WNM-3D: A WORLD NAVIGATION MODEL WITH 3D SCENE CONDITIONING FOR CLOSED-LOOP VLN

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent vision-language navigation (VLN) systems increasingly adapt pretrained vision-language models (VLMs) into vision-language-action (VLA) policies that map egocentric observations and language instructions directly to navigation actions. Although semantically capable, such action-centric training does not explicitly model how the agent's visual observations evolve under its predicted motion. Generative world-action models (WAMs) jointly predict future observations and actions, yet existing WAMs for continuous VLN do not condition joint future-view and action generation on geometry-aware representations inferred from the observation history. We present **WNM-3D**, a generative **W**orld **N**avigation **M**odel with **3D** scene conditioning for continuous VLN. A frozen feed-forward geometry encoder and trainable **3D Scene-to-Token Adapter** convert monocular RGB history into a fixed-length geometry-aware prefix that persistently conditions joint future-view and action generation. We train WNM-3D with supervised world-action learning, DAgger-style adaptation on policy-visited states, and Counterfactual DanceGRPO refinement. Across 3DGS-based GN-Bench and VLN-CE, WNM-3D outperforms the backbone-matched WNM-2D counterpart and strong VLN baselines on GN-Bench. We further demonstrate closed-loop deployment on a physical robot, while component and stage-wise ablations support the Adapter design and the complementary roles of DAgger-SFT and DanceGRPO.

1 INTRODUCTION

Vision-language navigation (VLN) requires an embodied agent to translate language instructions and streaming egocentric observations into navigation actions under partial observability and closed-loop interaction. Recent VLN methods increasingly build action-generating policies upon pretrained vision-language models (VLMs), including video-based VLM policies and explicitly formulated vision-language-action (VLA) models (Zhang et al., 2024; 2025c; Cheng et al., 2025). These models inherit strong semantic priors for instruction understanding, landmark grounding, and visual reasoning. However, they are optimized primarily through action prediction and therefore do not explicitly model how the agent's observations evolve under its predicted motion. This omission is consequential in continuous navigation, where every action changes the viewpoint and thus the visual evidence available to subsequent decisions. Action supervision alone provides no direct predictive constraint on this closed-loop observation–action evolution.

Generative world models provide a complementary perspective by predicting how observations evolve through interaction. Action-conditioned navigation world models generate future egocentric observations under candidate actions (Bar et al., 2025), whereas generative world-action models (WAMs) jointly model future observations and executable actions within a coupled predictive system (Ye et al., 2026; Zhao et al., 2026). Existing WAMs for continuous VLN, however, do not condition joint future-view and action generation on geometry-aware representations inferred from the observation history. RGB histories or video latents preserve appearance and temporal context, but do not directly consolidate multi-view observations into persistent geometry-aware scene context. As a result, the correspondence between predicted motion and its associated visual transition can remain underconstrained, particularly under large viewpoint changes. A monocular observation history nevertheless contains multiple views of the same environment as the agent moves. This motivates our central question: *how can geometry-aware information recovered from this history be incorporated as shared context for both future-view prediction and navigation-action generation?*

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.

We present **WNM-3D**, a generative **W**orld **N**avigation **M**odel with **3D** scene conditioning for continuous VLN. To the best of our knowledge, WNM-3D is the first generative world-action model for continuous VLN to incorporate history-derived 3D scene information into joint future-view and action generation. A frozen feed-forward geometry encoder extracts cross-view features from monocular egocentric RGB history, and a trainable **3D Scene-to-Token Adapter** converts them into a fixed-length prefix aligned with the world-action Diffusion Transformer (DiT). Through block-causal attention, the scene prefix remains visible to every future video-action block throughout joint denoising, providing persistent geometric context.

Closed-loop navigation exposes the policy to states absent from offline expert demonstrations, motivating a three-stage curriculum: supervised world-action learning on offline expert trajectories, DAgger-SFT on policy-visited states, and Counterfactual DanceGRPO for reward-guided refinement. We evaluate WNM-3D on GN-Bench (Li et al., 2026) and VLN-CE (Krantz et al., 2020), which provide high-fidelity 3DGS and reconstruction-based visual domains, respectively, and deploy it on a physical robot. WNM-3D consistently outperforms the backbone-matched WNM-2D on both benchmarks, outperforms VLN baselines on GN-Bench, and demonstrates closed-loop operation in real indoor environments. Component and stage-wise ablations validate the Adapter design and characterize the roles of DAgger-SFT and DanceGRPO. Our contributions are threefold:

- We formulate a geometry-conditioned WAM for continuous VLN, in which geometry-aware scene tokens derived from the observation history provide shared conditioning for future visual latents and temporally aligned navigation actions.
- We introduce a modular 3D Scene-to-Token Adapter that combines geometry-aware feature fusion, content-initialized target queries, anchored deformable resampling, and factorized spatiotemporal refinement to produce a fixed-length world-action DiT prefix.
- We establish a progressive training curriculum combining supervised world-action learning, DAgger-SFT, and Counterfactual DanceGRPO, with consistent WNM-3D gains on GN-Bench and VLN-CE and closed-loop physical-robot deployment.

2 Related Work

VLM-Based Navigation and Closed-Loop Policy Learning. Recent VLN methods adapt pre-trained vision-language models (VLMs) into action-generating policies, spanning video-based VLMs, vision-language-action (VLA) models, and cross-task navigation foundation models (Zhang et al., 2024; 2025c; Cheng et al., 2025; Zhang et al., 2025b; Wang et al., 2026b). Planning, dialogue, scene graphs, spatial memory, and hierarchical reasoning further improve decision making (Long et al., 2024; Sheng et al., 2026b; Zhou et al., 2026a; Huang et al., 2025; Su et al., 2026; Deng et al., 2026; Wang et al., 2026c; Ma et al., 2026b; He et al., 2025; Wang et al., 2023; 2024; Sheng et al., 2026a), but these approaches primarily map observation–instruction histories to actions rather than jointly modeling action-induced future observations. Closed-loop covariate shift is addressed by DAgger (Ross et al., 2011) and supervised/RL refinement pipelines such as GN0 (Li et al., 2026). Reinforcement learning has also been studied for policy optimization and adaptation (Li et al., 2024; Zhang et al., 2023; 2025a), while DanceGRPO extends group-relative optimization to diffusion and rectified-flow generators (Xue et al., 2025). Device–edge–cloud collaboration has additionally been explored to alleviate multimodal-model deployment bottlenecks (An et al., 2026; Shao & Li, 2025).

Generative and Geometry-Aware World-Action Models. Action-conditioned navigation world models predict future egocentric observations under candidate actions and rely on a separate planner or action selector (Bar et al., 2025). World-action models (WAMs) instead couple future-state prediction and action generation, with DreamZero providing the joint video-action flow formulation adopted by WNM-3D (Ye et al., 2026). In navigation, WAM-Nav and NavWAM jointly model visual foresight and actions (Yang et al., 2026; Azuma et al., 2026); SWAM generates intermediate RGB-D observations and action trajectories from start and goal RGB inputs (Chen et al., 2026). FutureNav jointly optimizes action, dynamics, and future spatial-state objectives (Zhang et al., 2026b), while NavWM uses depth- and semantics-supervised world tokens for foresight-based trajectory selection (Mei et al., 2026). These methods do not aggregate observation history into a persistent geometry-aware condition for joint future-view and action generation. Feed-forward geometry models such as VGGT and VGGT-Ω infer cross-view scene structure (Wang et al., 2025; 2026a), while

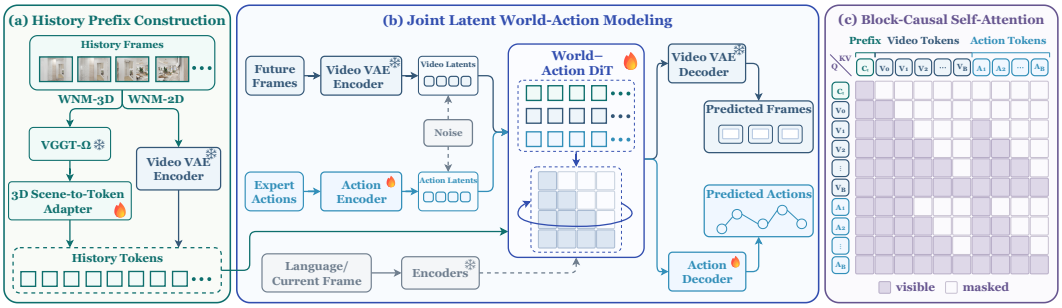


Figure 1: **Overview of WNM-3D.** (a) **History prefix construction.** WNM-3D converts history-derived VGGT-Ω features into a geometry-aware scene prefix, while WNM-2D uses VAE-encoded RGB history. (b) **Joint latent world-action modeling.** The world-action DiT jointly predicts future visual latents and temporally aligned actions conditioned on the history prefix, current frame, and language instruction. (c) **Block-causal self-attention.** Block-causal attention keeps the clean history prefix visible to all future blocks while preserving temporal causality.

geometry-aware WAMs in other embodied domains include DriveDreamer-Policy, GeoSem-WAM, and MECo-WAM (Zhou et al., 2026b; Ma et al., 2026a; Zhang et al., 2026a). These approaches either target non-VLN domains or use geometry mainly through auxiliary objectives or training-time priors. In contrast, WNM-3D converts history-derived geometry features into scene tokens retained as a shared inference-time condition for future-view and language-conditioned action generation.

3 Method

Overview. WNM-3D jointly predicts finite-horizon future views and temporally aligned navigation actions at each replanning step. As shown in Fig. 1, it builds on the joint video-action flow backbone of DreamZero (Ye et al., 2026) and introduces a frozen geometry encoder followed by a trainable 3D Scene-to-Token Adapter to construct a geometry-aware history prefix. **WNM-2D** shares the same world-action backbone, prediction targets, block-causal attention, and three-stage training curriculum, but replaces this prefix with the backbone's native VAE-encoded RGB history.

3.1 Problem Formulation

We consider continuous VLN under partial observability. At replanning step t, the agent receives a language instruction ℓ and maintains a history $\mathcal{H}_t = (I_{\tau_1}, \dots, I_{\tau_K})$ of K monocular egocentric RGB observations, where $\tau_1 < \cdots < \tau_K = t$. The model predicts B future visual blocks $\mathcal{Y}_t = (Y_{t,1}, \dots, Y_{t,B})$, each containing N_v future RGB frames, together with temporally aligned action blocks $\mathcal{A}_t = (\mathcal{A}_{t,1}, \dots, \mathcal{A}_{t,B})$, each containing N_a navigation actions. The current observation I_t immediately precedes the first predicted visual block.

For training-target construction, the expert trajectory is smoothed and expressed in a fixed local navigation frame anchored at the camera pose of I_t. Let $\mathbf{p}_j^{(t)} = (p_{j,x}^{(t)}, p_{j,y}^{(t)})$ denote the j-th sampled planar position, with $\mathbf{p}_0^{(t)} = \mathbf{0}$, and $\psi_j^{(t)}$ its tangent heading. These quantities only define action targets and are unavailable to the policy. Navigation actions are consecutive-sample increments:

$$a_j^{(t)} = \left(p_{j+1,x}^{(t)} - p_{j,x}^{(t)}, p_{j+1,y}^{(t)} - p_{j,y}^{(t)}, \mathrm{wrap}_{[-\pi,\pi)}\left(\psi_{j+1}^{(t)} - \psi_j^{(t)}\right)\right), \qquad j = 0, \dots, BN_a - 1. \quad (1)$$

Thus, $a_j^{(t)} = (\Delta x_j^{(t)}, \Delta y_j^{(t)}, \Delta\psi_j^{(t)})$ represents the incremental translation and heading change between consecutive samples in the local frame anchored at I_t. The actions are grouped as

$$\mathcal{A}_{t,b} = \left(a_{(b-1)N_a}^{(t)}, \dots, a_{bN_a-1}^{(t)}\right) \in \mathbb{R}^{N_a \times 3}, \qquad b \in \{1, \dots, B\}. \quad (2)$$

Each action block spans the temporal interval of its paired visual block. For WNM-3D, we model the joint distribution $p_{\theta,\phi}(\mathcal{Y}_t, \mathcal{A}_t \mid \mathcal{H}_t, \ell)$, where θ parameterizes the world-action backbone and its modality-specific layers, and ϕ the 3D Scene-to-Token Adapter. WNM-2D follows the same predictive formulation with the native RGB-history prefix and no adapter parameters ϕ.

3.2 History Conditioning and Block-Causal Attention

Let $\mathbf{C}_t \in \mathbb{R}^{N_c \times d}$ be the clean history-conditioning prefix prepended to the DiT self-attention stream, with fixed length N_c and hidden width d. As shown in Fig. 1(a), WNM-2D forms $\mathbf{C}_t^{\mathrm{2D}}$ from native VAE-encoded RGB history, whereas WNM-3D applies frozen E_{geo} followed by trainable T_ϕ:

$$\mathbf{C}_t^{\mathrm{3D}} = T_\phi(E_{\mathrm{geo}}(\mathcal{H}_t)) \in \mathbb{R}^{N_c \times d}. \tag{3}$$

In parallel, the current frame and language instruction are encoded through the backbone's native cross-attention pathway as $\mathbf{G}_t = [E_{\mathrm{img}}(I_t); E_\ell(\ell)]$. Following DreamZero, the video VAE latent of I_t defines the block-zero visual flow variable $\mathbf{V}_{\sigma_0,0}$.

During training, the current-frame visual group $\mathbf{V}_{\sigma_0,0}$, the future visual groups $\{\mathbf{V}_{\sigma_b,b}\}_{b=1}^{B}$, and the aligned action groups $\{\mathbf{A}_{\sigma_b,b}\}_{b=1}^{B}$ are serialized with the clean prefix as

$$\mathcal{S}_{\boldsymbol{\sigma}} = [\mathbf{C}_t, \mathbf{V}_{\sigma_0,0}, \mathbf{V}_{\sigma_1,1}, \ldots, \mathbf{V}_{\sigma_B,B}, \mathbf{A}_{\sigma_1,1}, \ldots, \mathbf{A}_{\sigma_B,B}]\,. \tag{4}$$

Each future visual-action pair shares the same flow timestep σ_b but uses independently sampled noise, while $\mathbf{C}_t$ and $\mathbf{G}_t$ remain clean. The two variants otherwise share the predicted variables, attention layout, backbone, and objectives.

Fig. 1(c) shows a block-causal attention mask indexed by temporal block rather than serialized token position: the prefix attends only to itself; block zero to the prefix and itself; and each future visual-action block to the prefix, block zero, and all visual-action blocks up to itself. Attention is bidirectional within blocks and causal across blocks.

3.3 Joint World-Action Flow Matching

As illustrated in Fig. 1(b), we adopt DreamZero's joint flow-matching formulation (Ye et al., 2026) with a shared world-action DiT. Let $\widetilde{\mathcal{Y}}_t = (I_t, Y_{t,1}, \ldots, Y_{t,B})$ denote the complete visual sequence, which the frozen video VAE encodes into blockwise latents $\{\mathbf{x}_{\mathrm{data}}^{v,b}\}_{b=0}^{B}$. Action rollouts are scaled, normalized using training-set statistics, and zero-padded to the backbone's fixed action width, yielding $\{\mathbf{x}_{\mathrm{data}}^{a,b}\}_{b=1}^{B}$.

For modality $m \in \{v, a\}$ and temporal block b, we sample $\boldsymbol{\epsilon}^{m,b} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and construct the linear conditional-flow path

$$\mathbf{x}_{\sigma_b}^{m,b} = (1-\sigma_b)\mathbf{x}_{\mathrm{data}}^{m,b} + \sigma_b \boldsymbol{\epsilon}^{m,b}, \qquad \mathbf{u}_{\sigma_b}^{m,b} = \boldsymbol{\epsilon}^{m,b} - \mathbf{x}_{\mathrm{data}}^{m,b}. \tag{5}$$

Visual blocks use $b \in \{0, \ldots, B\}$, action blocks $b \in \{1, \ldots, B\}$; conditioned on $\mathcal{S}_{\boldsymbol{\sigma}}$, flow timesteps, and $\mathbf{G}_t$, the DiT predicts modality-specific velocity fields. We optimize the world-action objective

$$\mathcal{L}_{\mathrm{WA}} = \mathbb{E}\left[\mathrm{MSE}_w\left(\widehat{\mathbf{u}}_{\theta,\phi}^{v}, \mathbf{u}^{v}\right) + \lambda_a\, \mathrm{MSE}_{w,\mathbf{M}_a}\left(\widehat{\mathbf{u}}_{\theta,\phi}^{a}, \mathbf{u}^{a}\right)\right], \tag{6}$$

where $\widehat{\mathbf{u}}^m$ and $\mathbf{u}^m$ collect blockwise velocity predictions and targets, and λ_a weights the action loss. MSE_w applies blockwise flow-timestep weighting, while $\mathrm{MSE}_{w,\mathbf{M}_a}$ additionally masks padded action dimensions with $\mathbf{M}_a$. The two modalities retain separate input/output transformations and velocity targets while interacting through the shared DiT.

3.4 3D Scene-to-Token Adapter

The frozen VGGT-Ω (Wang et al., 2026a) encoder produces L patch-feature grids $\{\mathbf{F}_t^{(l)}\}_{l=1}^{L}$, each of shape $K \times H_s \times W_s \times d_l$. As shown in Fig. 2, the adapter maps them to a fixed-length DiT prefix through multi-level fusion, structured query formation, anchored deformable resampling, and factorized spatiotemporal refinement.

Geometry Encoding. Each selected encoder level is normalized and projected to a common width d_s. Because the feature grids share the same history–height–width lattice, aligned locations can be fused directly without resampling. A lightweight gating network assigns location-adaptive weights across levels, producing the geometry-aware source memory $\bar{\mathbf{F}}_t$:

$$\mathbf{P}_t^{(l)} = P_l\left(\mathbf{F}_t^{(l)}\right), \qquad \alpha_{t,i}^{(l)} = \frac{\exp\left(g(\mathbf{P}_{t,i}^{(l)})\right)}{\sum_{r=1}^{L}\exp\left(g(\mathbf{P}_{t,i}^{(r)})\right)}, \qquad \overline{\mathbf{F}}_{t,i} = \sum_{l=1}^{L}\alpha_{t,i}^{(l)}\mathbf{P}_{t,i}^{(l)}. \tag{7}$$

Here, i indexes the shared history–height–width lattice, and $\overline{\mathbf{F}}_t \in \mathbb{R}^{K\times H_s\times W_s\times d_s}$ denotes the fused geometry-aware source memory.

3D Query Resampling. We adaptively pool the source memory onto a target lattice and initialize each target query with pooled scene content, a learned slot, and structured embeddings:

$$\mathbf{B}_t = \mathrm{Pool}_{T_c,H_c,W_c}\left(\overline{\mathbf{F}}_t\right), \qquad \mathbf{Q}_t^0 = \mathbf{Q}_{\mathrm{slot}} + P_B(\mathbf{B}_t) + \mathbf{E}_{\mathrm{struct}}. \tag{8}$$

The target lattice contains $N_c = T_cH_cW_c$ queries. $\mathbf{Q}_{\mathrm{slot}}$ contains learned target-grid slots, while $\mathbf{E}_{\mathrm{struct}}$ aggregates global, temporal, row, column, target-coordinate, and query-type embeddings.

The pooled base preserves coarse context but may lose fine-grained evidence. We therefore apply R anchored deformable-resampling layers, where each query predicts offsets and aggregation weights around a regular source-lattice anchor and retrieves a coordinate-augmented local neighborhood. This yields refined queries $\mathbf{Q}_t^R$ without global cross-attention over the full source memory.

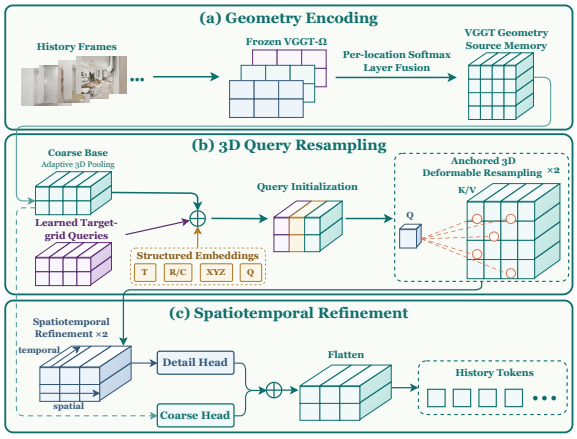


Figure 2: **3D Scene-to-Token Adapter.** (a) Multi-level geometry encoding. (b) Anchored 3D query resampling. (c) Factorized spatiotemporal refinement into fixed-length scene tokens.

Spatiotemporal Refinement. The resampled queries are reshaped to the target lattice and processed by factorized blocks comprising spatial self-attention within each temporal slice, temporal self-attention at each spatial location, and an MLP residual. A detail head projects the refined queries, while a coarse head projects the pooled scene base as a residual:

$$\mathbf{C}_t^{\mathrm{3D}} = \mathrm{Flatten}\left[H_{\mathrm{detail}}\left(\mathrm{STRefine}_\phi(\mathbf{Q}_t^R)\right) + H_{\mathrm{coarse}}(\mathbf{B}_t)\right] \in \mathbb{R}^{N_c\times d}. \tag{9}$$

The resulting tokens form the clean prefix in Eq. (3) and provide shared scene context for future-view and action generation. The adapter is trained jointly with the world-action components, while the geometry encoder remains frozen.

3.5 Progressive Training and Receding-Horizon Inference

We optimize WNM-2D and WNM-3D in three stages, progressively shifting from offline expert supervision to policy-induced state correction and reward-guided refinement, as illustrated in Fig. 3.

Stage-I: Offline Expert SFT. We optimize $\mathcal{L}_{\mathrm{WA}}$ on offline expert trajectories: A*-generated paths on GN-Bench (Hart et al., 1968) and dataset-provided reference paths on VLN-CE. Each expert action rollout is paired with the aligned future visual continuation. For WNM-3D, the adapter and world-action backbone are trained while VGGT-Ω remains frozen.

Stage-II: Closed-Loop DAgger-SFT. To mitigate closed-loop covariate shift, we roll out the Stage-I policy and query benchmark-specific experts at policy-visited states (Ross et al., 2011): A* on GN-Bench and the reference-aware navmesh oracle on VLN-CE (Appendix B.2). The resulting corrective trajectories and aligned future observations form the Stage-II DAgger dataset for further optimization with $\mathcal{L}_{\mathrm{WA}}$.

Stage-III: Counterfactual DanceGRPO. Starting from Stage-II, we adapt DanceGRPO (Xue et al., 2025) to the joint visual-action flow policy. We optimize the policy using current-policy world-action samples under a fixed set of navigation contexts without stepping the interactive simulator; generated views and actions are scored using reference observations, expert trajectories, occupancy maps, and navigation metadata. No auxiliary supervised $\mathcal{L}_{\mathrm{WA}}$ term is used.

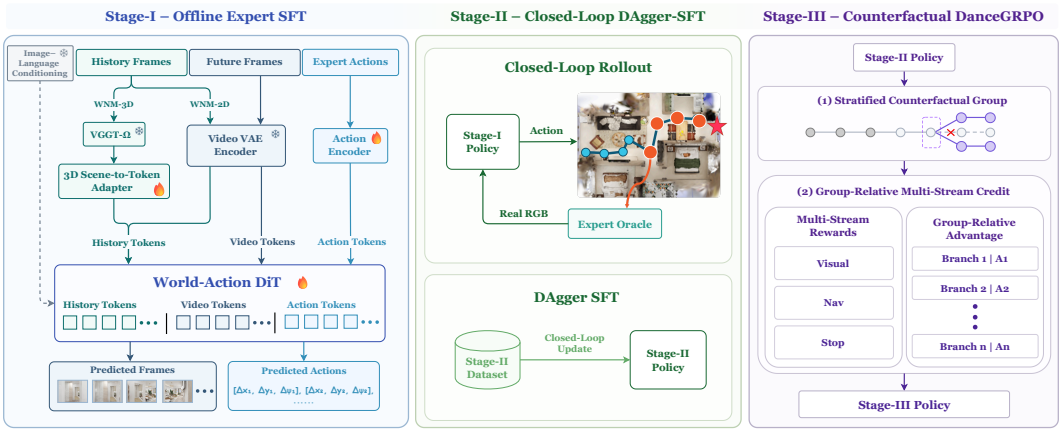


Figure 3: **Overview of the three-stage training curriculum.** (a) Stage-I: Offline Expert SFT with geometry-aware or RGB-history conditioning. (b) Stage-II: Closed-loop DAgger-SFT on expert corrections from policy-visited states. (c) Stage-III: Counterfactual DanceGRPO with stratified world-action groups and group-relative visual, navigation, and stopping rewards.

To localize outcome credit along the diffusion trajectory, we partition eligible DiT source transitions into noise strata and sample one intervention per stratum. For each conditioning instance and transition, counterfactual branches share their initial visual and action latents and all exogenous SDE noise except branch-specific noise at the intervention; each branch completes the reverse-SDE trajectory for reward evaluation, while gradients are replayed only through the intervened transition.

We maintain separate reward streams for future-view quality, navigation, and stopping. For each stream q, the branch advantage is standardized within its conditioning-instance–stratum group:

$$A^q_{i,k,h} = \frac{R^q_{i,k,h} - \mu^q_{i,k}}{\sigma^q_{i,k} + \epsilon}, \qquad \mu^q_{i,k} = \frac{1}{H}\sum_{h'=1}^{H} R^q_{i,k,h'}. \tag{10}$$

Here, H is the group size, $\sigma^q_{i,k}$ the within-group reward standard deviation, and ϵ a numerical stabilizer. Visual rewards score future-view quality and visual-action consistency, while navigation and stopping rewards evaluate action chunks against trajectory and occupancy information.

Let r^v denote the current-to-old transition-density ratio for visual variables at the intervened transition and r^a_c the corresponding ratio for action chunk c. Using the clipped DanceGRPO surrogate ℓ_{clip}, we route visual credit through r^v and navigation and stopping credit through r^a_c:

$$\begin{aligned} \mathcal{L}_v &= \mathbb{E}\left[\ell_{\text{clip}}\left(r^v, A^v\right)\right], \\ \mathcal{L}_{s,c} &= \mathbb{E}\left[\ell_{\text{clip}}\left(r^a_c, A^s_c\right)\right], \qquad s \in \{\text{nav}, \text{stop}\}. \end{aligned} \tag{11}$$

Although both modalities update the shared world-action DiT, this modality-routed surrogate prevents reward streams from being indiscriminately assigned across modalities.

The complete Stage-III objective combines the visual, navigation, and stopping losses:

$$\mathcal{L}_{\text{III}} = \mathcal{L}_v + \lambda_{\text{act}}\left(\mathcal{L}_{\text{nav}} + \lambda_{\text{stop}}\mathcal{L}_{\text{stop}}\right). \tag{12}$$

Here, $\mathcal{L}_{\text{nav}}$ and $\mathcal{L}_{\text{stop}}$ are chunk-aggregated losses and λ_{act}, λ_{stop} weight the action-side and stopping objectives; Appendix A.4 gives reward and optimization details.

Receding-Horizon Inference. At replanning step t, both variants sample K observations up to I_t, recompute the conditioning prefix, and jointly generate finite-horizon visual-action rollouts conditioned on $\mathbf{G}_t$. Only controls derived from the first action block $\mathcal{A}_{t,1}$ are executed before replanning; predicted visual latents are neither decoded into RGB nor fed back into $\mathcal{H}_t$.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks and metrics. We evaluate WNM-2D and WNM-3D on GN-Bench Seen/Unseen and VLN-CE R2R Val-Unseen using only language and monocular egocentric RGB at inference. We report Navigation Error (NE), Oracle Success (OS), Success Rate (SR), Trajectory Length (TL), and Success weighted by Path Length (SPL) on GN-Bench and NE, OS, SR, and SPL on VLN-CE. Lower NE and higher OS, SR, and SPL are better. Detailed protocols are provided in Appendix B.

Baselines. On GN-Bench, we compare with CMA (Krantz et al., 2020), NaVid (Zhang et al., 2024), UniNaVid (Zhang et al., 2025c), InternNav(S2) (InternNav Team, 2025), and GN-BAE (Li et al., 2026), including GN-Matrix-supervised variants reported by GN0 where available. On VLN-CE, we compare with NaVid, NaVILA (Cheng et al., 2025), UniNaVid, StreamVLN (Wei et al., 2025), DualVLN (Wei et al., 2026), CorrectNav (Yu et al., 2026), and GN-BAE. External baselines follow their training settings and are not data-matched. WNM-2D is our backbone-matched counterpart, replacing the geometry-aware prefix with VAE-encoded RGB history while retaining the same backbone, targets, attention layout, objective, three-stage training protocol, and inference scheme.

Implementation details. Both variants initialize from Wan2.2-TI2V-5B, use $K = 33$ history frames, and predict $B = 4$ blocks with $N_a = 8$ actions each. WNM-3D keeps VGGT-Ω frozen while optimizing the Adapter and world-action backbone. Training and benchmark-specific details are in Appendices A and B.2.

4.2 CLOSED-LOOP NAVIGATION

4.2.1 GN-BENCH

Table 1 reports GN-Bench results. Both WNM variants use only monocular RGB at inference; WNM-3D internally derives geometry-aware scene tokens without depth, BEV observations, or an explicit metric map. Additional rollouts are in Appendix B.1.

Table 1: **Evaluation Metrics on GN-Bench.** Depth/BEV/FPV denote depth, bird's-eye-view, and first-person RGB inputs. Bold/underline mark best/second-best results; † denotes GN-Matrix SFT.

Method	Observation			Metrics on Seen					Metrics on Unseen				
	Depth	BEV	FPV	TL	NE ↓	OS ↑	SR ↑	SPL ↑	TL	NE ↓	OS ↑	SR ↑	SPL ↑
CMA	✓		✓	2.9	8.3	15.7	12.5	11.9	3.0	8.1	19.6	15.5	14.9
NaVid			✓	3.4	7.9	20.1	14.6	12.8	3.5	7.7	20.3	14.5	12.8
UniNaVid			✓	4.6	7.9	22.2	15.0	12.5	5.2	7.8	20.7	12.8	10.3
InternNav(S2)			✓	3.6	7.4	23.1	18.8	17.5	3.7	7.2	26.7	22.1	20.3
NaVid†			✓	2.7	7.4	19.4	18.8	18.8	2.7	7.1	23.8	23.1	23.0
UniNaVid†			✓	3.7	7.2	24.1	22.5	21.9	5.8	7.5	23.1	20.8	20.2
InternNav(S2)†			✓	2.9	7.1	22.5	22.4	22.4	2.9	6.9	24.9	24.0	23.7
GN-BAE			✓	5.2	4.9	48.9	46.4	44.7	5.0	5.6	43.6	38.9	37.3
GN-BAE		✓	✓	5.2	4.3	59.3	58.6	58.6	4.0	5.8	40.2	38.5	38.2
WNM-2D			✓	7.9	<u>2.8</u>	<u>80.7</u>	<u>75.6</u>	<u>72.9</u>	7.9	<u>4.8</u>	<u>52.6</u>	<u>45.9</u>	<u>42.8</u>
WNM-3D			✓	10.6	**1.9**	**94.2**	**88.7**	**80.9**	10.2	**4.1**	**62.2**	**53.5**	**46.6**

Comparison with prior methods. On Seen environments, WNM-3D improves SR and SPL over the strongest baseline in Table 1, GN-BAE with BEV observations, by 30.1 and 22.3 points. On Unseen environments, it improves over the strongest FPV-only by 14.6 SR and 9.3 SPL points.

Effect of 3D scene conditioning. WNM-3D improves over WNM-2D by 13.1 SR and 8.0 SPL points on Seen environments and by 7.6 SR and 3.8 SPL points on Unseen environments. This backbone-matched comparison supports the benefit of geometry-aware history conditioning. The gains are larger on Seen, however, and do not indicate a reduced Seen-to-Unseen performance gap.

4.2.2 VLN-CE

Table 2 compares monocular-RGB continuous VLN configurations on R2R Val-Unseen. WNM-3D achieves 4.35 NE, 66.7% OS, 63.1% SR, and 57.9% SPL, yielding competitive performance among methods. Unlike the high-fidelity 3DGS observations in GN-Bench, VLN-CE relies on reconstruction-based observations with artifacts and variations in lighting and scene details (Krantz & Lee, 2022), providing a complementary evaluation under a substantially different visual domain.

Effect of 3D scene conditioning. Relative to WNM-2D, WNM-3D improves OS, SR, and SPL by 11.5, 13.0, and 12.6 points, respectively, while reducing NE by 0.99 m. Together with GN-Bench, these gains show that the benefit of geometry-aware history conditioning persists across the two visual domains.

Table 2: **Evaluation on the VLN-CE R2R Val-Unseen split.** WNM-2D and WNM-3D follow the same three-stage curriculum and evaluation protocol.

Method	R2R Val-Unseen			
	NE ↓	OS ↑	SR ↑	SPL ↑
NaVid	5.47	49.1	37.4	35.9
NaVILA	5.22	62.5	54.0	49.0
UniNaVid	5.58	53.3	47.0	42.7
StreamVLN	4.98	64.2	56.9	51.9
DualVLN	4.05	70.7	64.3	58.5
CorrectNav	4.24	67.5	65.1	62.3
GN-BAE	3.50	75.6	67.7	63.4
WNM-2D	5.34	55.2	50.1	45.3
WNM-3D	4.35	66.7	63.1	57.9

4.3 Real-World Deployment

We further deploy WNM-3D on a physical robot to qualitatively assess its closed-loop operation beyond simulation. The robot follows language instructions from live monocular RGB observations across multiple indoor environments using a real-world-adapted checkpoint and the same high-level receding-horizon action-generation and policy-stopping mechanism used in simulation. Figure 4 shows representative rollouts involving multi-step navigation, turning, landmark approach, and stopping. Platform and adaptation details are provided in Appendix C.

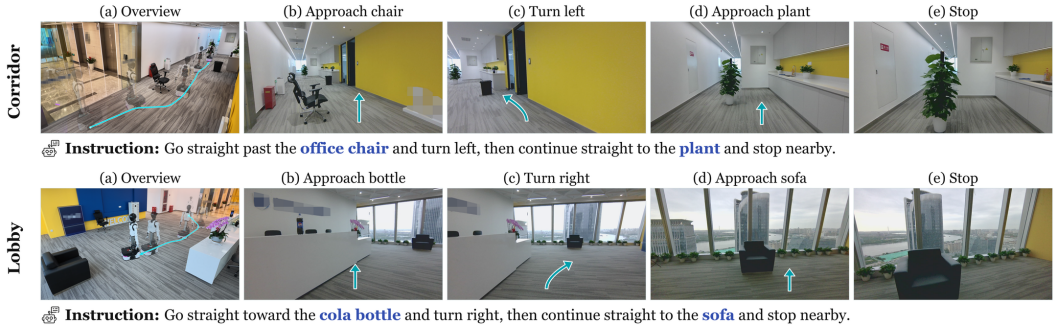


Figure 4: **Qualitative real-world deployment of WNM-3D.** Each row shows a language instruction and selected egocentric observations from a representative closed-loop physical-robot rollout.

4.4 Adapter Component Ablation

To isolate Adapter architecture from subsequent policy adaptation and reward-guided refinement, all variants use only Stage-I Offline Expert SFT with the same frozen VGGT-Ω encoder, world-action backbone, data, optimization settings, and GN-Bench Unseen evaluation. *Pool + Linear* retains only deepest-level pooling and projection, while the four single-component variants remove multi-level fusion, content initialization, deformable resampling, or spatiotemporal refinement, respectively.

Table 3: **Ablation of the 3D Scene-to-Token Adapter on GN-Bench Unseen.**

Variant	GN-Bench Unseen			
	NE ↓	OS ↑	SR ↑	SPL ↑
Pool + Linear	7.1	28.0	24.5	22.3
w/o Multi-level Fusion	5.7	41.5	38.2	35.1
w/o Content Initialization	6.4	37.2	35.1	32.8
w/o Deformable Resampling	6.9	34.9	32.0	30.1
w/o Spatiotemporal Refinement	5.6	43.4	39.8	37.5
Full Adapter	**5.0**	**47.8**	**43.1**	**41.6**

Results. Removing any Adapter component degrades Stage-I Unseen performance. *Pool + Linear* performs worst, showing that direct pooling and projection underuse VGGT-Ω features. Each single-component ablation reduces SR and SPL, supporting complementary roles for cross-level fusion, scene-conditioned query initialization, fine-grained resampling, and spatiotemporal refinement.

4.5 Training-Stage Ablation

Table 4: **Training-stage ablation on GN-Bench.** Checkmarks ✓ denote included stages; WNM-2D and WNM-3D share the downstream backbone and evaluation protocol.

Model	Training Stage			Metrics on Seen					Metrics on Unseen				
	A* SFT	DAgger-SFT	DanceGRPO	TL	NE ↓	OS ↑	SR ↑	SPL ↑	TL	NE ↓	OS ↑	SR ↑	SPL ↑
WNM-3D	✓			7.0	3.8	61.2	56.7	54.4	5.9	5.0	47.8	43.1	41.6
	✓	✓		13.9	2.0	92.9	80.7	66.6	13.2	4.3	62.8	49.7	38.2
	✓		✓	6.9	4.6	51.0	47.1	42.6	6.1	5.4	44.2	39.2	36.0
	✓	✓	✓	10.6	1.9	94.2	88.7	80.9	10.2	4.1	62.2	53.5	46.6
WNM-2D	✓			3.9	5.2	39.9	38.2	38.0	3.7	5.6	37.2	35.6	35.3
	✓	✓		9.0	3.0	79.4	70.9	66.2	8.9	5.0	51.9	43.7	39.1
	✓		✓	3.6	5.6	35.7	34.1	33.8	3.4	6.0	33.7	31.8	31.4
	✓	✓	✓	7.9	2.8	80.7	75.6	72.9	7.9	4.8	52.6	45.9	42.8

We compare Stage-I, Stage-II, the full curriculum, and direct Stage-I→DanceGRPO, which skips DAgger-SFT but uses records from the same Stage-I-policy collection. All configurations share the same inference procedure and GN-Bench evaluation episodes.

Effect of DAgger-SFT. For WNM-3D, Stage-II improves Seen SR/SPL by 24.0/12.2 points and Unseen SR by 6.6 points, but Unseen SPL drops 3.4 points as TL increases from 5.9 to 13.2. Thus, supervised correction at policy-induced states substantially improves success but does not alone ensure path efficiency under scene-level distribution shift.

Effect of DanceGRPO. From Stage-II, DanceGRPO improves WNM-3D by 8.0 SR / 14.3 SPL points on Seen and 3.8 / 8.4 points on Unseen, while reducing TL from 13.9 to 10.6 and 13.2 to 10.2, respectively. WNM-2D shows the same trend (+4.7/+6.7 Seen and +2.2/+3.7 Unseen), indicating improved success and path efficiency after DAgger adaptation.

Role of DAgger-SFT before DanceGRPO. Direct Stage-I→DanceGRPO degrades SR for both variants (WNM-3D: 56.7→47.1 Seen, 43.1→39.2 Unseen; WNM-2D: 38.2→34.1, 35.6→31.8). Starting from the same Stage-I checkpoint and policy-state collection, this comparison contrasts reward-guided refinement with supervised correction, although objectives and sample budgets differ. The full pipeline then recollects Stage-III records from the Stage-II policy. We hypothesize that supervised correction broadens feasible policy behavior before group-relative optimization; verifying this mechanism requires direct measurements of within-group action diversity and reward spread.

5 Conclusion

We presented WNM-3D, a geometry-conditioned generative world-action model for continuous VLN that converts history-derived geometry into a shared scene-token prefix for joint future-view and action generation. WNM-3D consistently outperforms the backbone-matched WNM-2D on GN-Bench and VLN-CE, outperforms strong prior policies on GN-Bench, and demonstrates qualitative closed-loop robot deployment. Component ablations support the 3D Scene-to-Token Adapter design. Training-stage ablations show that DAgger-SFT improves policy adaptation, after which DanceGRPO further improves navigation success and path efficiency; applying DanceGRPO directly after Stage-I instead degrades performance.

AI USE STATEMENT

We used generative AI tools to assist with drafting and polishing the manuscript, including language refinement, presentation, and LaTeX formatting, and for literature retrieval and related-work discovery. All AI-assisted content and retrieved references were independently reviewed and verified by the authors, who take full responsibility for the manuscript.

ETHICS STATEMENT

This work studies vision-language navigation through simulated benchmarks and qualitative real-world robot demonstrations. Navigation failures may cause collisions or unsafe behavior, and the reported demonstrations do not establish safety for unrestricted deployment. Further deployment should include independent safety controls and human oversight. RGB observations of indoor environments can also contain private information; their collection, storage, and sharing require appropriate privacy protections. Use of the underlying datasets and pretrained models should respect their licenses and intended uses.

REPRODUCIBILITY STATEMENT

Section 3 describes the model and three-stage training curriculum. Appendix A documents the architecture, action encoding and execution, training-data construction, optimization settings, and reward definitions. Appendix B specifies the benchmark evaluation protocols, including data splits, inference and stopping procedures, and evaluation metrics. Appendix C describes the robot platform and real-world adaptation procedure. These details support reproduction of the experimental setup and clarify the scope of the reported results.

REFERENCES

Hongjun An, Wenhan Hu, Sida Huang, Siqi Huang, Ruanjun Li, Yuanzhi Liang, Jiawei Shao, Yiliang Song, Zihan Wang, Cheng Yuan, et al. Ai flow: Perspectives, scenarios, and approaches. *Vicinagearth*, 3(1):1, 2026.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683, 2018.

Daichi Azuma, Taiki Miyanishi, Koya Sakamoto, Shuhei Kurita, Yaonan Zhu, Petr Khrapchenkov, Motoaki Kawanabe, Yusuke Iwasawa, and Yutaka Matsuo. Navwam: A navigation world action model for goal-conditioned visual navigation. *arXiv preprint arXiv:2606.13494*, 2026.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15791–15801, 2025.

Hong Chen, Daqi Liu, Zehan Zhang, Haiguang Wang, Tianhao Lu, Longfei Yan, Haiyang Sun, Fangzhen Li, Hongwei Xie, Bing Wang, et al. Pondering the way: Spatial-perceiving world action model for embodied navigation. *arXiv preprint arXiv:2606.29908*, 2026.

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 *RSS*, 2025.

Yucheng Deng, Pingrui Lai, Xinhai Li, Chenjia Bai, Xiaoheng Deng, Chengnuo Sun, Xuelong Li, and Hua Yang. Spacevln: A zero-shot vision-and-language navigation agent with online spatial cognitive memory and reasoning. *arXiv preprint arXiv:2606.08992*, 2026.

Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

Honglin He, Yukai Ma, Brad Squicciarini, Wayne Wu, and Bolei Zhou. From seeing to experiencing: Scaling navigation foundation models with reinforcement learning. *arXiv preprint arXiv:2507.22028*, 2025.

Yuehao Huang, Liang Liu, Shuangming Lei, Yukai Ma, Hao Su, Jianbiao Mei, Pengxiang Zhao, Yaqing Gu, Yong Liu, and Jiajun Lv. Cogddn: A cognitive demand-driven navigation with decision optimization and dual-process thinking. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 5237–5246, 2025.

InternNav Team. InternVLA-N1: An open dual-system navigation foundation model with learned latent plans, 2025.

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

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the navgraph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pp. 104–120. Springer, 2020.

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.

Lanqing Li, Hai Zhang, Xinyu Zhang, Shatong Zhu, Yang Yu, Junqiao Zhao, and Pheng-Ann Heng. Towards an information theoretic framework of context-based offline meta-reinforcement learning. *Advances in Neural Information Processing Systems*, 37:75642–75667, 2024.

Xinhai Li, Xiaotao Zhang, Yuehao Huang, Jiankun Dong, Tianhang Wang, Sunyao Zhou, Yunzi Wu, Chengnuo Sun, Yunfei Ge, Qizhen Weng, et al. Gn0: Toward a unified paradigm for generation, evaluation, and policy learning in visual-language navigation. *arXiv preprint arXiv:2606.03682*, 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.

Fulong Ma, Daojie Peng, Wenjun Yue, Jiahang Cao, Bintao Wang, Qiang Zhang, and Jun Ma. Geosem-wam: Geometry-and semantic-aware world action models. *arXiv preprint arXiv:2606.03188*, 2026a.

Yukai Ma, Honglin He, Selina Song, Wayne Wu, and Bolei Zhou. Aura: Multi-modal shared autonomy for urban navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18171–18181, 2026b.

Yanghong Mei, Longteng Guo, Ming-Ming Yu, Guiyu Zhao, Xingjian He, and Jing Liu. Navwm: A unified navigation world model for foresight-driven planning. *arXiv preprint arXiv:2606.24101*, 2026.

Stéphane 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*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Jiawei Shao and Xuelong Li. Ai flow at the network edge. *IEEE Network*, 40(1):330–336, 2025.

Kai Sheng, Liuyi Wang, Haojie Dai, Jinlong Li, Yongrui Qin, Zongtao He, Chengju Liu, and Qijun Chen. P2dnav: Panorama-to-downview reasoning for zero-shot vision-and-language navigation, 2026a.

Kai Sheng, Liuyi Wang, Zongtao He, Xiao Lin, Chengju Liu, and Qijun Chen. Dream: Dynamic routing of experts via attention-based mixture for vision-language-action modeling. *Knowledge-Based Systems*, 340:115585, 2026b. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2026.115585.

Hao Su, Yuehao Huang, Yukai Ma, Yong Liu, and Jiajun Lv. Sage-nav: Leveraging llm planning and alignment fusion for hierarchical scene graph-guided navigation. *arXiv preprint arXiv:2606.25497*, 2026.

Hongcheng Wang, Andy Guan Hong Chen, Xiaoqi Li, Mingdong Wu, and Hao Dong. Find what you want: Learning demand-conditioned object attribute space for demand-driven navigation. *Advances in Neural Information Processing Systems*, 36:16353–16366, 2023.

Hongcheng Wang, Peiqi Liu, Wenzhe Cai, Mingdong Wu, Zhengyu Qian, and Hao Dong. Moddn: a coarse-to-fine attribute-based exploration agent for multi-object demand-driven navigation. *Advances in Neural Information Processing Systems*, 37:64176–64214, 2024.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-ω. *arXiv preprint arXiv:2605.15195*, 2026a.

Liuyi Wang, Kai Sheng, Zongtao He, Jinlong Li, Yongrui Qin, Haojie Dai, Xiangyi Wang, Jingwei Yang, Qingqing Yan, Chengju Liu, et al. A comprehensive survey and systematic real-world evaluation of embodied vision-and-language navigation. *IEEE Transactions on Automation Science and Engineering*, 2026b.

Tianhang Wang, Xinhai Li, Fan Lu, Tianshi Gong, Jiankun Dong, Weiyi Xue, Sanqing Qu, Chenjia Bai, and Guang Chen. Conavbench: Collaborative long-horizon vision-language navigation benchmark. In *The Fourteenth International Conference on Learning Representations*, 2026c.

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*, 2025.

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, and Xihui Liu. Ground slow, move fast: A dual-system foundation model for generalizable vision-language navigation. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

Ning Yang, Yan Huang, Kaiwen Peng, Ziheng He, Kai Wang, Cui Miao, Kailin Lyu, Guo Li, Xiaofeng Wang, Zheng Zhu, et al. Wam-nav: Asymmetric latent world-action modeling for unified visual navigation. *arXiv preprint arXiv:2606.04907*, 2026.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Zhuoyuan Yu, Yuxing Long, Zihan Yang, Chengyan Zeng, Hongwei Fan, Jiyao Zhang, and Hao Dong. Correctnav: Self-correction flywheel empowers vision-language-action navigation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18737–18745, 2026.

Hai Zhang, Hang Yu, Junqiao Zhao, Di Zhang, Hongtu Zhou, Chang Huang, Chen Ye, et al. How to fine-tune the model: Unified model shift and model bias policy optimization. *Advances in Neural Information Processing Systems*, 36:59252–59272, 2023.

Hai Zhang, Boyuan Zheng, Tianying Ji, Jinhang Liu, Anqi Guo, Junqiao Zhao, and Lanqing Li. Scrutinize what we ignore: Reining in task representation shift of context-based offline meta reinforcement learning. In *International Conference on Learning Representations*, 2025a.

Jianjun Zhang, Jian Zhu, Taiyi Su, Chong Ma, Zitai Huang, Yi Xu, and Hanli Wang. Learning 4d geometric priors for inference-efficient world action models. *arXiv preprint arXiv:2607.05468*, 2026a.

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

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. *arXiv preprint arXiv:2509.12129*, 2025b.

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. *Robotics: Science and Systems*, 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.

Baining Zhao, Jiacheng Xu, Weicheng Feng, Xin Zhang, Zhaolu Wang, Haoyang Wang, Shilong Ji, Ziyou Wang, Jianjie Fang, Zhiheng Zheng, et al. Worldvln: Autoregressive world action model for aerial vision-language navigation. *arXiv preprint arXiv:2605.15964*, 2026.

Sunyao Zhou, Yunzi Wu, Tianhang Wang, Xinhai Li, Guang Chen, Lizheng Liu, Chenjia Bai, and Xuelong Li. Deconav: Dialog enhanced long-horizon collaborative vision-language navigation. *arXiv preprint arXiv:2604.12486*, 2026a.

Yang Zhou, Xiaofeng Wang, Hao Shao, Letian Wang, Guosheng Zhao, Jiangnan Shao, Jiagang Zhu, Tingdong Yu, Zheng Zhu, Guan Huang, et al. Drivedreamer-policy: A geometry-grounded world-action model for unified generation and planning. *arXiv preprint arXiv:2604.01765*, 2026b.

A Implementation Details

This appendix provides implementation details omitted from the main text, including model configuration, action processing, training-data construction, optimization settings, and the complete Stage-III reward. Benchmark-specific evaluation protocols are provided in Appendix B. Unless stated otherwise, we follow the notation of Sec. 3.

A.1 Architecture and Input Configuration

WNM-2D and WNM-3D share the same world-action backbone, prediction targets, block-causal attention layout, history-sampling rule, and receding-horizon execution procedure. Their architectural difference is the representation of observed visual context. WNM-2D uses the backbone's native VAE-encoded RGB-history pathway, whereas WNM-3D constructs the clean history-conditioning prefix $\mathbf{C}_t$ with the frozen geometry encoder E_{geo} and trainable 3D Scene-to-Token Adapter T_ϕ.

At each replanning step, both variants receive a history $\mathcal{H}_t$ of $K = 33$ RGB observations sampled uniformly from the inclusive interval $[I_0, I_t]$. The world-action backbone operates at 160×320. WNM-3D additionally resizes the same sampled history to 512×512 before applying VGGT-Ω. The prediction horizon comprises $BN_v = 32$ future RGB frames and $BN_a = 32$ actions, partitioned into $B = 4$ aligned blocks with $N_v = 8$ frames and $N_a = 8$ actions per block.

Table 5: **Architecture configuration.** Shared world-action settings and WNM-3D-specific geometry and Adapter settings.

Symbol	Configuration	Value
–	World-action initialization	Wan2.2-TI2V-5B
K	History length	33
–	World-action input resolution	160×320
–	Geometry-encoder input resolution	512×512
B	Number of visual/action blocks	4
N_v	Future RGB frames per visual block	8
N_a	Actions per block	8
–	Selected VGGT-Ω encoder blocks	5, 12, 18, 24
L	Number of selected encoder blocks	4
d_s	Common adapter width	512
$K \times H_s \times W_s$	Source-memory lattice	$33 \times 32 \times 32$
$T_c \times H_c \times W_c$	Target lattice	$9 \times 5 \times 10$
N_c	History-prefix length (WNM-2D / WNM-3D)	450 / 450
R	Deformable resampling layers	2
–	Resampling heads / points per head	8 / 8
–	Spatiotemporal refinement blocks	2
–	Trainable adapter parameters	23.97M

3D Scene-to-Token Adapter configuration. For each selected encoder block l, the adapter projects $\mathbf{F}_t^{(l)}$ to width d_s and applies the content-adaptive encoder-level fusion in Eq. (7). The fused source memory is pooled onto a $T_c \times H_c \times W_c$ target lattice to initialize the query content. A structured embedding $\mathbf{E}_{\mathrm{struct}}$—the sum of learned global, temporal, row, and column slots, a Fourier embedding of target coordinates, and a learned query-type embedding—is then added to the linearly projected pooled content. Each of the $R = 2$ anchored deformable-resampling layers uses 8 attention heads with 8 sampling points per head. Two factorized spatiotemporal blocks further refine the queries before the detail and coarse branches are projected to the DiT width d.

WNM-2D prefix. WNM-2D uses the backbone's native VAE-encoded RGB-history prefix and does not include E_{geo} or T_ϕ. Its 33 RGB history frames produce 9 VAE latent frames, each contributing a 5×10 patch grid. The resulting prefix therefore contains $9 \times 5 \times 10 = 450$ tokens of width $d = 3072$, matching the prefix length of WNM-3D. Neither WNM variant receives depth, camera pose, a bird's-eye-view observation, or an explicit metric map at inference time.

A.2 Action Encoding, Decoding, and Execution

The policy represents the physical increments in Eq. (1) using the backbone's fixed-width action vector. Before flow matching, the three physical dimensions are multiplied by $s_a = 4$, normalized with training-set action quantiles, and zero-padded to width $d_a = 32$. At inference, the inverse transform is applied to the first three predicted dimensions. Let $\widehat{\mathbf{a}}_j^{(t)} \in [-1, 1]^3$ denote the unpadded normalized prediction associated with $a_j^{(t)}$. We decode it as

$$a_j^{(t)} = \frac{1}{s_a}\left[\frac{\widehat{\mathbf{a}}_j^{(t)} + 1}{2} \odot (\mathbf{q}_{99} - \mathbf{q}_{01}) + \mathbf{q}_{01}\right], \qquad s_a = 4, \tag{13}$$

where $\mathbf{q}_{01}$ and $\mathbf{q}_{99}$ are the checkpoint-associated 1% and 99% action quantiles.

The action coordinate system is anchored at the camera pose associated with the current observation I_t. Simulator pose is used only to transform predicted local increments for execution and reward computation and is never exposed to the policy. For $j \in \{0, \ldots, 31\}$, the world position after applying action j is

$$\mathbf{p}_{j+1}^{\text{world}} = \mathbf{p}_0^{\text{world}} + R_t \sum_{u=0}^{j} \left(\Delta x_u^{(t)}, \Delta y_u^{(t)}\right), \tag{14}$$

where R_t is the planar camera-to-navigation rotation at I_t. All planar increments are expressed in this fixed local frame; predicted yaw increments are accumulated separately and do not recursively rotate the XY increments.

For chunk $c \in \{1, \ldots, B\}$, let its physical XY path length be

$$L_c^{xy} = \sum_{j=(c-1)N_a}^{cN_a-1} \sqrt{\left(\Delta x_j^{(t)}\right)^2 + \left(\Delta y_j^{(t)}\right)^2}. \tag{15}$$

Stage-III stopping credit uses the following ordered STOP detector. If $L_c^{xy} < 0.15\,\text{m}$, STOP is emitted before any action in the chunk is executed. Otherwise, actions are inspected in temporal order, and the first action satisfying

$$\max\left(\left|\Delta x_j^{(t)}\right|, \left|\Delta y_j^{(t)}\right|, \left|\Delta \psi_j^{(t)}\right|\right) \le 10^{-3}$$

emits STOP immediately before that action is applied. If neither condition is satisfied, the chunk does not terminate. The path-length comparison is strict, i.e., $< 0.15\,\text{m}$ rather than $\le 0.15\,\text{m}$.

During Stage-III reward computation, this detector is evaluated independently for all four action chunks to construct counterfactual chunk-wise stopping credit. At receding-horizon inference, execution is based only on the first predicted action chunk before replanning. Collision-aware reward computation distinguishes the raw generated plan from a deployment-simulated execution of the first chunk. The first chunk is executed against a four-pixel-dilated occupancy map: when a segment contacts the dilated obstacle set, execution is clipped to the last free point and the remaining predicted increments continue from that actual position rather than terminating the chunk. A separate two-pixel-dilated raw-plan check is retained for hard-collision diagnostics and supplies the hard-collision event for later counterfactual chunks. Independently, every raw chunk is scored by a soft-clearance risk over the two-to-six-pixel warning band. Recovery credit is deployment-specific: only the first chunk can receive a bounded recovery bonus, evaluated from its contact-clipped execution using a six-pixel tail-clearance requirement together with escape-distance and post-collision STOP checks. All collision, clearance, and recovery terms remain inside the navigation reward; no separate collision GRPO stream is enabled. Full definitions are provided in Sec. A.4.3, and stopping definitions are given in Sec. A.4.4.

A.3 Stage-I and Stage-II Training

Unless otherwise stated, the Stage-I and Stage-II data statistics below correspond to GN-Bench training; the VLN-CE data configuration is described in Appendix B.2.

Stage-I: Offline Expert SFT. We construct the offline training set from 16K instruction-conditioned trajectories generated by an A* expert. For each sampled trajectory, a valid replanning timestep is selected uniformly, and the observation context and subsequent 32-step visual-action targets are constructed at that timestep. Each WNM variant is trained for 20 epochs with $\mathcal{L}_{\mathrm{WA}}$ from Eq. (6).

Stage-II: Closed-loop DAgger-SFT. Starting from its Stage-I checkpoint, each WNM variant is rolled out on the same 16K navigation tasks. At every policy-visited state, the A* expert provides a corrective action sequence, and the simulator renders the trajectory-consistent future observations. These policy-induced samples form the Stage-II DAgger datasets, which contain approximately 691K chunks for WNM-2D and 633K chunks for WNM-3D. Each variant is then optimized for 5 epochs with $\mathcal{L}_{\mathrm{WA}}$.

Table 6: **GN-Bench Stage-I and Stage-II training configuration.** Data and optimization settings for offline expert SFT and closed-loop DAgger-SFT.

Configuration	Stage-I	Stage-II
Data source	A* demonstrations	Policy-specific DAgger
Tasks / trajectories	16K	16K tasks
WNM-2D chunks	–	691K
WNM-3D chunks	–	633K
Epochs	20	5
Optimizer	AdamW	AdamW
Initial learning rate	1×10^{-5}	1×10^{-5}
Warmup ratio	5%	5%
Weight decay	1×10^{-5}	1×10^{-5}
Global batch: WNM-2D / WNM-3D	256 / 192	256 / 192
GPUs	16 H100	16 H100

Flow-matching timesteps and loss weighting. Stages I and II use the same discrete training schedule with 1,000 indices. For each future block b, we sample

$$k_b \sim \mathrm{Uniform}\{0, \ldots, 999\}, \quad \bar{\sigma}_{k_b} = 1 - \frac{k_b}{1000}, \\ \sigma_b = \frac{5\bar{\sigma}_{k_b}}{1 + 4\bar{\sigma}_{k_b}}. \tag{16}$$

The two VAE latent frames representing the eight RGB frames in block b and the corresponding eight action steps share σ_b; their visual and action Gaussian noise samples remain independent. The block-zero latent corresponding to the current target-prefix frame is assigned an independently sampled index from the same schedule.

The timestep-dependent weight in Eq. (6) is defined on this shifted schedule as

$$q_k = \exp\left[-2\left(\sigma_k - \frac{1}{2}\right)^2\right], \quad q_{\min} = \min_{0 \le r \le 999} q_r, \\ w_k = \frac{1000\,(q_k - q_{\min})}{\sum_{j=0}^{999}(q_j - q_{\min})}. \tag{17}$$

Thus, the weights have unit mean over the 1,000 training indices. MSE_w denotes the mean elementwise squared velocity error after weighting each block by its corresponding w_k. For $\mathrm{MSE}_{w,\mathbf{M}_a}$, the mask $\mathbf{M}_a$ sets the squared residuals of zero-padded action dimensions to zero before the same weighting and reduction are applied. We use equal coefficients for the visual and action terms, i.e., $\lambda_a = 1$.

Stages I and II use AdamW with an initial learning rate of 1×10^{-5}, $\beta_1 = 0.95$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay 1×10^{-5}, and a cosine schedule with a 5% warm-up and a zero terminal learning rate. Gradients are clipped to a maximum norm of 1.0. Training uses BF16 arithmetic, TF32 matrix multiplication, and 16 NVIDIA H100 GPUs. The global batch size is 256 for WNM-2D and 192 for WNM-3D; the latter is reduced to accommodate geometry-aware history encoding.

A.4 STAGE-III COUNTERFACTUAL DANCEGRPO

For GN-Bench, Stage-III optimization uses about 96K fixed policy-generated records collected in simulation for each configuration. For the direct A* SFT–DanceGRPO ablation, we use records from the same Stage-I-policy collection used to construct the DAgger-SFT dataset and initialize DanceGRPO directly from the Stage-I checkpoint, without performing the DAgger-SFT update. For the full A* SFT–DAgger-SFT–DanceGRPO curriculum, we instead recollect the Stage-III records using the resulting Stage-II policy and initialize DanceGRPO from the corresponding Stage-II checkpoint. Each record contains simulator-rendered future observations, an expert trajectory, occupancy information, and navigation metadata. These fixed annotations define the visual, navigation, and stopping reward streams; the simulator is not stepped during Stage-III optimization, and no auxiliary supervised $\mathcal{L}_{\mathrm{WA}}$ term is used.

A.4.1 DENOISING-STRATUM COUNTERFACTUAL SAMPLING

Following Sec. 3.5, we partition the eligible denoising transitions into K_s strata $\{\mathcal{S}_k\}_{k=1}^{K_s}$. The final configuration uses $K_s = 4$ and

$$\mathcal{S}_1 = \{0, 1, 2\}, \qquad \mathcal{S}_2 = \{6\}, \qquad \mathcal{S}_3 = \{10\}, \qquad \mathcal{S}_4 = \{13, 14, 15\}, \tag{18}$$

where each element is a zero-based transition index in the 16-step sampler. At every optimization step, one transition $\tau_k \in \mathcal{S}_k$ is sampled from each stratum, and the resulting set $\{\tau_k\}_{k=1}^{K_s}$ is shared by the entire condition batch.

For every conditioning instance i and stratum k, we generate a group of $H = 4$ counterfactual branches $h \in \{1, 2, 3, 4\}$. The four branches share their initial visual and action latents and all exogenous SDE increments except at τ_k, where branch-specific visual and action noise is injected. Each conditioning instance therefore produces $HK_s = 4K_s = 16$ complete world-action samples. Rewards are evaluated on the completed world-action samples, but gradients are replayed only through the intervened transition τ_k. Visual, navigation, and stopping rewards are standardized independently within each four-branch group according to Eq. (10). If all valid branches in a group have identical rewards, their advantages are zero.

The visual and action transitions use independent Gaussian SDE noise factors with stochasticity coefficients

$$\eta_v = 0.70, \qquad \eta_a = 0.20. \tag{19}$$

Transition log densities are reduced over their event dimensions before forming the modality-specific likelihood ratios r^v and r_c^a. As described in Sec. 3.5, the visual reward is routed through r^v, whereas the navigation and stopping rewards of chunk c are routed through r_c^a.

For all reward streams, we use the standard clipped surrogate

$$\ell_{\mathrm{clip}}(r, A; \varepsilon) = -\min\left(rA,\ \mathrm{clip}\left(r, 1-\varepsilon, 1+\varepsilon\right) A\right), \tag{20}$$

with modality-specific clipping thresholds given in Table 10. The action density is reduced only over the three deployed dimensions $(\Delta x, \Delta y, \Delta\psi)$ of each chunk; padded action dimensions do not contribute to the policy ratio.

A.4.2 VISUAL REWARD

The visual reward combines a base future-view quality score with a visual-side flow-action consistency bonus:

$$R^v = R_{\text{vision-base}} + R^v_{\text{flow}}. \tag{21}$$

The base score evaluates the generated future frames in terms of structural similarity, pixel reconstruction, and temporal consistency:

$$Q_v = 0.60 S_{\text{Pyr-SSIM}} + 0.25\left(1 - E_{\text{charb}}\right) + 0.15\left(1 - E_{\text{temp}}\right), \tag{22}$$

where $S_{\text{Pyr-SSIM}}$ is our pyramid-averaged SSIM score. For the T evaluated frames, it is defined as

$$S_{\text{Pyr-SSIM}} = \frac{1}{4T} \sum_{m=1}^{4} \sum_{t=1}^{T} \mathrm{SSIM}\left(\mathcal{Y}_t^{(m)}, \mathcal{Y}_{t,\mathrm{GT}}^{(m)}\right), \tag{23}$$

where scale $m = 1$ corresponds to 64×128, and each subsequent scale is obtained by 2×2 average pooling with stride 2. This score is an arithmetic average of per-scale SSIM values and should not be interpreted as the standard multiplicative MS-SSIM index. E_{charb} is the Charbonnier reconstruction error, and E_{temp} is the Charbonnier error between predicted and ground-truth temporal differences. Both error terms use $\epsilon = 10^{-3}$ and are clipped to $[0, 1]$. A degradation factor $D_{\text{deg}} \in [0, 1]$ downweights invalid, frozen, low-contrast, exposure-corrupted, flickering, or excessively dynamic generations:

$$R_{\text{vision-base}} = D_{\text{deg}} Q_v. \tag{24}$$

The target-prefix frame is excluded from the future-frame quality and degradation statistics. After checking both sequences for finite values and rejecting predictions with more than 1% of values outside $[0, 1]$, the remaining predicted and ground-truth RGB frames are clipped to $[0, 1]$ and bilinearly resized with antialiasing to 64×128. Let $\widetilde{\mathcal{Y}}_t^{\text{pred}}, \widetilde{\mathcal{Y}}_t^{\text{GT}} \in [0, 1]^{3\times 64\times 128}$ denote these resized future frames. We define the predicted and ground-truth motion magnitudes as the mean absolute RGB change between adjacent future frames,

$$\begin{aligned} M_{\text{pred}} &= \operatorname*{mean}_{t=2,\dots,T;\, c,h,w} \left| \widetilde{\mathcal{Y}}^{\text{pred}}_{t,c,h,w} - \widetilde{\mathcal{Y}}^{\text{pred}}_{t-1,c,h,w} \right|, \\ M_{\text{GT}} &= \operatorname*{mean}_{t=2,\dots,T;\, c,h,w} \left| \widetilde{\mathcal{Y}}^{\text{GT}}_{t,c,h,w} - \widetilde{\mathcal{Y}}^{\text{GT}}_{t-1,c,h,w} \right|. \end{aligned} \tag{25}$$

For the flicker check, let

$$\mu_t^{\text{pred}} = \operatorname*{mean}_{c,h,w} \widetilde{\mathcal{Y}}^{\text{pred}}_{t,c,h,w}, \qquad \mu_t^{\text{GT}} = \operatorname*{mean}_{c,h,w} \widetilde{\mathcal{Y}}^{\text{GT}}_{t,c,h,w}, \tag{26}$$

and define the maximum adjacent-frame mean-intensity changes

$$\Delta_{\text{int}}^{\text{pred}} = \max_{t=2,\dots,T} \left| \mu_t^{\text{pred}} - \mu_{t-1}^{\text{pred}} \right|, \qquad \Delta_{\text{int}}^{\text{GT}} = \max_{t=2,\dots,T} \left| \mu_t^{\text{GT}} - \mu_{t-1}^{\text{GT}} \right|. \tag{27}$$

Thus, M_{pred}, M_{GT}, $\Delta_{\text{int}}^{\text{pred}}$, and $\Delta_{\text{int}}^{\text{GT}}$ are dimensionless statistics in normalized RGB units; they are not optical-flow magnitudes. Within the future-frame quality and degradation statistics, the excluded target-prefix frame is used only for the separate prefix-copy diagnostic; the flow-action term below additionally uses it to form the first adjacent-frame optical-flow pair. The per-frame RGB spatial standard deviation used by the constant-video and low-contrast checks is computed on the clipped future prediction before the 64×128 resize, whereas the mean-intensity and saturation-ratio statistics are computed on the resized future prediction.

We set $D_{\text{deg}} = 0$ for non-finite predictions, predictions with more than 1% of values outside $[0, 1]$, or predictions whose mean per-frame RGB-value standard deviation is below 0.01. Otherwise, D_{deg} is initialized to 1.0. It is capped at 0.10 for a frozen video, defined by $M_{\text{GT}} \geq 0.01$ and $M_{\text{pred}} < \max(0.002, 0.10 M_{\text{GT}})$, or for prefix copying, defined by a mean ground-truth prefix change of at least 0.02 and a mean prediction-to-prefix error of at most 0.01. The factor is capped at 0.25 if the prediction has an RGB-value standard deviation below 0.03, a mean RGB intensity outside $[0.03, 0.97]$, more than 85% of its values at or below 0.02 or at or above 0.98, $\Delta_{\text{int}}^{\text{pred}} > \max(0.08, 3\Delta_{\text{int}}^{\text{GT}})$, or $M_{\text{pred}} > \max(0.12, 3M_{\text{GT}})$. If several conditions apply, we use the smallest factor.

Because receding-horizon execution is based on $\mathcal{A}_{t,1}$ before replanning, the flow-action term is evaluated on the generated visual block aligned with $\mathcal{A}_{t,1}$. Let

$$\Delta \mathbf{p}_a = \sum_{j=0}^{N_a - 1} \left(\Delta x_j^{(t)}, \Delta y_j^{(t)} \right) \in \mathbb{R}^2 \tag{28}$$

denote the cumulative XY displacement of the executed action block. The decoded sequence used for this comparison contains the current target-prefix frame followed by the aligned visual block's $N_v = 8$ generated future frames. These nine frames are converted to 64×128 grayscale images, and forward and backward DIS flow is computed for the resulting eight adjacent-frame pairs. The target-prefix frame is used only to form the first pair and is excluded from the future-frame visual-quality score. For each pair, we form a descriptor from the median (u, v) flow in each cell of a 4×6 grid and the global p50, p75, and p90 flow-magnitude statistics; φ_{flow} is the mean of these eight descriptors.

For each Stage-III configuration, a ridge regressor is fitted once on 480 randomly sampled ground-truth clips from its corresponding training records and kept fixed during optimization to map optical-flow descriptors to local camera motion:

$$\widehat{\mathbf{a}}_{\text{flow}} = W_{\text{flow}}\varphi_{\text{flow}} + \mathbf{b}_{\text{flow}} \in \mathbb{R}^3, \tag{29}$$

which estimates $(\Delta x, \Delta y, \Delta\psi)$. We define $\widehat{\Delta\mathbf{p}}_{\text{flow}} \in \mathbb{R}^2$ as its first two components; the yaw component is used only for diagnostics.

For a moving action block, the directional agreement is

$$s_{xy} = \max\left(0, \frac{\Delta\mathbf{p}_a^{\top}\widehat{\Delta\mathbf{p}}_{\text{flow}}}{\|\Delta\mathbf{p}_a\|_2\|\widehat{\Delta\mathbf{p}}_{\text{flow}}\|_2}\right), \tag{30}$$

provided $\|\Delta\mathbf{p}_a\|_2 \geq \delta_a$ and $\|\widehat{\Delta\mathbf{p}}_{\text{flow}}\|_2 \geq \delta_f$. If the action block is near stationary, $\|\Delta\mathbf{p}_a\|_2 < \delta_a$, we instead measure whether the generated video is also approximately stationary:

$$s_{xy} = \exp\left[-\left(\frac{m_{90}}{\delta_{\text{stat}}}\right)^2\right], \tag{31}$$

where m_{90} is the p90 component of the temporally averaged flow descriptor. When the action requests clear motion but the inferred visual translation is below δ_f, we set $s_{xy} = 0$.

For adjacent-frame pair q, flow confidence combines forward–backward consistency and image-texture reliability. The video-level confidence is

$$C_{\text{flow}} = \text{median}_{q=1,\ldots,8}\left(C_{\text{fb},q}C_{\text{tex},q}\right), \tag{32}$$

where $C_{\text{fb},q}$ is the fraction of consistent pixels in pair q. Specifically, for a forward vector $\mathbf{f}(\mathbf{x})$ and backward flow $\mathbf{b}$ sampled at $\mathbf{x} + \mathbf{f}(\mathbf{x})$, a pixel is consistent only if the destination is in bounds and $\|\mathbf{f}(\mathbf{x}) + \mathbf{b}(\mathbf{x} + \mathbf{f}(\mathbf{x}))\|_2 \leq 1.0 + 0.05\|\mathbf{f}(\mathbf{x})\|_2$ pixels. The texture statistic g_q is the mean magnitude of the 3×3 Sobel gradient in the preceding grayscale frame, and $C_{\text{tex},q} = \text{clip}(g_q/8, 0, 1)$. The final consistency score and reward contribution are

$$S_{\text{flow}-\text{act}} = s_{xy}C_{\text{flow}}, \qquad R^v_{\text{flow}} = \lambda^v_{\text{flow}}D_{\text{deg}}S_{\text{flow}-\text{act}}, \qquad \lambda^v_{\text{flow}} = 0.05. \tag{33}$$

The inferred yaw component does not contribute to $S_{\text{flow}-\text{act}}$ or either flow-action reward branch. The same gated score $D_{\text{deg}}S_{\text{flow}-\text{act}}$ also provides the action-side first-chunk bonus defined in Appendix A.4.3.

Table 7: **Visual-reward configuration.** Parameters used for future-view quality and visual-action consistency in Stage-III.

Symbol	Description	Value
$w_{\text{pyr-ssim}}$	Pyramid-SSIM weight	0.60
w_{charb}	Charbonnier reconstruction weight	0.25
w_{temp}	Temporal-consistency weight	0.15
λ^v_{flow}	Visual flow-action weight	0.05
δ_a	Action-motion threshold	0.05 m
δ_f	Flow-motion threshold	0.01 m
δ_{stat}	Stationary-flow scale	0.10 px
–	Flow input resolution	64×128
–	Flow spatial pooling grid	4×6
–	Flow descriptor statistics	Grid medians + p50/p75/p90
–	Flow calibration samples	480 ground-truth clips

A.4.3 Navigation Reward

For each valid action chunk $c \in \{1, \ldots, B\}$, the navigation reward combines signed geodesic progress, path-length agreement, goal-potential improvement, goal entry, collision and clearance

shaping, route adherence, forward motion, orientation consistency, the action-side flow-action bonus, and a navigation-side penalty for an executed premature STOP:

$$\begin{aligned}\widetilde{R}_c^{\mathrm{nav}} = {} & w_P P_c + w_L S_c^{\mathrm{len}} + w_G G_c + w_{\mathrm{entry}} I_c^{\mathrm{entry}} + R_{c,\mathrm{flow}}^{a} + R_{c,\mathrm{premature}}^{\mathrm{nav}} \\ & - w_{\mathrm{hard}} I_c^{\mathrm{hard}} - w_{\mathrm{repeat}} N_c^{\mathrm{repeat}} - w_{\mathrm{clear}} Q_c^{\mathrm{clear}} + w_{\mathrm{rec}} R_c^{\mathrm{rec}} \\ & - w_D D_c - w_B B_c - Y_c^{\mathrm{yaw}},\end{aligned} \tag{34}$$

followed by

$$R_c^{\mathrm{nav}} = \mathrm{clip}\left(\widetilde{R}_c^{\mathrm{nav}}, -1, 1\right). \tag{35}$$

The signed geodesic-progress term is

$$P_c = \mathrm{clip}\left(\frac{d_c^s - d_c^e}{\max\left(L_{c,\mathrm{gt}}^{\mathrm{nav}}, \delta_L\right)}, -1, 1\right), \tag{36}$$

where d_c^s and d_c^e are the geodesic distances from the endpoints of the navigation-scored portion of chunk c to the goal. This portion is truncated at the first goal-entry transition or an earlier execution termination. Accordingly, $L_{c,\mathrm{gt}}^{\mathrm{nav}}$ is the ground-truth path length over exactly these navigation-active transitions. The path-length agreement is

$$S_c^{\mathrm{len}} = \exp\left[-\left(\frac{|L_c^{\mathrm{pred}} - L_c^{\mathrm{gt}}|}{\gamma_L L_c^{\mathrm{gt}} + \beta_L}\right)^2\right], \tag{37}$$

where L_c^{pred} and L_c^{gt} are, respectively, the predicted and ground-truth path lengths over the full fixed horizon of N_a transitions. In particular, L_c^{pred} includes all actions originally predicted for chunk c, including actions subsequently removed by STOP or collision resolution. Thus, L_c^{gt} differs from $L_{c,\mathrm{gt}}^{\mathrm{nav}}$ whenever navigation credit terminates before the end of the chunk; the two coincide otherwise. The goal-potential improvement is

$$G_c = \Phi(d_c^e) - \Phi(d_c^s), \qquad \Phi(d) = \exp(-d/\tau_g). \tag{38}$$

Let $j_{\mathrm{enter}} > 0$ denote the first predicted position inside the goal radius. We set $I_c^{\mathrm{entry}} = 1$ only for $c = 1 + \lfloor (j_{\mathrm{enter}} - 1)/N_a \rfloor$; a trajectory initialized inside the goal region does not receive an entry bonus. Navigation credit in the entry chunk terminates at j_{enter}, and subsequent executed transitions are assigned to the stopping stream.

Flow-action and premature-STOP terms. The gated flow-action score from Appendix A.4.2 also provides action-side credit to the first chunk:

$$R_{c,\mathrm{flow}}^{a} = \mathbf{1}[c = 1]\, \lambda_{\mathrm{flow}}^{a} D_{\mathrm{deg}} S_{\mathrm{flow-act}}, \qquad \lambda_{\mathrm{flow}}^{a} = 0.05. \tag{39}$$

This term is set to zero when action decoding is invalid or when the first chunk is inactive. The flow-inferred yaw remains diagnostic only and does not enter $S_{\mathrm{flow-act}}$. An active chunk containing an actually executed incorrect STOP additionally receives

$$R_{c,\mathrm{premature}}^{\mathrm{nav}} = -\left(\alpha_{\mathrm{nav}} + \beta_{\mathrm{nav}} q(d_{c,\mathrm{geo}}^{\mathrm{stop}})\right), \qquad \alpha_{\mathrm{nav}} = 0.25, \quad \beta_{\mathrm{nav}} = 0.75, \tag{40}$$

and $R_{c,\mathrm{premature}}^{\mathrm{nav}} = 0$ otherwise. Here, $q(\cdot)$ is the geodesic-distance scale defined in Eq. (58). This penalty is part of the navigation stream and is applied before the navigation reward is clipped.

Collision, clearance, and recovery shaping. The collision terms distinguish the raw generated plan from the deployment-simulated execution used for the first chunk. For $c = 1$, let n_1^{coll} be the number of hard contacts encountered when executing the first chunk against the four-pixel-dilated occupancy map with last-free-on-segment clipping and continued execution after contact. The hard-contact indicator is

$$I_c^{\mathrm{hard}} = \begin{cases} \mathbf{1}\left[n_1^{\mathrm{coll}} > 0\right], & c = 1, \\ \mathbf{1}[\text{raw chunk } c \text{ collides under the 2-pixel margin}], & c > 1. \end{cases} \tag{41}$$

Repeated-contact credit is deployment-specific:

$$N_c^{\mathrm{repeat}} = \begin{cases} \min\left(\max(n_1^{\mathrm{coll}} - 1, 0), 2\right), & c = 1, \\ 0, & c > 1. \end{cases} \tag{42}$$

The two-pixel raw-plan collision check is also retained as a diagnostic for the first chunk.

Soft-clearance shaping is evaluated on the raw generated XY polyline of every active chunk, independently of the contact-clipped deployment execution. Let $\mathcal{P}_c$ denote the raster cells traversed by the raw polyline after excluding its fixed initial cell, and let $\kappa_{c,r}$ be the Euclidean obstacle clearance, in occupancy-map pixels, at cell $r \in \mathcal{P}_c$. With hard and soft margins $m_{\mathrm{hard}} = 2$ px and $m_{\mathrm{soft}} = 6$ px, the quadratic clearance risk is

$$Q_c^{\mathrm{clear}} = \max_{r \in \mathcal{P}_c} \left[\mathrm{clip}\left(\frac{m_{\mathrm{soft}} - \kappa_{c,r}}{m_{\mathrm{soft}} - m_{\mathrm{hard}}}, 0, 1 \right) \right]^2 . \tag{43}$$

Following the implementation, $Q_c^{\mathrm{clear}} = 1$ if the raw path leaves the occupancy-map extent, and $Q_c^{\mathrm{clear}} = 0$ if no raster cell beyond the fixed start is traversed.

Recovery credit is defined only for the deployment-simulated first chunk. After the last contact, let e_1 be the Euclidean distance from the contact-clipped position to the final executed position, and let κ_1^{tail} be the minimum raw-map clearance over the final two collision-free executed endpoints. A recovery is eligible only if the first chunk has at least one contact, is neither start-invalid nor out of bounds, has at least two collision-free actions after the last contact, satisfies $e_1 \geq 0.20\,\mathrm{m}$ and $\kappa_1^{\mathrm{tail}} \geq 6$ px, and does not emit a post-collision premature STOP. For an eligible recovery,

$$R_1^{\mathrm{rec}} = \min \left[\mathrm{clip}\left(\frac{e_1}{0.40\,\mathrm{m}}, 0, 1 \right), \mathrm{clip}\left(\frac{\kappa_1^{\mathrm{tail}} - 4}{6 - 4}, 0, 1 \right) \right], \qquad R_c^{\mathrm{rec}} = 0 \text{ for } c > 1, \tag{44}$$

and $R_1^{\mathrm{rec}} = 0$ otherwise. Thus, the first hard contact receives weight $w_{\mathrm{hard}} = 0.60$, at most two additional contacts are penalized with weight $w_{\mathrm{repeat}} = 0.10$ each, raw-path soft-clearance risk is weighted by $w_{\mathrm{clear}} = 0.25$, and an eligible first-chunk recovery receives the bounded bonus $w_{\mathrm{rec}} R_1^{\mathrm{rec}}$ with $w_{\mathrm{rec}} = 0.10$. These terms remain part of the navigation reward because the separate collision-credit stream is disabled.

Route and orientation terms. Let $\mathcal{I}_c$ denote the position indices of the navigation-active predicted trajectory supplied to the route diagnostic for chunk c, from the chunk start through its navigation endpoint inclusive. The endpoint is truncated at goal entry or an earlier execution termination; for the first chunk, the positions come from the four-pixel deployment-simulated trajectory, while later chunks use their corresponding predicted trajectories. If no navigation transition is active, the implementation sets $D_c = 0$. Otherwise $n_c = |\mathcal{I}_c| \geq 2$. For $i \in \mathcal{I}_c$, let d_i^{route} denote the distance from the predicted position to the nearest segment of the complete ground-truth route. The normalized route-deviation term is

$$D_c = \mathrm{clip}\left(\frac{\sqrt{\frac{1}{n_c} \sum_{i \in \mathcal{I}_c} [(d_i^{\mathrm{route}} - 0.75\,\mathrm{m})_+]^2}}{1.50\,\mathrm{m}}, 0, 1 \right). \tag{45}$$

For predicted steps longer than 0.03 m, B_c is the mean of $\max(-\cos \angle(\Delta \mathbf{p}_i, \mathbf{t}_i^{\mathrm{GT}}), 0)$, where $\mathbf{t}_i^{\mathrm{GT}}$ is the tangent of the ground-truth route segment nearest the predicted step midpoint. We set $B_c = 0$ if no step satisfies the motion threshold. For the deployment-simulated first chunk, transitions marked by the collision-recovery grace mask are excluded from B_c so that bounded recovery motion is not penalized as backward progress.

For orientation consistency, let θ_i^a be the cumulative predicted action yaw immediately before action i, θ_i^p the predicted-path heading relative to the first predicted XY step, and θ_i^{GT} the corresponding ground-truth path heading. Both path-heading sequences are formed before applying the navigation aggregation masks. If the first XY step is numerically stationary ($\|\Delta \mathbf{p}_0\|_2 < 10^{-6}$), the canonical positive-X direction is used as the reference; subsequent steps with norm below 10^{-6} retain the most recent valid heading. The path-heading increment aligned with action i is the forward difference

$$\Delta \theta_i^p = \mathrm{wrap}_{[-\pi,\pi)} \left(\theta_{i+1}^p - \theta_i^p \right), \tag{46}$$

which is computed in radians and has the same action index as $\Delta \psi_i$.

The final Stage-III run uses the linear-guard yaw score

$$\ell(e; a, b) = \mathrm{clip}\left(\frac{|\,\mathrm{wrap}_{[-\pi,\pi)}(e)|_{\mathrm{deg}} - a}{b - a}, 0, 1 \right), \tag{47}$$

where $|\cdot|_{\deg}$ converts the wrapped angular magnitude to degrees. Let $\mathcal{J}_c$ be the navigation-active action indices in chunk c whose predicted XY translation satisfies $\|\Delta\mathbf{p}_i\|_2 > 0.03$ m. Path and rate consistency are evaluated only on $\mathcal{J}_c$. Their per-action errors are

$$e_i^{\text{path}} = \text{wrap}_{[-\pi,\pi)}(\theta_i^a - \theta_i^p), \qquad e_i^{\text{rate}} = \text{wrap}_{[-\pi,\pi)}(\Delta\psi_i - \Delta\theta_i^p). \tag{48}$$

For the collision-recovery grace transitions of the deployment-simulated first chunk, path consistency remains active but uses the folded path-error magnitude

$$\bar{e}_i^{\text{path}} = \min\Big(|\,\text{wrap}(e_i^{\text{path}})|, \Big|\pi - |\,\text{wrap}(e_i^{\text{path}})|\Big|\Big), \tag{49}$$

while the rate error is unchanged. Outside recovery grace, $\bar{e}_i^{\text{path}} = |\,\text{wrap}(e_i^{\text{path}})|$. Hence the path and rate scores are

$$z_i^{\text{path}} = \ell(\bar{e}_i^{\text{path}}; 10^\circ, 45^\circ), \qquad z_i^{\text{rate}} = \ell(e_i^{\text{rate}}; 8^\circ, 35^\circ). \tag{50}$$

Gross-GT orientation credit additionally supervises pure-yaw controls. Let $\mathcal{K}_c$ be the navigation-active action indices satisfying either $\|\Delta\mathbf{p}_i\|_2 > 0.03$ m or $|\Delta\psi_i| > 0.01$ rad, and let $\mathcal{G}_c$ denote the collision-recovery grace indices ($\mathcal{G}_c = \varnothing$ for chunks without deployment recovery). For a pure-yaw action, i.e., $\|\Delta\mathbf{p}_i\|_2 \le 0.03$ m and $|\Delta\psi_i| > 0.01$ rad, the implementation evaluates the GT-heading error after applying the yaw increment; otherwise it evaluates the error before the action:

$$e_i^{\text{gross}} = \begin{cases} \text{wrap}_{[-\pi,\pi)}(\theta_{i+1}^a - \theta_i^{\text{GT}}), & \|\Delta\mathbf{p}_i\|_2 \le 0.03\,\text{m},\ |\Delta\psi_i| > 0.01\,\text{rad}, \\ \text{wrap}_{[-\pi,\pi)}(\theta_i^a - \theta_i^{\text{GT}}), & \text{otherwise.} \end{cases} \tag{51}$$

The per-action gross score is $z_i^{\text{gross}} = \ell(e_i^{\text{gross}}; 12^\circ, 60^\circ)$ and is aggregated only over $\mathcal{K}_c \setminus \mathcal{G}_c$; recovery-grace controls are therefore excluded from gross-GT orientation credit. Path and rate remain active on $\mathcal{J}_c$ during recovery grace, with only the path error folded as above.

Chunk-level yaw scores use the spike-sensitive mean–max aggregation

$$\mathcal{A}_\rho(\{z_i\}_{i\in\mathcal{S}}) = (1-\rho)\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} z_i + \rho \max_{i\in\mathcal{S}} z_i, \qquad \rho = 0.50, \tag{52}$$

with value zero when $\mathcal{S}$ is empty. We then set

$$\begin{aligned} Y_c^{\text{path}} &= \mathcal{A}_{0.50}\Big(\{z_i^{\text{path}}\}_{i\in\mathcal{J}_c}\Big), \\ Y_c^{\text{rate}} &= \mathcal{A}_{0.50}\big(\{z_i^{\text{rate}}\}_{i\in\mathcal{J}_c}\big), \\ Y_c^{\text{gross}} &= \mathcal{A}_{0.50}\big(\{z_i^{\text{gross}}\}_{i\in\mathcal{K}_c\setminus\mathcal{G}_c}\big). \end{aligned} \tag{53}$$

Finally, the three weighted yaw terms are capped jointly:

$$Y_c^{\text{yaw}} = \min\big(\kappa_{\text{yaw}}, w_{\text{path}}Y_c^{\text{path}} + w_{\text{rate}}Y_c^{\text{rate}} + w_{\text{gross}}Y_c^{\text{gross}}\big), \qquad \kappa_{\text{yaw}} = 0.24, \tag{54}$$

with $w_{\text{path}} = 0.16$, $w_{\text{rate}} = 0.10$, and $w_{\text{gross}} = 0.08$. Equation (34) subtracts this non-negative capped penalty once, matching the implementation's negative yaw penalty with a lower bound of -0.24.

Table 8: **Navigation-reward coefficients.** Coefficients, yaw-guard settings, flow-action credit, premature-STOP feedback, and collision-shaping settings used for Stage-III navigation credit.

Symbol	Description	Value	Symbol	Description	Value
w_P	Geodesic-progress weight	0.70	w_{path}	Path-orientation weight	0.16
w_L	Path-length agreement weight	0.15	w_{rate}	Yaw-rate weight	0.10
w_G	Goal-potential weight	0.15	w_{gross}	Gross-GT yaw weight	0.08
w_{entry}	Goal-entry bonus	0.15	κ_{yaw}	Total yaw-penalty cap	0.24
w_D	Route-deviation penalty	0.35	ρ_{yaw}	Mean–max mix	0.50
w_B	Backward-motion penalty	0.30	–	Yaw score mode	linear guard
δ_L	Progress normalization floor	0.50m	$(a_{\text{path}}, b_{\text{path}})$	Path guard	$10^\circ/45^\circ$
γ_L	Path-length scale	0.50	$(a_{\text{rate}}, b_{\text{rate}})$	Rate guard	$8^\circ/35^\circ$
β_L	Path-length offset	0.10m	$(a_{\text{gross}}, b_{\text{gross}})$	Gross-GT guard	$12^\circ/60^\circ$
τ_g	Goal-potential scale	0.75m	$\delta_{\text{xy}}^{\text{yaw}}$	Translation-motion floor	0.03 m
λ_{flow}^a	Action flow-action weight	0.05	$\delta_{\text{rot}}^{\text{yaw}}$	Rotation-motion floor	0.01 rad
α_{nav}	Premature-STOP nav base	0.25	β_{nav}	Premature-STOP nav add.	0.75
w_{hard}	Hard-contact penalty	0.60	–	Repeat-contact cap	2
w_{repeat}	Repeat-contact penalty	0.10	–	Hard diagnostic margin	2 px
w_{clear}	Soft-clearance penalty	0.25	–	Deployment execution margin	4 px
w_{rec}	Recovery-bonus weight	0.10	–	Recovery clearance threshold	6 px
–	Soft-clearance margin	6 px	–	Recovery tail-free steps	2
–	Recovery min./full escape	0.20 / 0.40 m	–	–	–

A.4.4 Stopping Reward

Stage-III applies the ordered STOP detector independently to every chunk of $N_a = 8$ actions. If the chunk's cumulative XY path length is strictly below 0.15 m, STOP is emitted at the chunk-start position. Otherwise, the detector selects the first action satisfying $\max(|\Delta x_j|, |\Delta y_j|, |\Delta \psi_j|) \le 10^{-3}$ and emits STOP immediately before that action is applied. A STOP in an earlier hypothetical chunk does not mask the independent STOP supervision of later chunks.

After the predicted trajectory enters the goal region, the stop-well term contains only a motion-energy penalty. Let j_c^{end} denote the exclusive upper transition index actually scored for chunk c after STOP, collision, and other execution truncations are applied. Define

$$E_j = \frac{(\|\mathbf{p}_{j+1} - \mathbf{p}_j\|_2 - \delta_{xy})_+}{r_g} + w_\psi \frac{(|\Delta\psi_j| - \delta_\psi)_+}{\tau_\psi}, \tag{55}$$

with $r_g = 1.5$ m, $\delta_{xy} = 0$, $w_\psi = 0.18$, $\delta_\psi = 0.02$ rad, and $\tau_\psi = 0.25$ rad. The continuous stop-well reward is

$$R_{c,\text{well}} = -w_E \sum_{\substack{j=(c-1)N_a,\dots,cN_a-1 \\ j \ge j_{\text{enter}},\ j < j_c^{\text{end}}}} E_j, \qquad w_E = 0.20. \tag{56}$$

Thus, unlike the navigation term G_c in Eq. (38), the final stopping stream contains no separate goal-potential improvement term.

For chunk c, let $\mathbf{p}_c^{\text{start}} = \mathbf{p}_{(c-1)N_a}$ denote its start position. If the low-translation branch emits STOP, the stop position is exactly $\mathbf{p}_c^{\text{start}}$. More generally, let j_c^{stop} denote the position index at which the ordered detector emits an executable STOP event and let $\mathbf{p}_c^{\text{stop}}$ denote the corresponding position; for a componentwise near-zero action, this is the position immediately before that action is applied. STOP success is determined by Euclidean distance to the goal center,

$$\|\mathbf{p}_c^{\text{stop}} - \mathbf{g}\|_2 \le r_g, \tag{57}$$

whereas premature-STOP severity uses the occupancy-geodesic distance $d_{c,\mathrm{geo}}^{\mathrm{stop}}$ from the same stop position to the goal. Define

$$q(d) = 1 - \exp\left[-\frac{(d - r_g - \delta_{\mathrm{fail}})_+}{\tau_{\mathrm{fail}}}\right], \qquad \delta_{\mathrm{fail}} = 0.25\,\mathrm{m}, \quad \tau_{\mathrm{fail}} = 2.0\,\mathrm{m}, \tag{58}$$

with $q(d) = 1$ for a disconnected geodesic query. The hard STOP reward is

$$R_{c,\mathrm{hard}} = \begin{cases} r_{\mathrm{succ}}, & \text{an executable STOP is emitted and } \|\mathbf{p}_c^{\mathrm{stop}} - \mathbf{g}\|_2 \le r_g, \\ -\left(\alpha_{\mathrm{fail}} + \beta_{\mathrm{fail}} q(d_{c,\mathrm{geo}}^{\mathrm{stop}})\right), & \text{an executable STOP is emitted outside the goal region}, \\ 0, & \text{no executable STOP is emitted}, \end{cases} \tag{59}$$

where $r_{\mathrm{succ}} = 0.15$, $\alpha_{\mathrm{fail}} = 0.75$, and $\beta_{\mathrm{fail}} = 0.25$, giving an incorrect-STOP reward in $[-1.00, -0.75]$. The same geodesic scale supplies the navigation-side premature-STOP term in Eq. (40); that term uses $\alpha_{\mathrm{nav}} = 0.25$ and $\beta_{\mathrm{nav}} = 0.75$ but is not part of the stopping stream.

Leaving the goal region after entry incurs an indicator penalty I_c^{exit} with coefficient $w_{\mathrm{exit}} = 1.0$. The complete stopping reward is

$$R_c^{\mathrm{stop}} = \mathrm{clip}\left(R_{c,\mathrm{well}} + R_{c,\mathrm{hard}} - w_{\mathrm{exit}} I_c^{\mathrm{exit}}, -1, 1\right). \tag{60}$$

Table 9: **Stopping-reward configuration.** Parameters used for goal settling, STOP detection, and distance-scaled premature-STOP supervision in Stage-III.

Symbol	Description	Value	Symbol	Description	Value
r_g	Goal radius	1.5 m	δ_{chunk}	Chunk STOP threshold	0.15 m
δ_{xy}	Translation dead zone	0 m	ϵ_{stop}	Componentwise STOP threshold	10^{-3}
w_ψ	Yaw-energy weight	0.18	δ_{fail}	Premature-STOP deadband	0.25 m
δ_ψ	Yaw dead-zone threshold	0.02 rad	τ_{fail}	Premature-STOP distance scale	2.0 m
τ_ψ	Yaw normalization scale	0.25 rad	α_{fail}	Failure base magnitude	0.75
w_E	Action-energy weight	0.20	β_{fail}	Failure distance add.	0.25
r_{succ}	STOP success reward	0.15	–	Incorrect-STOP range	$[-1.00, -0.75]$
w_{exit}	Goal-exit penalty	1.0	–	Stop-reward clip	$[-1, 1]$

A.4.5 Chunk Aggregation and Stage-III Objective

We aggregate the chunk-wise navigation and stopping losses defined in Eq. (11) as

$$\mathcal{L}_{\mathrm{nav}} = \sum_{c=1}^{B} \alpha_c \mathcal{L}_{\mathrm{nav},c}, \qquad \sum_{c=1}^{B} \alpha_c = 1, \tag{61}$$

and

$$\mathcal{L}_{\mathrm{stop}} = \frac{1}{B} \sum_{c=1}^{B} \mathcal{L}_{\mathrm{stop},c}. \tag{62}$$

The navigation weights emphasize early chunks because receding-horizon inference executes $\mathcal{A}_{t,1}$ before replanning. For $B = 4$, we use

$$(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = \frac{(8, 4, 2, 1)}{15}. \tag{63}$$

Substituting Eqs. (61) and (62) into Eq. (12) gives the implemented Stage-III objective, which contains no auxiliary $\mathcal{L}_{\mathrm{WA}}$ term.

A.4.6 STAGE-III OPTIMIZATION SETTINGS

Table 10: **Stage-III optimization hyperparameters.** Main settings for Counterfactual Dance-GRPO.

Symbol	Setting	Value	Symbol	Setting	Value
–	Global condition batch size	64	η_v	Visual SDE coefficient	0.70
HK_s	Counterfactual branches / condition	16	η_a	Action SDE coefficient	0.20
–	Sampler transitions	16	ϵ	Advantage stabilizer	1×10^{-4}
–	Eligible sampler transitions	8	ε_v	Visual clipping threshold	5×10^{-4}
–	Replayed branch transitions / condition	16	ε_a	Action clipping threshold	0.01
–	CFG scale	5	–	Optimizer	AdamW
–	Learning rate	5×10^{-6}	–	Stage-III updates	700
λ_{act}	Action-side weight	0.25	λ_{stop}	Stopping weight	0.50
$\{\alpha_c\}_{c=1}^{B}$	Navigation chunk weights	$(8, 4, 2, 1)/15$			

B BENCHMARK EVALUATION DETAILS

B.1 GN-BENCH CLOSED-LOOP EVALUATION

We evaluate every WNM checkpoint on the GN-Bench Seen and Unseen splits using the same receding-horizon inference procedure. The Seen and Unseen splits contain 1,000 and 5,000 episodes, respectively. The evaluator executes at most the first 8 actions before replanning. It emits STOP when the cumulative translational path length of this block, $\sum_{j=0}^{7} \|\Delta \mathbf{p}_j\|_2$, is below 0.15 m; an individual action whose three components have absolute value at most 10^{-3} also terminates the episode. These deployment checks are shared by WNM-2D and WNM-3D. Each checkpoint is evaluated once per episode with inference seed 1140. Navigation Error, Oracle Success, Success Rate, Trajectory Length, and SPL follow the benchmark definitions. We report point estimates over all episodes and do not compute multi-seed standard deviations or bootstrap confidence intervals.

Additional qualitative rollouts. Figure 5 shows representative closed-loop WNM-3D rollouts on GN-Bench.



Figure 5: **Representative closed-loop WNM-3D rollouts on GN-Bench.** Each row shows a language instruction together with egocentric observations and the evolving navigation trajectory.

B.2 VLN-CE Evaluation

We evaluate WNM-2D and WNM-3D on the R2R Val-Unseen split of VLN-CE (Krantz et al., 2020). For benchmark-specific training, both variants use the R2R (Anderson et al., 2018) and RxR (Ku et al., 2020) training splits and follow the complete three-stage curriculum described in Sec. 3. In Stage-I, the dataset-provided reference paths are used as expert trajectories for supervised world-action learning. In Stage-II, corrective trajectories at policy-visited states are generated by a reference-aware navmesh oracle. When the current position is within 1 m of the reference route, the oracle replans from the current position directly to the goal using Habitat-Sim's navmesh shortest-path query; otherwise, it first replans to a progress-aligned checkpoint on the reference route and then resumes the stored Stage-I reference trajectory. The implementation uses `habitat_sim.ShortestPath` and `pathfinder.find_path`; the resulting paths provide trajectory supervision rather than discrete actions from `ShortestPathFollower`. Stage-III then applies Counterfactual DanceGRPO to policy-specific rollouts. At inference, both variants receive only monocular egocentric RGB observations and language instructions. Each predicted action block contains eight local motion increments. A map-free kinematic beam-search controller tracks the predicted XY polyline using discrete turn and forward primitives, of which only the first four are executed before replanning; STOP is handled separately as described below. Episode horizon and other environment-side execution settings follow the official VLN-CE protocol. Neither model receives depth, camera pose, a bird's-eye-view observation, or an explicit metric map at inference time. Navigation Error, Oracle Success, Success Rate, and SPL follow the official benchmark definitions. Unless otherwise stated, optimizer, schedule, and Stage-III hyperparameter settings follow Appendices A.3 and A.4; dataset sizes and rollout counts are benchmark-specific.

Action adaptation. WNM predicts eight local motion increments $(\Delta x, \Delta y, \Delta\psi)$ per action block. We accumulate the predicted $(\Delta x, \Delta y)$ components directly in the current query frame, without additional smoothing, to form a target XY polyline and track it with a map-free kinematic beam-search controller. Each expansion consists of zero or more same-direction turns followed by a 0.25 m forward move, with turn candidates spanning up to $\pm 180^\circ$; turns are discretized at 15° on R2R and 30° on RxR. We use a beam width of 32 and search up to four forward moves. For a candidate containing $H \leq 4$ forward moves, the tracking cost is

$$J = \sum_{h=0}^{H-1} 0.7^h \left[\frac{1}{5} \sum_{r=1}^{5} d_{h,r}^2 + d_{h,5}^2 + 2 \times 10^{-5} |n_h| - 10^{-3} \Delta s_h \right],$$

where $d_{h,r}$ measures the distance to the target polyline at 20%, 40%, 60%, 80%, and 100% of the h-th forward move, n_h is the number of preceding discrete turns, and Δs_h is the corresponding arc-length progress. Polyline association is constrained to move monotonically forward, with the admissible arc length capped by the starting progress plus the simulated travel distance and 0.015 m. The predicted yaw increment is not used in this tracking cost. The inference-time controller uses no map, navmesh query, obstacle cost, or ground-truth goal distance.

Execution and stopping. Only the first four discrete primitives of the selected sequence are executed before replanning; these may consist entirely of in-place turns. A collision during execution clears the remaining primitive queue and triggers an immediate model query. STOP is handled independently of beam search. The agent stops immediately if the cumulative translational path length of the predicted block is below 0.15 m or if all three components of the first predicted action have magnitude at most 10^{-3}. A later near-zero action truncates the target motion prefix; STOP is appended only when the selected plan reaches that prefix endpoint with at most 0.1 m remaining arc length and 0.2 m terminal distance. Reaching the beam-search horizon alone does not trigger STOP.

C Real-World Deployment Details

Deployment setup. We deploy WNM-3D on a laboratory-developed mobile robot equipped with an Intel RealSense D456 camera, from which only the monocular RGB stream is provided to the policy. Model inference runs on a remote GPU cluster as a server, while an NVIDIA Jetson Orin onboard the robot runs a ROS 2 client that sends navigation requests to the server through an SSH-tunneled TCP connection and receives the predicted action chunks for execution. A ROS 2 motion-execution node converts each predicted local pose increment $(\Delta x, \Delta y, \Delta\psi)$ into short-horizon base commands, which are tracked by the robot's low-level controller before the next replanning step. No additional onboard sensor signals are provided to WNM-3D. Action decoding and policy stopping follow Appendix A.2.

Real-world adaptation. Starting from the simulation-trained Stage-III checkpoint, WNM-3D is further fine-tuned for 8,000 optimization steps on 210 self-collected real-world navigation trajectories using the supervised world-action objective $\mathcal{L}_{\mathrm{WA}}$ from Stage-I. We update the same trainable components as in Stage-I, namely the 3D Scene-to-Token Adapter and the world-action backbone, while keeping VGGT-Ω frozen. The adaptation scenes are disjoint from the scenes used in the qualitative real-world demonstrations.