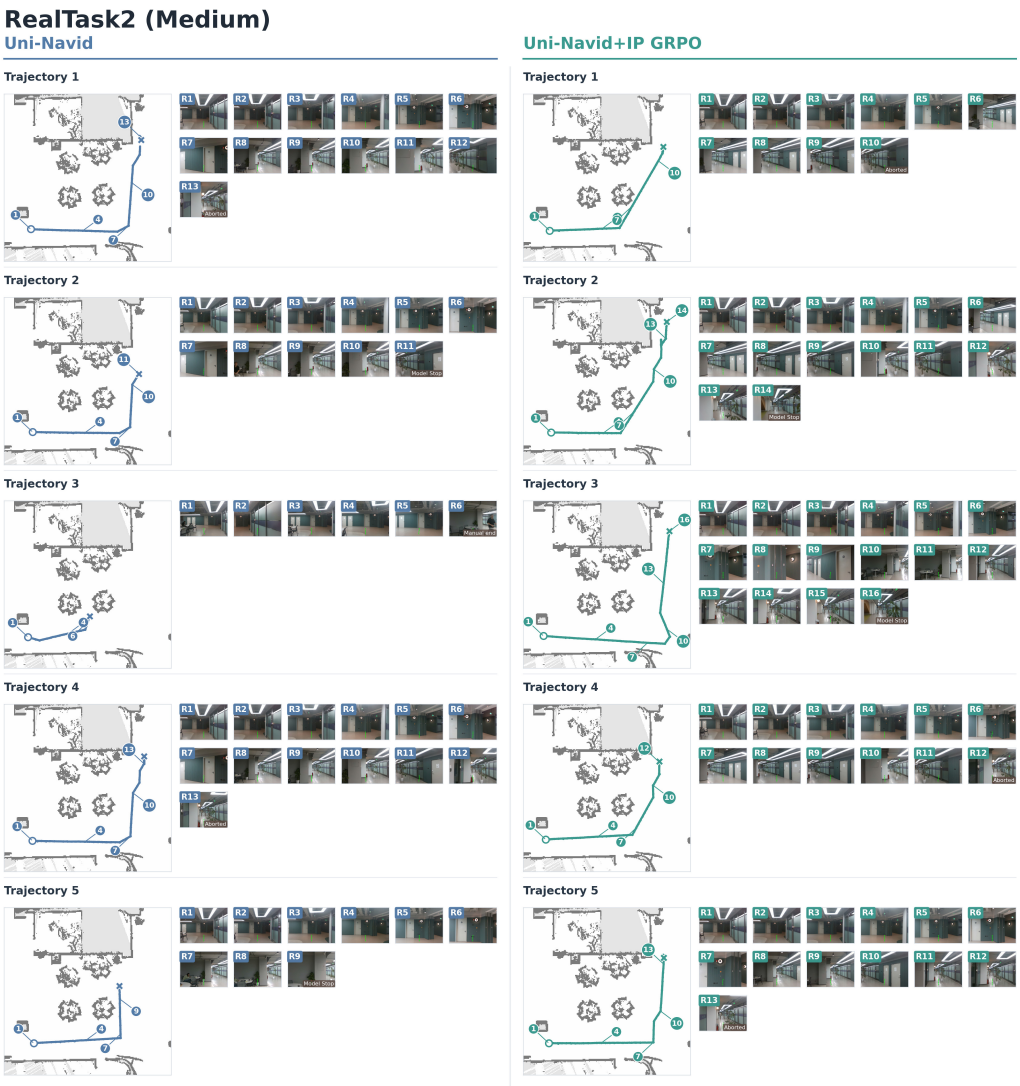

Figure 5: **RealTask2.**



Figure 6: **RealTask3.**