OMNIMAPNAV: DUAL GROUNDING AND SEMANTIC FORESIGHT FOR MAP-CENTRIC MULTIMODAL NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual navigation should support intuitive goal specification and efficient adaptation to unfamiliar environments. However, PointNav, image-goal navigation, and conventional route-following vision-language navigation require coordinates, reference images, or route descriptions that users may not readily provide; ObjectNav category labels leave ambiguity among target instances. Without a global spatial prior, inferring unfamiliar layouts from partial observations can require extensive exploration in large, multi-room environments. To bridge this gap, we introduce OMNIMAPNAVBENCH, where one overhead RGB map serves as an interactive goal interface and a navigation prior. Its 50,241 episodes across 4,724 scenes pair map-selected points, hand-drawn route sketches, and language grounded in map–room, map–object, and object–object relations for evaluating individual cues and their combinations. Effective map utilization requires grounding local observations and heterogeneous goals in this common spatial frame. To this end, we propose OMNIMAPNAV: its Dual-Grounding Map Reader estimates an agent pose-heading belief and a goal anchor field to retrieve agent- and goal-relevant map tokens. Furthermore, a Semantic Foresight Predictor estimates one-step, action-conditioned next-observation features; a semantic foresight scorer combines these with grounded context to refine action scores. OMNIMAPNAV achieves an average geodesic success rate of 37.31% at 3 m versus 26.22% for the strongest baseline, with gains across all four goal conditions and stricter thresholds. Real-robot experiments demonstrate map-based goal specification and closed-loop navigation. Code is available at https://anonymous.4open.science/r/omnimapnav-9DDF.

1 INTRODUCTION

Visual navigation is a fundamental capability of embodied agents, enabling them to reach intended destinations and providing a foundation for subsequent interaction with the physical world. PointNav, image-goal navigation, vision-language navigation (VLN), and ObjectNav offer complementary goal interfaces (Anderson et al., 2018a; Zhu et al., 2017; Anderson et al., 2018b; Batra et al., 2020), with progress in visual representation learning and imitation-based policy training (Yadav et al., 2023; Ramrakhya et al., 2023). ViNT and NoMaD learn transferable policies from diverse robot experience (Shah et al., 2023b; Sridhar et al., 2024). Foundation models expand language reasoning and visual understanding (Achiam et al., 2023; Liu et al., 2023; Bai et al., 2025), motivating generalist instruction-following navigators (Zheng et al., 2024; Zhang et al., 2024) and semantic exploration systems (Yokoyama et al., 2024; Yin et al., 2025). Two practical challenges remain: accessible goal specification and efficient adaptation to unfamiliar environments.

From the interaction perspective, common navigation tasks require goal information that users may find difficult to provide. PointNav (Wijmans et al., 2019) assumes metric coordinates, image-goal navigation (Sridhar et al., 2024) requires a reference view of the destination, and conventional route-following VLN (Krantz et al., 2020) relies on descriptions of intermediate movements and landmarks. ObjectNav simplifies the interface to category names but leaves ambiguity among same-category instances. From the environmental adaptation perspective, general navigation knowledge does not reveal the layout of a particular building. Without a global spatial prior, robots must discover that

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.

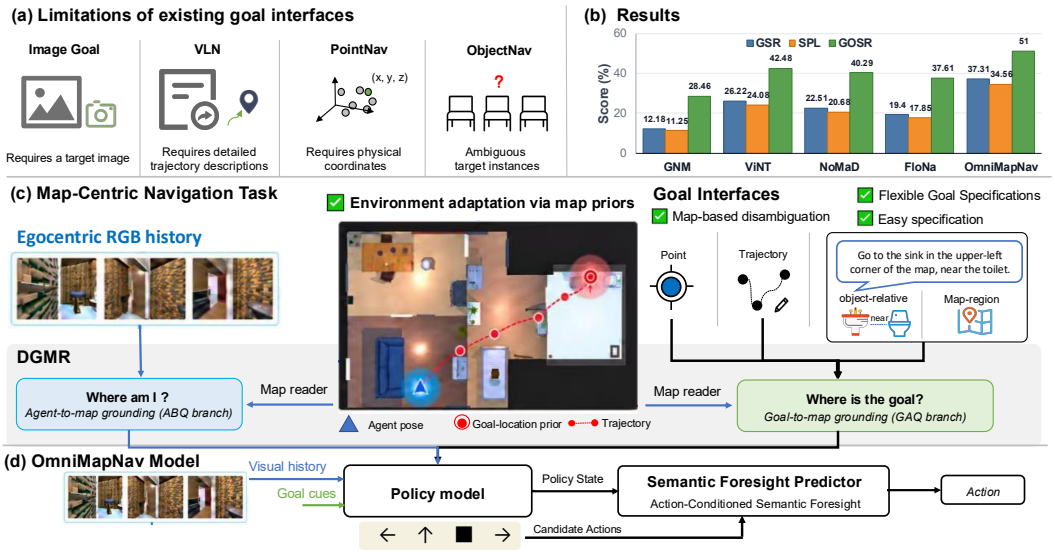


Figure 1: **Map-centric navigation.** (a, c) Goal interfaces. (b) Mean GSR, SPL, and GOSR at 3 m across four cues. (d) OMNIMAPNAV.

layout from partial egocentric observations. In large, multi-room environments, this can entail extensive exploration of irrelevant regions and substantial detours.

These challenges motivate a navigation paradigm that combines *intuitive goal expression for users with readily usable environmental context for robots*. People consult maps to identify destinations and understand routes through unfamiliar buildings without first exploring every location. An overhead RGB map could serve both roles for robot navigation: users point, sketch, or describe a destination in a visible spatial reference, while the robot obtains information about rooms and routes beyond its current view. FloNa demonstrates the value of floor plans for navigation to specified positions (Li et al., 2025). We extend position-based goals to multiple map-centric expressions, studying how individual cues and their combinations guide closed-loop navigation (Figure 1).

To bridge this gap, we introduce OMNIMAPNAVBENCH, where a single overhead RGB map serves as both an interactive goal interface and a navigation prior. It supports *Point*, a destination selected on the map; *Trajectory*, a coarse hand-drawn route sketch; and *Language*, a destination description grounded in map-relative spatial relationships. Language relates rooms to map regions, locates objects within the map and its rooms, and distinguishes targets through object-to-object relations. For example, "the table beside the sofa in the room at the top right of the map" combines room-level and object-level constraints to identify an intended instance without prescribing a sequence of movements. Cues can be used individually or together, with Combined supplying all three. The benchmark pairs these cues for each scene–start–target configuration across 50,241 episodes in 4,724 indoor scenes. This enables controlled analysis of how destination precision, route guidance, and relational descriptions affect navigation, and when combining cues provides additional benefit.

Effective map utilization, however, requires more than supplying map features to a policy. The robot must ground partial egocentric observations in an overhead view despite viewpoint differences, repeated appearances, and motion. It must also associate heterogeneous goal cues with the appropriate map regions. These correspondences have different temporal properties: the agent's pose changes during execution, whereas the destination remains anchored. Grounding the agent and goal on the map identifies relevant spatial evidence, but does not explicitly account for how candidate actions will change the next observation. This motivates combining grounded map retrieval with action-conditioned semantic prediction to inform immediate action selection.

To this end, we propose OMNIMAPNAV, a navigation framework combining dual grounding with one-step semantic foresight. We design a *Dual-Grounding Map Reader* (DGMR) with two complementary query groups. *Agent Belief Queries* (ABQ) combine visual matching with motion history to maintain a pose-heading belief and retrieve agent-grounded map tokens. *Goal Anchor Queries* (GAQ) associate the available cues with a goal anchor field and retrieve goal-grounded map tokens. Furthermore, we introduce a *Semantic Foresight Predictor* (SFP) that predicts next-observation semantic features for

each candidate action. A semantic foresight scorer combines these predictions with the grounded policy representation to refine immediate action scores. Grounding thus determines which map information is retrieved, while foresight helps select the next action through its anticipated perceptual consequences.

Experiments compare OMNIMAPNAV with GNM, ViNT, NoMaD, and FloNa (Shah et al., 2023a;b; Sridhar et al., 2024; Li et al., 2025). OMNIMAPNAV leads in geodesic success rate (GSR) across all four goal conditions at 1, 2, and 3 m, averaging 37.31% versus 26.22% for the strongest baseline at 3 m and 24.00% versus 12.36% at 1 m. Cue comparisons reveal the benefits of route guidance and cue combination; ablations and robot experiments support complementary grounded retrieval and semantic foresight.

Our main contributions are: **(1) A map-centric navigation benchmark.** We introduce OMNIMAPNAVBENCH, pairing map-selected points, route sketches, and relational spatial language to study individual cues and their combinations. **(2) Dual grounding and semantic foresight.** We propose OMNIMAPNAV, which combines agent-to-map and goal-to-map grounding with one-step semantic prediction for action selection. **(3) Extensive simulation and real-robot validation.** Extensive experiments demonstrate that OMNIMAPNAV consistently outperforms navigation baselines across goal conditions and success thresholds, benefits from complementary spatial grounding and semantic foresight, and enables effective map-centric navigation on a physical robot.

2 RELATED WORK

Visual navigation and multimodal goals. PointNav, ImageNav, ObjectNav, and VLN specify destinations through coordinates, images, categories, and language instructions, respectively (Wijmans et al., 2019; Zhu et al., 2017; Batra et al., 2020; Anderson et al., 2018b). GOAT-Bench evaluates category-, image-, and language-goal navigation (Khanna et al., 2024); UniGoal unifies these goal types (Yin et al., 2025), while OmniVLA supports pose, image, and language combinations (Hirose et al., 2025). However, coordinates, reference views, and detailed route descriptions can be difficult for users to provide, while category labels leave instance ambiguity. These approaches primarily perceive environments through egocentric images; without a supplied global spatial prior, navigation in unfamiliar, multi-room environments can require extensive exploration. OMNIMAPNAVBENCH addresses both challenges using an overhead RGB map for intuitive multimodal goal specification and global layout context.

Map representations and utilization. Maps support navigation as memory, reasoning context, and layout priors. GridMM and MapNav organize observation history into spatial memory (Wang et al., 2023; Zhang et al., 2025); MapGPT and semantic-map planners use maps for instruction-guided planning (Chen et al., 2024; Wang et al., 2025). VLMaps and AVLMaps associate spatial locations with language and multimodal features for goal retrieval and spatial queries (Huang et al., 2023b;a). Prebuilt floor plans support position-goal navigation in FloNa (Li et al., 2025) and instruction following in FloorPlan-VLN (Chen et al., 2026), while NavOne predicts paths on top-down maps (Zhan et al., 2026). Our task design starts from *goal expression through a shared map*. OMNIMAPNAVBENCH pairs map-selected points, route sketches, and language grounded in map–room, map–object, and object–object relations for controlled comparisons of individual and combined goals. During execution, DGMR *separately grounds the moving agent and intended destination*, retrieving map tokens through an evolving pose-heading belief and a goal anchor field.

Predictive representations for navigation. World models predict future observations or representations (Ha & Schmidhuber, 2018), supporting visual trajectory planning (Bar et al., 2025), joint image–waypoint generation (Zhou et al., 2026), and adaptive latent dynamics for VLN (Yao et al., 2025). V-JEPA 2 further demonstrates prediction and planning in latent space (Assran et al., 2025). We use prediction to complement spatial grounding: DGMR retrieves map context, SFP predicts one-step semantic features for candidate actions, and the semantic foresight scorer uses these predictions to refine immediate action scores without multi-step rollout.

3 OMNIMAPNAVBENCH: A MAP-CENTRIC NAVIGATION TASK

Given an overhead RGB map M and goal cues G, the agent must navigate to the target region and stop. At step t, it selects an action from $\pi(a_t \mid H_t, M, G)$, where H_t contains egocentric RGB

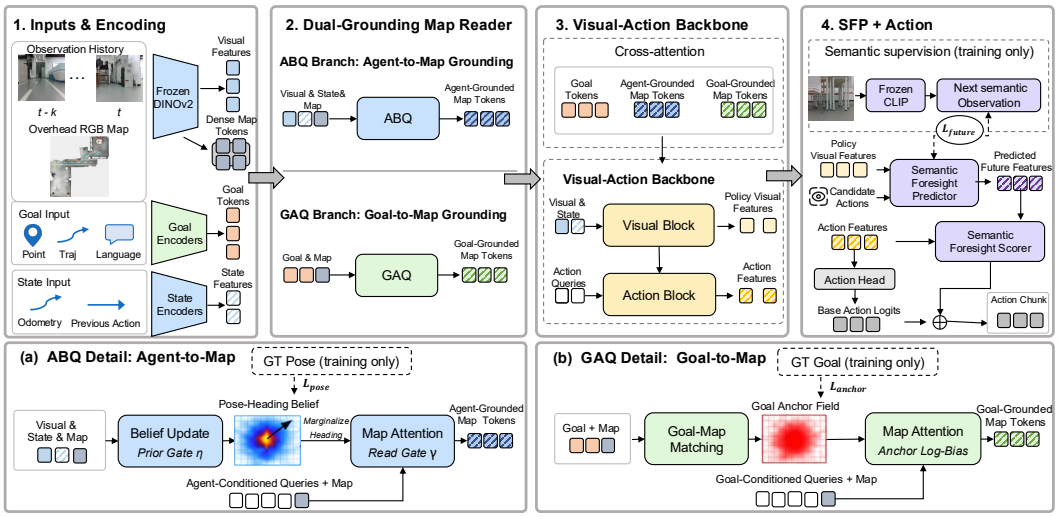


Figure 2: **OMNIMAPNAV overview.** Dual-Grounding Map Reader and Semantic Foresight Predictor.

observations $O_{1:t}$, start-relative odometry $U_{1:t}$, and previous actions. Displacements use map-aligned axes, and metric map scale is available; absolute agent position and heading, occupancy masks, and connectivity graphs are not inputs. Appendix D specifies the observation and action interface.

Goal conditions. Point supplies a normalized endpoint $g^p \in [0,1]^2$; Trajectory supplies ordered map anchors g^r along a coarse route; Language supplies a referring expression g^l with object-relative or map-region constraints. **Combined**, denoted All in tables, provides all three. These conditions refer to the same scene, start, and target. Appendix B details cue construction and sampling.

Benchmark and evaluation. OMNIMAPNAVBENCH is built in AI2-THOR and ProcTHOR (Kolve et al., 2017; Deitke et al., 2022), with 924 evaluation episodes across 126 unseen scenes. Test environments contain 3–10 rooms, and discrete-graph shortest paths range from 5.02 to 19.97 m, averaging 9.59 m. These routes require navigation beyond the initial local view, making the overhead map relevant to both destination grounding and route selection. Appendix B provides the generation pipeline, scene splits, and validation checks.

We report GSR, success weighted by path length (SPL) (Anderson et al., 2018a), geodesic oracle success rate (GOSR) (Anderson et al., 2018b), and normalized/success-weighted dynamic time warping (nDTW/sDTW). GSR@3m measures reaching the target vicinity across point, route, and language cues; GSR@1m tests precise approach and stopping. Geodesic distance accounts for obstacles. We report 1, 2, and 3 m thresholds.

4 OMNIMAPNAV: DUAL GROUNDING AND SEMANTIC FORESIGHT

4.1 OVERVIEW

DGMR retrieves agent-grounded map tokens $F_t^A = \mathcal{R}_A(H_t, M)$ from history H_t and map M, and goal-grounded tokens $F_t^G = \mathcal{R}_G(G, M)$ from cues G. The policy forms $r_t = f_\pi(H_t, G, F_t^A, F_t^G)$; the semantic foresight scorer s evaluates each future feature $\hat{Z}_{t+1,a}$ predicted by SFP:

$$\pi(a \mid H_t, M, G) \propto \exp\Big(y_a^{\text{base}}(r_t) + s(r_t, \hat{Z}_{t+1,a})\Big). \tag{1}$$

Figure 2 summarizes the framework. The interleaved backbone combines retrieved map tokens with visual history and goal tokens; foresight scoring adjusts its base logits using predicted semantic features for six candidate actions.

4.2 GOAL-CUE ENCODING AND POLICY BACKBONE

Visual and state encoding. A frozen DINOv2-S/14 (Oquab et al., 2023) encodes the observation history and overhead map into CLS and patch features. A state encoder combines odometry with

a previous-action embedding. Each observation CLS feature and its state feature form the ABQ ego representation e_t; the policy processes temporal patch features with this state conditioning.

Goal encoding. An MLP embeds Point coordinates, an order-aware Transformer (Vaswani et al., 2017) embeds Trajectory anchors, and a frozen CLIP text encoder (Radford et al., 2021) embeds Language. Projection yields three goal tokens G with a cue-presence mask m; Appendix D specifies the encoders.

Interleaved visual–action backbone. Visual and Action Blocks update observation features and action queries, respectively, cross-attending to goal tokens and map tokens $F^{\text{map}} = [F^A; F^G]$. Action features yield base logits; visual features condition SFP.

4.3 Dual-Grounding Map Reader

DGMR takes map tokens Z^M, visual–state history $E = \{e_1, \dots, e_t\}$, and goal tokens G with presence mask m. Its ABQ and GAQ branches each use four learned queries to retrieve agent-grounded and goal-grounded map tokens, respectively.

Agent-to-map grounding. The ABQ branch maintains a temporally stable belief over the agent's map pose from partial observations. Its pose-heading belief $b_t \in \Delta^{HN-1}$ is a joint distribution over $N = 1024$ map cells and $H = 12$ heading bins. A deterministic action model transports the previous posterior as $\bar{b}_t = \mathcal{T}(b_{t-1}; a_{t-1}, s_M)$, where a_{t-1} is the executed action and s_M is the map scale. Heading-conditioned cross-view matching updates this prior:

$$s_{t,h,i} = \frac{\langle W_o(e_t + q_h), W_m z_i^M \rangle}{\|W_o(e_t + q_h)\| \, \|W_m z_i^M\|}, \qquad b_t = \text{softmax}_{h,i}(\eta_t \log(\bar{b}_t + \epsilon) + s_t/\tau). \tag{2}$$

The first valid frame initializes belief from visual evidence alone. Training and evaluation apply this filter within the recent observation window; Appendix D specifies initialization and unrolling.

Marginalizing over heading gives the spatial belief $b^S_{t,i} = \sum_h b_{t,h,i}$. The ABQ queries $Q^A_t = Q^A_0 + f_A(e_t, \hat{\theta}_t)$ use the current ego state and posterior heading to retrieve agent-grounded map tokens:

$$w_{t,q} = (1 - \gamma_{t,q}) \, \text{softmax}(Q^A_{t,q} K^\top) + \gamma_{t,q} b^S_t, \qquad F^A_t = w_t V, \tag{3}$$

where $\gamma_{t,q} = \sigma(g(Q^A_{t,q}))$ and $w_{t,q}$ are the ABQ read weights. The prior gate $\eta_t \in [0, 1]$ controls temporal belief propagation; the read gate $\gamma_{t,q}$ balances content-based retrieval against the spatial belief. Ground-truth position and heading define the Gaussian training target.

Goal-to-map grounding. The GAQ branch estimates a goal anchor field through cue–map matching:

$$\ell_{k,i} = \frac{(W_g g_k)^\top (W_z z_i^M + p_i)}{\sqrt{d}}, \qquad \ell^G_i = \frac{\sum_k m_k \ell_{k,i}}{\sum_k m_k}, \tag{4}$$

Here p_i is a learned 2D map-position embedding; softmax gives the goal anchor field $p^G = \text{softmax}(\ell^G) \in \mathbb{R}^{B \times 1024}$, which biases retrieval toward cue-compatible locations.

Pooling $c^G = \sum_i p^G_i z^M_i$ conditions the GAQ queries $Q^G = Q^G_0 + f_G([\text{pool}(G, m), c^G])$, which retrieve goal-grounded map tokens $F^G = \text{Attn}(Q^G, Z^M, Z^M; \log p^G)$. The GAQ read weights are the normalized attention weights with the $\log p^G$ bias. The true endpoint defines a Gaussian anchor target for all present cues during training, but is not supplied during Language-only inference.

4.4 Action-Conditioned Semantic Foresight

Action-conditioned prediction and scoring. The future-feature predictor SFP, parameterized by P_θ, estimates a 2×2 grid of pooled CLIP features for the next observation under each candidate action. Given the latest visual features Z^V_t and action embedding e_a, it produces $\hat{Z}^F_{t+1,a} = P_\theta([Z^V_t; e_a])$. The semantic foresight scorer projects the pooled prediction to an action-specific key $k_{t,a}$ and obtains $q_t = W_q r_{t,1}$ from the first action token. Their compatibility modifies the immediate action logit:

$$\beta_{t,a} = \frac{q_t^\top k_{t,a}}{\sqrt{384}}, \qquad Y_{t,1,a} = Y^{\text{base}}_{t,1,a} + \beta_{t,a}. \tag{5}$$

The query q_t contains grounded visual and goal context, while the future keys are action-specific. The predictor is shared across all six candidates. Foresight scoring modifies only the first action in the five-step chunk; the policy executes it and predicts again after a new observation.

Future-feature supervision. The demonstrated first-action prediction is supervised by frozen CLIP features of the next observation; all six branches receive action-loss gradients through β_t. Future observations are training targets only, while the predictor and scorer remain active at inference. Appendix D specifies target construction and the cosine/smooth-L1 loss.

Training objective. The common objective is

$$\mathcal{L} = \lambda_{\text{act}}\mathcal{L}_{\text{action}} + \lambda_{\text{pose}}\mathcal{L}_{\text{pose}} + \lambda_{\text{anchor}}\mathcal{L}_{\text{anchor}} + \lambda_{\text{future}}\mathcal{L}_{\text{future}}. \tag{6}$$

$\mathcal{L}_{\text{action}}$ supervises action prediction, $\mathcal{L}_{\text{pose}}$ and $\mathcal{L}_{\text{anchor}}$ supervise spatial retrieval, and $\mathcal{L}_{\text{future}}$ supervises semantic prediction. Loss weights and optimization settings are in Appendix D.6.

5 Experiments

5.1 Experimental Setup

Each method is evaluated on 924 episodes per round under Point, Trajectory, Language, and Combined (All); Avg. is their equal-weight mean. Inputs comprise RGB history, map-aligned odometry, an overhead RGB map, and goal cues. Simulation scores average three rounds. Appendix D.6 details the protocol, and Appendix A reports per-condition ablations.

5.2 Q1: What Do Paired Goal Cues Reveal?

Table 1 compares OmniMapNav with adaptations of GNM, ViNT, NoMaD, and FloNa under each goal condition and on average. Tables 2 and 6 report four-condition averages for the ablations.

Table 1: **Navigation across goal conditions and thresholds (%).**

Model	Goal	1 m			2 m			3 m			DTW	
		GSR	SPL	GOSR	GSR	SPL	GOSR	GSR	SPL	GOSR	nDTW	sDTW
GNM (Shah et al., 2023a)	Point	3.57	3.37	9.42	7.25	6.78	18.07	11.36	10.41	28.46	54.38	9.68
	Trajectory	4.76	4.49	10.17	9.42	8.77	19.48	12.99	12.01	28.90	55.43	11.27
	Language	3.79	3.56	9.42	8.23	7.67	18.18	11.36	10.53	28.57	54.75	9.83
	All	4.76	4.47	9.74	9.74	9.08	19.16	12.99	12.04	27.92	54.89	11.25
	Avg.	4.22	3.97	9.69	8.66	8.07	18.72	12.18	11.25	28.46	54.86	10.51
ViNT (Shah et al., 2023b)	Point	9.42	8.72	16.23	17.53	16.13	30.41	24.13	21.84	42.21	60.38	21.21
	Trajectory	14.72	13.81	21.75	22.62	21.09	33.12	28.68	26.46	43.07	62.66	25.69
	Language	8.23	7.67	14.83	15.37	14.27	28.03	20.89	19.17	39.50	58.49	18.49
	All	17.10	15.95	24.46	26.08	24.16	35.39	31.17	28.84	45.13	64.37	28.14
	Avg.	12.36	11.54	19.32	20.40	18.91	31.74	26.22	24.08	42.48	61.47	23.38
NoMaD (Sridhar et al., 2024)	Point	7.47	6.97	14.39	14.72	13.66	29.00	20.24	18.45	39.72	58.69	17.53
	Trajectory	12.01	11.32	18.83	20.02	18.74	31.06	24.68	22.81	40.80	60.74	21.91
	Language	5.95	5.59	12.55	13.64	12.70	27.71	19.70	18.06	39.29	57.84	16.98
	All	11.80	11.04	19.16	20.45	19.07	32.58	25.43	23.38	41.34	61.97	22.60
	Avg.	9.31	8.73	16.23	17.21	16.04	30.09	22.51	20.68	40.29	59.81	19.75
FloNa (Li et al., 2025)	Point	6.28	5.95	13.64	12.23	11.34	26.08	19.05	17.48	37.66	57.71	16.48
	Trajectory	6.82	6.49	12.66	12.34	11.57	25.43	19.16	17.64	37.23	57.88	16.59
	Language	6.82	6.46	14.07	13.10	12.20	26.52	19.81	18.22	38.20	57.94	17.13
	All	8.23	7.77	13.53	13.53	12.57	25.76	19.59	18.03	37.34	58.06	17.18
	Avg.	7.03	6.67	13.47	12.80	11.92	25.95	19.40	17.85	37.61	57.90	16.84
OmniMapNav	Point	23.70	22.12	29.44	33.01	30.65	41.99	37.01	34.30	51.52	68.03	33.58
	Trajectory	28.79	26.83	34.63	38.10	35.44	46.32	41.56	38.56	54.55	70.76	38.17
	Language	14.29	13.41	20.78	23.81	22.15	33.55	29.00	26.85	44.16	62.81	25.85
	All	29.22	27.17	34.96	38.10	35.34	45.56	41.67	38.53	53.79	70.45	38.19
	Avg.	24.00	22.38	29.95	33.25	30.90	41.86	37.31	34.56	51.00	68.01	33.95

Consistent gains across cues and thresholds. OmniMapNav leads in GSR under every goal condition at 1, 2, and 3 m (Table 1). Against ViNT, average GSR rises from 26.22% to 37.31% at 3 m and from 12.36% to 24.00% at 1 m. Language GSR@3m improves from 20.89% to 29.00%, so the advantage is not confined to a lenient threshold or coordinate-bearing cues.

Success gains accompany better path quality. Relative to ViNT, SPL@3m rises from 24.08% to 34.56% and nDTW from 61.47% to 68.01%. Improved path efficiency and reference-route fidelity

support the map's role in guiding closed-loop execution beyond specifying an endpoint. The GOSR–GSR gap indicates that reaching the target neighborhood does not ensure successful stopping, while gains at 1 m show improvements in precise approach and termination.

Cue difficulty is consistent across several models. For ViNT, NoMaD, and OMNIMAPNAV, Trajectory yields higher GSR than Point, and Point higher GSR than Language, at all three thresholds. Trajectory also exceeds Point for GNM and FloNa, although their Language ranking differs. The route cue supplies intermediate spatial guidance that an endpoint alone does not provide. Language remains OMNIMAPNAV's weakest condition despite the constrained instruction grammar, indicating a grounding challenge beyond lexical coverage.

Combined cues achieve the highest GSR. Combined matches or exceeds OMNIMAPNAV's best single-cue GSR at every threshold. Its 41.67% at 3 m exceeds Point and Language by 4.66 and 12.67 points, but gains over Trajectory are modest: 0.43 points at 1 m, a tie at 2 m, and 0.11 points at 3 m. Extracting additional information beyond an explicit route remains a challenge for cue fusion. Figure 3 and Appendix C.2 illustrate the resulting behaviors.

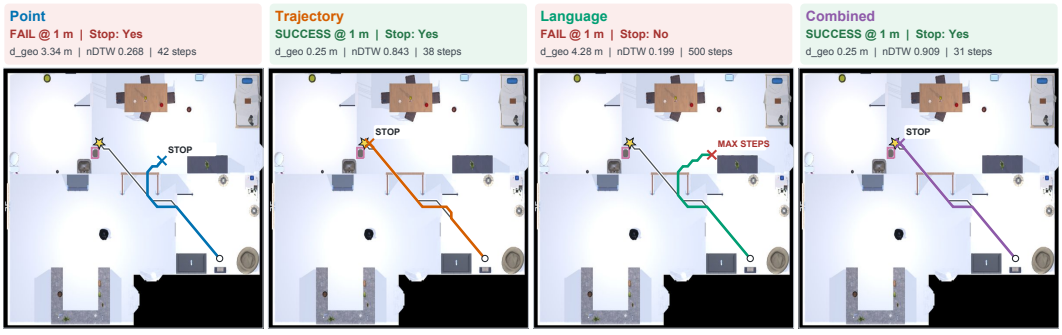


Figure 3: **Navigation to a laundry hamper under four goal cues.** Language: "Go to a laundry hamper in the bedroom." Colored: rollouts; gray: reference paths.

5.3 Q2: DOES DUAL GROUNDING IMPROVE MAP READING?

Table 2 compares dense reading, spatially supervised dense reading, single-branch readers, and DGMR. All variants are trained without the future-feature predictor or semantic foresight scorer; we additionally disable belief propagation in the DGMR checkpoint at inference.

Table 2: **Dual grounding improves navigation without semantic foresight.**

Model	1 m			2 m			3 m			DTW	
	GSR	SPL	GOSR	GSR	SPL	GOSR	GSR	SPL	GOSR	nDTW	sDTW
Dense reader	16.64	15.56	22.70	26.11	24.18	36.15	31.82	29.40	46.73	65.03	28.41
Dense + spatial sup.	19.29	17.91	24.59	27.44	25.33	37.18	32.41	29.86	46.67	64.64	28.99
ABQ Only	18.53	17.28	25.30	27.76	25.66	37.55	33.71	30.88	47.97	66.30	30.17
GAQ Only	20.54	19.14	26.22	28.19	26.19	37.09	32.44	29.94	46.78	65.65	29.31
DGMR w/o propagation	20.18	18.92	26.00	28.90	26.94	38.07	34.15	31.65	48.84	66.46	30.65
DGMR (ABQ+GAQ)	21.83	20.42	27.81	30.95	28.78	39.91	35.31	32.64	49.62	66.75	31.86

Spatial auxiliary supervision. With both DGMR branches and semantic foresight retained, pose and goal-anchor supervision improves average GSR by 4.82 points at 1 m and 3.22 at 3 m (Table 3). These gains support explicit spatial learning beyond the retrieval architecture alone. Appendix Table 13 reports all thresholds and path metrics.

Table 3: **Spatial supervision (Avg., %).**

Model	GSR		SPL		nDTW
	@1 m	@3 m	@1 m	@3 m	@3 m
w/o spatial sup.	19.18	34.09	17.86	31.47	65.03
OMNIMAPNAV	**24.00**	**37.31**	**22.38**	**34.56**	**68.01**

Spatial supervision raises the dense reader's GSR@3m from 31.82% to 32.41%. DGMR reaches 35.31%, a further 2.90 points, supporting grounded retrieval beyond auxiliary position learning. DGMR also exceeds both single-branch readers at all three thresholds. ABQ Only is stronger at 3 m and GAQ Only at 1 m, supporting their complementary roles in navigation. The dense-reader comparison separates learning spatial information from using it to select policy inputs. Auxiliary targets encourage localization, but DGMR additionally directs retrieval toward the estimated agent

and goal regions. The single-branch results suggest that destination anchoring is particularly useful for precise approach, whereas tracking the agent provides context for progress through the environment.

Does DGMR learn the intended spatial correspondences?

Table 4 evaluates localization from the two spatial estimates. The ABQ branch's mean error is 2.25 m under Combined input. The GAQ branch yields 0.38–0.40 m errors with coordinate-bearing cues, versus 3.59 m with Language. Appendix A.4.1 gives the measurement protocol.

Language's median error is 1.96 m, substantially below its mean, indicating that larger errors raise the average. By contrast, Point and Trajectory have similar mean and median errors. This motivates examining relational grounding failures alongside average navigation scores, particularly when a description must distinguish objects in different rooms.

The larger Language error aligns with its lower navigation success, linking cue-level performance to grounding accuracy. Coordinate-bearing cues yield accurate goal anchors but still require agent localization and closed-loop control. Together with Table 2, these results support both estimating spatial correspondences and using them to retrieve map tokens.

Table 4: **Grounding accuracy.**

Branch	Goal	Mean (m) ↓	Median (m) ↓	Mean (cells) ↓
ABQ	Combined	2.25	1.89	5.28
GAQ	Point	**0.38**	0.34	**0.91**
GAQ	Trajectory	0.40	0.36	0.97
GAQ	Language	3.59	1.96	8.41
GAQ	Combined	0.39	**0.34**	0.92

Temporal belief propagation. Disabling propagation at inference reduces GSR@1/2/3m by 1.65/2.05/1.16 points (Table 2). These losses support carrying spatial hypotheses across partial views and correcting them through visual updates. Appendix A reports additional thresholds and per-cue results.

Odometry sensitivity. Under All cues, independent Gaussian noise with per-axis translation / heading standard deviations of 1.5 m / 30° and 2.0 m / 40° reduces GSR@3m by 0.97/1.73 points (Table 5). The larger perturbation costs 3.90 points at 1 m, indicating greater sensitivity in precise stopping. Appendix A.4.2 details the noise protocol.

Table 5: **Odometry sensitivity (All cues).**

Noise std.	@0.5 m	@1 m	@2 m	@3 m
No-noise retest	15.69	29.33	38.10	41.23
1.5 m / 30°	13.74	26.19	35.93	**40.26**
2.0 m / 40°	13.64	25.43	35.50	**39.50**

5.4 Q3: How Does Semantic Foresight Improve Action Scoring?

Table 6 isolates future-feature supervision: both variants retain DGMR, the future-feature predictor, and the semantic foresight scorer, but one omits the future-feature loss.

Table 6: **Future-feature supervision improves grounded action selection.**

Model	1 m			2 m			3 m			DTW	
	GSR	SPL	GOSR	GSR	SPL	GOSR	GSR	SPL	GOSR	nDTW	sDTW
w/o future-feature loss	22.65	21.19	27.65	31.20	29.18	39.56	35.28	32.82	49.46	66.95	31.97
OmniMapNav	24.00	22.38	29.95	33.25	30.90	41.86	37.31	34.56	51.00	68.01	33.95

Future-feature supervision improves GSR under every cue condition at all three thresholds (Appendix Table 11). Average GSR/SPL@3m rise by 2.03/1.74 points. Gains are larger for Point and Language (2.92/2.71 points) than for Combined and Trajectory (1.52/0.97), suggesting that predicted observations are more useful for action selection without explicit route guidance. Retaining the predictor and scorer isolates the supervision's contribution from additional inference-time capacity.

Table 7: **Inference cost.**

Model	Params (M) ↓	GFLOPs ↓	Mean (ms) ↓
GNM (Shah et al., 2023a)	**97.45**	**64.95**	**21.39**
ViNT (Shah et al., 2023b)	116.33	65.78	29.73
NoMaD (Sridhar et al., 2024)	101.90	65.64	30.99
FloNa (Li et al., 2025)	104.29	70.86	31.12
Ours w/o foresight	125.48	87.62	39.11
Ours	125.48	87.67	44.60

Table 8: **Foresight scoring (All cues, %).**

Radius	GSR		SPL	
	On	Off	On	Off
0.5 m	15.69	3.14	14.55	2.96
1 m	29.33	9.96	27.32	9.45
2 m	38.10	15.26	35.36	14.53
3 m	41.23	16.67	38.19	15.87

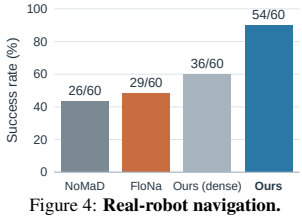


Figure 4: **Real-robot navigation.**

Inference efficiency. With a 448×448 map, semantic foresight adds 0.050 GFLOPs and 5.49 ms per step, raising latency from 39.11 to 44.60 ms (about 14%). The full model is slower than the adapted baselines (Table 7; Appendix D.5). Foresight evaluates six one-step feature predictions without multi-step rollout or planning search.

Removing semantic foresight scores at inference. Setting $\beta_{t,a} = 0$ in Eq. (5) at inference reduces All-cue GSR@3m from 41.23% to 16.67% (Table 8) and nDTW@3m from 70.30% to 57.09%. Foresight scoring changes the top-ranked action in 18.55% of decisions along score-disabled rollouts; Appendix A.4.3 gives the intervention and diagnostic protocol. The nDTW reduction implicates route execution as well as stopping. The two effects answer different questions: training without the feature loss tests the value of a semantic prediction target, whereas suppressing the learned score tests how the deployed policy uses that pathway. Together, these results support coupling observation-grounded prediction with immediate action selection.

5.5 Q4: Can the Map Interface Support Physical Deployment?

Real-world experimental setup. We deploy an AgileX TRACER with an iPhone-scanned overhead RGB map. Two RealSense D435i cameras and a RoboSense LiDAR support sensing and localization; policy inputs are RGB, odometry, map, and goal cues. Following Ehsani et al. (2024), methods are adapted on 45 human-collected trajectories and evaluated over three rounds totaling 60 Point, Trajectory, and Language trials per method, with a 1 m success radius (Appendix C.3).

Navigation in the real world. OmniMapNav (Ours) achieves 90.00% success (54/60), compared with 48.33% (29/60) for FloNa (Li et al., 2025) and 43.33% (26/60) for NoMaD (Sridhar et al., 2024) (Figure 4). Ours (dense) retains the overhead map but replaces DGMR with dense reading, reaching 60.00% (36/60). This 30-point gap supports structured retrieval beyond map availability in physical deployment, consistent with the simulation ablations.

Navigation associates onboard observations with the scanned overhead map despite differences in perspective, visibility, and appearance. Figure 5 illustrates complementary retrieval during a doorway task: ABQ read weights track robot motion, whereas GAQ read weights remain anchored to the destination, separating agent motion from the persistent goal. Appendix C provides additional visualizations; the supplementary video shows robot navigation, goal instructions, and evolving ABQ/GAQ read weights.

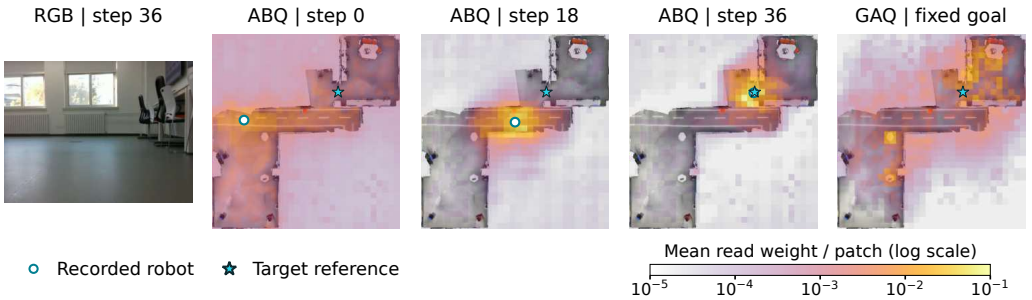


Figure 5: **Robot grounding.** Motion-tracking ABQ and goal-anchored GAQ read weights.

6 Conclusion

We introduced map-centric navigation, using one overhead RGB map for intuitive goal expression and environmental context. OmniMapNavBench pairs map-selected points, route sketches, and relational spatial language to study individual cues and their combinations. OmniMapNav combines DGMR, which retrieves agent- and goal-relevant map tokens, with SFP, which predicts next-observation features for action scoring. Experiments show consistent gains across goal conditions and success thresholds and successful real-robot deployment. Comparisons across goal conditions highlight trajectory guidance and the gains of Combined over Point or Language alone. These contributions establish a framework connecting multimodal goal specification, grounded map retrieval, and one-step semantic foresight for navigation in unfamiliar environments.

Limitations. For goal specification, Language evaluation uses a constrained grammar, motivating richer room- and object-level relations and free-form instructions. On the training side, only demonstrated actions receive direct future-feature supervision; predictions for other actions are optimized indirectly through the action loss. Training also relies mainly on simulation with limited real-robot adaptation, motivating large-scale joint training on simulated and real-world data.

Reproducibility Statement

Appendix B documents dataset generation and scene splits; Appendix D gives the model configuration and pseudocode; Appendix D.6 specifies optimization and evaluation metrics.

AI Use Statement

Generative AI tools assisted with programming, code review, manuscript revision and language polishing, and early drafts of selected figures. The authors reviewed and revised all AI-assisted code and text for technical accuracy, consistency, and clarity. Final figures were manually redrawn and edited by the authors to reflect the intended scientific content and presentation. The authors take full responsibility for the final work, including all AI-assisted code, text, claims, and figures.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018a.

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

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

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

Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020.

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

Kehan Chen, Yan Huang, Dong An, Jiawei He, Yifei Su, Jing Liu, Nianfeng Liu, and Liang Wang. Floorplan-vln: A new paradigm for floor plan guided vision-language navigation. *arXiv preprint arXiv:2603.17437*, 2026.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *Advances in neural information processing systems*, 35:5982–5994, 2022.

Kiana Ehsani, Tanmay Gupta, Rose Hendrix, Jordi Salvador, Luca Weihs, Kuo-Hao Zeng, Kunal Pratap Singh, Yejin Kim, Winson Han, Alvaro Herrasti, et al. Spoc: Imitating shortest paths in simulation enables effective navigation and manipulation in the real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16238–16250, 2024.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Noriaki Hirose, Catherine Glossop, Dhruv Shah, and Sergey Levine. Omnivla: An omni-modal vision-language-action model for robot navigation. *arXiv preprint arXiv:2509.19480*, 2025.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Audio visual language maps for robot navigation. In *International Symposium on Experimental Robotics*, pp. 105–117. Springer, 2023a.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10608–10615. IEEE, 2023b.

Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. Goat-bench: A benchmark for multi-modal lifelong navigation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16373–16383. IEEE, 2024.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

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

Jiaxin Li, Weiqi Huang, Zan Wang, Wei Liang, Huijun Di, and Feng Liu. Flona: Floor plan guided embodied visual navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 14610–14618, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Ram Ramrakhya, Dhruv Batra, Erik Wijmans, and Abhishek Das. Pirlnav: Pretraining with imitation and rl finetuning for objectnav. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17896–17906, 2023.

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

Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. *arXiv preprint arXiv:2306.14846*, 2023b.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Zehao Wang, Mingxiao Li, Minye Wu, Marie-Francine Moens, and Tinne Tuytelaars. Instruction-guided path planning with 3d semantic maps for vision-language navigation. *Neurocomputing*, 625:129457, 2025.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15579–15590. IEEE, 2023.

Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. *arXiv preprint arXiv:1911.00357*, 2019.

Karmesh Yadav, Arjun Majumdar, Ram Ramrakhya, Naoki Yokoyama, Alexei Baevski, Zsolt Kira, Oleksandr Maksymets, and Dhruv Batra. Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav. *arXiv preprint arXiv:2303.07798*, 2023.

Xuan Yao, Junyu Gao, and Changsheng Xu. Navmorph: A self-evolving world model for vision-and-language navigation in continuous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Hang Yin, Xiuwei Xu, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. Unigoal: Towards universal zero-shot goal-oriented navigation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19057–19066, 2025.

Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 42–48. IEEE, 2024.

Dijia Zhan, Jinyi Li, Chenxi Zheng, Shaoyu Huang, Yong Li, Jie Tang, and Xuemiao Xu. Navone: One-step global planning for vision-language navigation on top-down maps. *arXiv preprint arXiv:2605.06317*, 2026.

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. *arXiv preprint arXiv:2402.15852*, 2024.

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

Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13624–13634. IEEE, 2024.

Changqing Zhou, Yueru Luo, Zeyu Jiang, and Changhao Chen. Uninav: A unified world-action diffusion model for visual navigation. *arXiv preprint arXiv:2608.03244*, 2026.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 3357–3364. ieee, 2017.

APPENDIX

- **Appendix A:** Additional results and ablations.
- **Appendix B:** Dataset details.
- **Appendix C:** Qualitative results.
- **Appendix D:** Implementation and evaluation.

A ADDITIONAL RESULTS AND ABLATIONS

These results complement the baseline comparison in Table 1. Each condition uses 924 episodes evaluated over three rounds. All denotes Combined; Avg. equally weights Point, Trajectory, Language, and All. Scores are percentages, with nDTW and sDTW evaluated at 3 m.

A.1 ALL SEVEN NON-EMPTY GOAL COMBINATIONS

Table 9 evaluates every non-empty subset of Point (P), Trajectory (T), and Language (L). Avg. (4 settings) matches the main comparison; Avg. (7 settings) includes all seven combinations. Episode counts are per round.

Table 9: **Navigation with flexible goal combinations.** Evaluating all seven non-empty cue subsets reveals how combining goal information affects success and path quality.

Goal combination	1 m			2 m			3 m			DTW @ 3 m		Episodes per round
	GSR	SPL	GOSR	GSR	SPL	GOSR	GSR	SPL	GOSR	nDTW	sDTW	
Point	23.70	22.12	29.44	33.01	30.65	41.99	37.01	34.30	51.52	68.03	33.58	924 / 924
Trajectory	28.79	26.83	34.63	38.10	35.44	46.32	41.56	38.56	54.55	70.76	38.17	924 / 924
Language	14.29	13.41	20.78	23.81	22.15	33.55	29.00	26.85	44.16	62.81	25.85	924 / 924
Point + Trajectory	27.92	26.17	33.12	37.45	34.98	45.67	40.37	37.61	53.03	70.22	37.12	924 / 924
Point + Language	23.92	22.38	29.76	34.09	31.67	42.97	37.34	34.63	51.52	68.33	34.04	924 / 924
Trajectory + Language	29.98	28.07	35.82	39.94	37.22	46.97	43.18	40.14	55.41	71.03	39.72	924 / 924
All (P + T + L)	29.22	27.17	34.96	38.10	35.34	45.56	41.67	38.53	53.79	70.45	38.19	924 / 924
Avg. (4 settings)	**24.00**	**22.38**	**29.95**	**33.25**	**30.90**	**41.86**	**37.31**	**34.56**	**51.00**	**68.01**	**33.95**	**3,696 / 3,696**
Avg. (7 settings)	**25.40**	**23.74**	**31.22**	**34.93**	**32.49**	**43.29**	**38.59**	**35.80**	**51.99**	**68.80**	**35.24**	**6,468 / 6,468**

A.2 MAP-READER ABLATIONS

Table 10: **Map retrieval by goal condition.** Dense, single-branch, and dual-grounding readers reveal the contribution of structured map retrieval without semantic foresight, expanding Table 2.

Goal	1 m			2 m			3 m			DTW	
	GSR	SPL	GOSR	GSR	SPL	GOSR	GSR	SPL	GOSR	nDTW	sDTW
Dense reader											
Point	13.96	13.02	20.35	24.03	22.29	35.82	30.84	28.52	46.65	64.51	27.17
Trajectory	20.24	19.01	26.62	31.82	29.45	39.94	37.55	34.71	50.54	68.10	33.80
Language	11.36	10.58	16.88	17.32	16.07	28.79	21.86	20.21	39.29	59.61	19.30
All	21.00	19.64	26.95	31.28	28.92	40.04	37.01	34.17	50.43	67.89	33.39
Avg.	16.64	15.56	22.70	26.11	24.18	36.15	31.82	29.40	46.73	65.03	28.41
Dense + spatial sup.											
Point	13.31	12.36	19.26	21.32	19.67	32.79	29.22	26.92	44.37	63.34	25.60
Trajectory	26.52	24.58	30.74	35.06	32.33	43.40	37.66	34.62	50.87	68.12	34.28
Language	9.96	9.26	16.13	17.42	16.18	28.46	24.24	22.46	40.15	58.70	21.00
All	27.38	25.41	32.25	35.93	33.14	44.05	38.53	35.45	51.30	68.41	35.07
Avg.	19.29	17.91	24.59	27.44	25.33	37.18	32.41	29.86	46.67	64.64	28.99

Continued on next page

Table 10: **Map retrieval by goal condition (continued).**

Goal	1 m			2 m			3 m			DTW	
	GSR	SPL	GOSR	GSR	SPL	GOSR	GSR	SPL	GOSR	nDTW	sDTW
ABQ Only											
Point	15.37	14.32	22.08	25.43	23.56	36.36	32.79	30.01	46.97	65.91	29.04
Trajectory	22.08	20.66	29.33	32.58	30.19	41.23	38.42	35.31	51.41	69.19	34.76
Language	11.58	10.80	17.75	18.51	17.10	29.44	24.35	22.19	41.13	60.36	21.25
All	25.11	23.33	32.03	34.52	31.77	43.18	39.29	36.01	52.38	69.76	35.65
Avg.	18.53	17.28	25.30	27.76	25.66	37.55	33.71	30.88	47.97	66.30	30.17
GAQ Only											
Point	20.35	18.84	26.08	27.81	25.79	38.42	32.14	29.64	47.62	65.81	28.95
Trajectory	25.76	23.97	31.06	32.90	30.47	40.80	36.58	33.70	49.03	68.01	33.32
Language	11.36	10.65	17.42	19.59	18.25	28.79	24.78	22.86	41.02	60.99	21.90
All	24.68	23.11	30.30	32.47	30.26	40.37	36.26	33.55	49.46	67.79	33.05
Avg.	20.54	19.14	26.22	28.19	26.19	37.09	32.44	29.94	46.78	65.65	29.31
DGMR (ABQ+GAQ)											
Point	21.32	19.88	27.49	30.41	28.25	40.26	34.96	32.17	50.22	67.33	31.44
Trajectory	26.52	24.74	32.36	35.50	33.02	44.37	39.18	36.17	52.06	69.16	35.80
Language	12.23	11.55	18.72	21.65	20.12	31.17	27.06	25.10	42.64	61.22	23.82
All	27.27	25.53	32.68	36.26	33.73	43.83	40.04	37.12	53.57	69.28	36.39
Avg.	21.83	20.42	27.81	30.95	28.78	39.91	35.31	32.64	49.62	66.75	31.86

A.3 Future-Feature Supervision

Table 11: **Future-feature supervision across goal conditions.** With the predictor and scorer retained, supervision improves GSR for every cue condition and threshold, extending Table 6.

Goal	1 m			2 m			3 m			DTW	
	GSR	SPL	GOSR	GSR	SPL	GOSR	GSR	SPL	GOSR	nDTW	sDTW
w/o future-feature loss											
Point	21.32	19.84	26.52	30.30	28.11	38.85	34.09	31.46	48.81	66.31	30.76
Trajectory	28.35	26.64	32.25	37.34	35.17	44.59	40.58	38.12	53.68	70.23	37.11
Language	13.85	12.93	19.70	21.00	19.61	31.28	26.30	24.40	43.29	61.77	23.40
All	27.06	25.35	32.14	36.15	33.81	43.51	40.15	37.31	52.06	69.49	36.62
Avg.	22.65	21.19	27.65	31.20	29.18	39.56	35.28	32.82	49.46	66.95	31.97
OmniMapNav											
Point	23.70	22.12	29.44	33.01	30.65	41.99	37.01	34.30	51.52	68.03	33.58
Trajectory	28.79	26.83	34.63	38.10	35.44	46.32	41.56	38.56	54.55	70.76	38.17
Language	14.29	13.41	20.78	23.81	22.15	33.55	29.00	26.85	44.16	62.81	25.85
All	29.22	27.17	34.96	38.10	35.34	45.56	41.67	38.53	53.79	70.45	38.19
Avg.	24.00	22.38	29.95	33.25	30.90	41.86	37.31	34.56	51.00	68.01	33.95

A.4 Controlled Ablations

Map-reader diagnostics report four-condition averages; odometry and foresight interventions use All.

A.4.1 Map-Reader Diagnostics

Grounding accuracy. Table 4 measures Euclidean localization error by decoding the maximum-probability cell on the 32×32 map grid. ABQ position errors are averaged within trajectories and then across trajectories; the median is over episode means. GAQ goal-position error is evaluated once per trajectory, and cell errors are measured in map patches.

Belief propagation. Disabling propagation reduces GSR at all radii (1.16 points at 3 m).

Table 12: **Temporal belief propagation supports navigation.** Disabling propagation at inference reduces success across distance thresholds.

Model	0.5 m			1 m			2 m			3 m			DTW	
	GSR	SPL	GOSR	GSR	SPL	GOSR	GSR	SPL	GOSR	GSR	SPL	GOSR	nDTW	sDTW
DGMR w/o propagation	8.58	7.98	12.82	20.18	18.92	26.00	28.90	26.94	38.07	34.15	31.65	48.84	66.46	30.65
DGMR (ABQ+GAQ)	**10.61**	**9.86**	**15.10**	**21.83**	**20.43**	**27.81**	**30.95**	**28.78**	**39.91**	**35.31**	**32.64**	**49.62**	**66.75**	**31.86**

Spatial auxiliary supervision. No-Spatial-Aux sets $\lambda_{\text{pose}} = \lambda_{\text{anchor}} = 0$, retaining both DGMR branches, the future-feature predictor, and the semantic foresight scorer (Table 13).

Table 13: **Spatial supervision in the full model.**

Model	0.5 m			1 m			2 m			3 m			DTW	
	GSR	SPL	GOSR	GSR	SPL	GOSR	GSR	SPL	GOSR	GSR	SPL	GOSR	nDTW	sDTW
No-Spatial-Aux	8.60	8.03	13.10	19.18	17.86	24.70	27.98	25.92	37.18	34.09	31.47	46.59	65.03	30.56
OMNIMAPNAV	**12.34**	**11.46**	**16.26**	**24.00**	**22.38**	**29.95**	**33.25**	**30.90**	**41.86**	**37.31**	**34.56**	**51.00**	**68.01**	**33.95**

A.4.2 ODOMETRY PERTURBATION PROTOCOL

Perturbation protocol. At each new observation, including the first, we add independent noise $\epsilon_t^x, \epsilon_t^z \sim \mathcal{N}(0, \sigma_p^2)$ and $\epsilon_t^\theta \sim \mathcal{N}(0, \sigma_\theta^2)$, with $(\sigma_p, \sigma_\theta) = (1.5\,\text{m}, 30^\circ)$ or $(2.0\,\text{m}, 40^\circ)$. The translation standard deviation is per axis. Noise precedes odometry normalization and encoding; historical frames retain their first noise sample. The initial reference pose is unperturbed, so this tests observation-wise noise rather than accumulated drift. Only policy odometry is perturbed, not the simulator or evaluation state. Table 5 averages three evaluations of 924 All episodes per setting, including the no-noise retest.

A.4.3 INFERENCE-TIME FORESIGHT INTERVENTION

For Table 8, we set the foresight score $\beta_{t,a}$ to zero while retaining the trained checkpoint and all other computation, including future-feature prediction. Each setting is evaluated three times on 924 episodes under All cues without odometry noise, and navigation metrics are averaged across evaluations. For the action-ranking diagnostic, we compare logits before and after adding the foresight score in the same forward pass at states visited by the score-disabled policy. The top-ranked action differs in 18.55% of 363,164 decisions.

B DATASET DETAILS

B.1 DATASET SPLITS

The dataset contains 50,241 trajectories across 4,724 scenes. The training pool comprises 49,317 trajectories from 4,598 scenes: 48,238 trajectories from 4,499 scenes for Optimization and 1,079 trajectories from 99 scenes for Validation. Parameter updates use Optimization. Test contains 924 trajectories from 126 additional scenes, with no scene overlap across partitions.

B.2 SCENE STRUCTURE AND NAVIGATION STATISTICS

Multi-room environments. The 924 Test episodes span 126 scenes containing between **3 and 10 rooms** each. Figure 6(a) shows the episode distribution and scene counts for each room-count group.

Planned shortest-path length. Discrete-graph shortest paths range from **5.018 to 19.968 m**, with a mean of **9.590 m**; 37.45% are at least 10 m (Figure 6(b)). These lengths describe task scale, whereas navigation success uses NavMesh geodesic distance (Appendix D.6).

Trajectory distribution. Figure 7 summarizes scene coverage, room types, actions, targets, and instructions, counting each Test trajectory once.

B.3 EPISODE GENERATION AND GOAL CUES

For each episode, we sample a ProcTHOR house and target object, select a reachable goal region, and execute an expert route in AI2-THOR (Deitke et al., 2022; Kolve et al., 2017) (Figure 8). Each record pairs the overhead map and egocentric observations with actions, relative motion, a reference route, and three goal cues. Validation removes incomplete, non-executable, or misaligned episodes before scene-level splitting.

Point encodes the normalized executable endpoint; Trajectory uses up to four ordered map anchors, $g^r \in [0,1]^{4\times 2}$, with a validity mask; Language specifies the target through spatial relations without endpoint coordinates. Training samples uniformly from all seven non-empty cue subsets. Evaluation

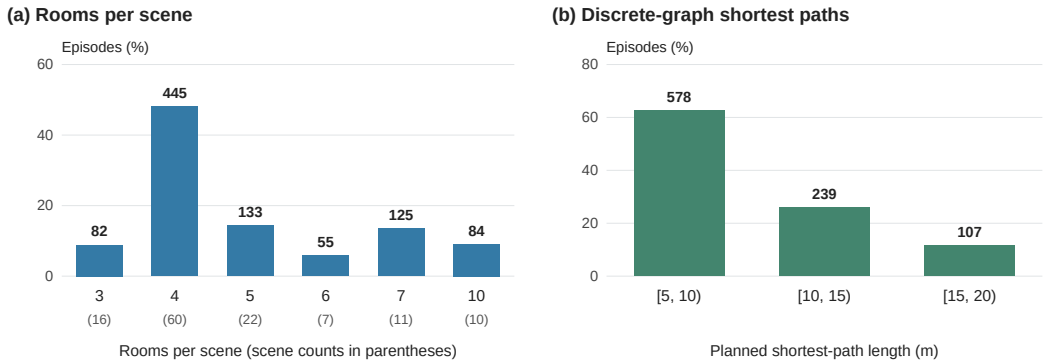


Figure 6: **Multi-room scenes and nontrivial route lengths.** The 924 Test episodes span scenes with at least three rooms (a) and discrete-graph shortest paths of 5–20 m (b). Bars show episode shares; labels give episode counts, with scene counts in parentheses.

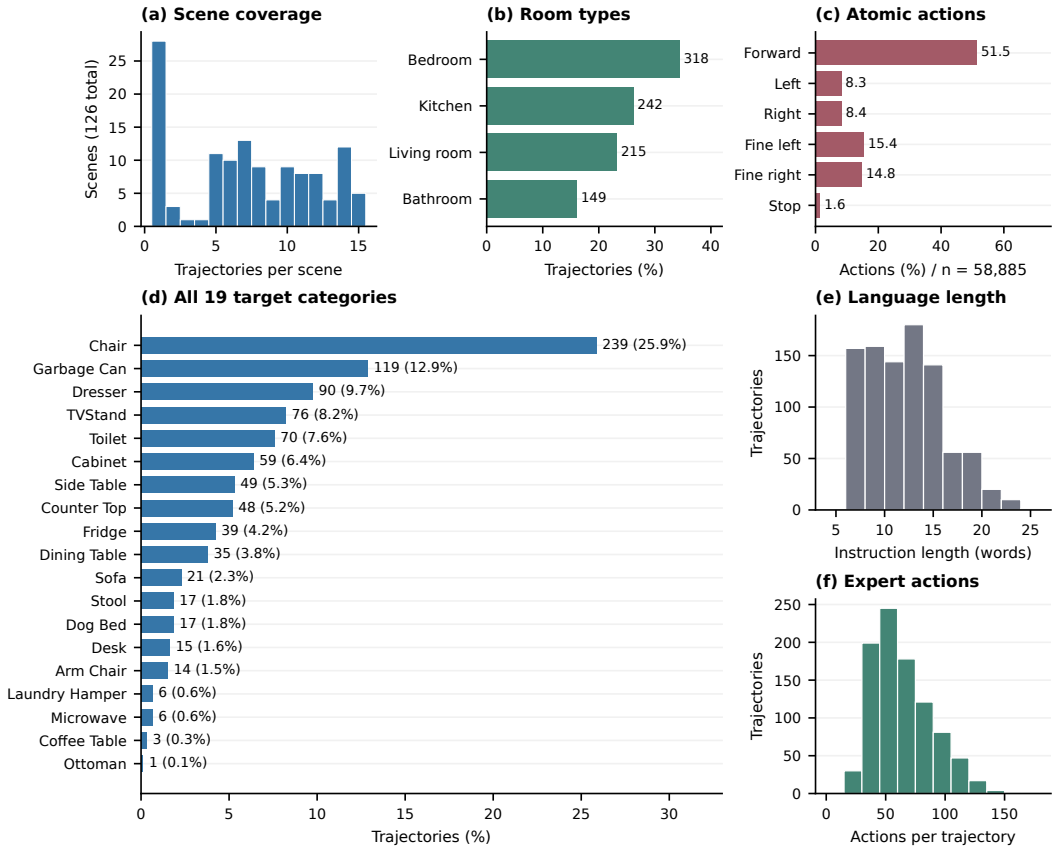


Figure 7: **Coverage and variation in the evaluation set.** The 924 trajectories vary in scene coverage (a), room type (b), action composition (c), target category (d), instruction length (e), and expert action count (f), complementing the scene and route-scale distributions in Figure 6.

uses Point, Trajectory, Language, and Combined; Appendix A.1 additionally reports all seven combinations.

B.4 Language Construction and Coverage

Language generation protocol. Instructions are generated by a deterministic constraint solver using a compositional grammar. The grammar combines `go to` with a target type and room type, optionally adding support, a coarse 3×3 room region, an object-relative anchor, or an extremal/rank descriptor. All Test grammar structures occur in Optimization.

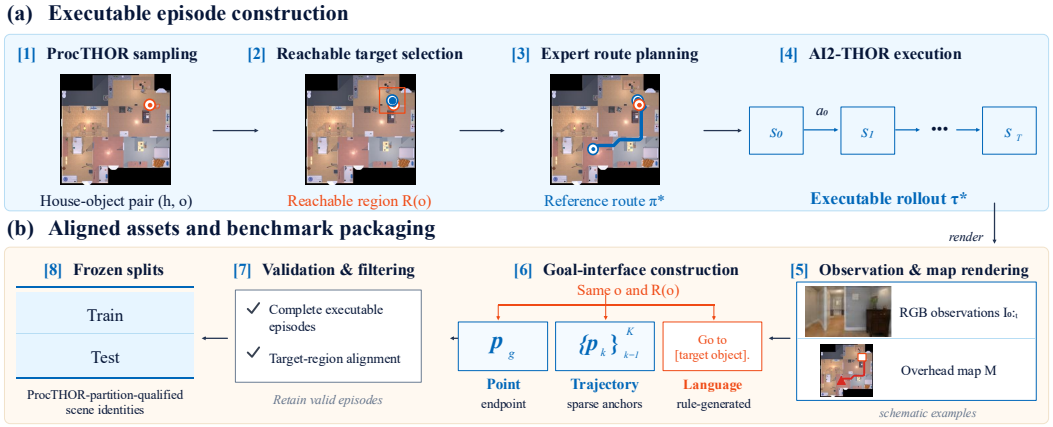


Figure 8: **Benchmark construction.** Expert routes pair overhead maps and observations with three goal cues; validation produces scene-disjoint splits.

Lexical coverage across splits. Optimization contains 5,776 distinct instructions spanning 20 target categories and four room types. Test covers 19 target categories and all four room types, with no unseen vocabulary; generalization is evaluated in unseen houses.

Target ambiguity. The mean number of other target-category instances is 2.31 per house in Optimization and 2.02 in Test, falling to 0.67 and 0.63 within the target room.

C Qualitative Results

C.1 Spatial Grounding Visualizations

C.1.1 Agent-to-Map and Goal-to-Map Grounding

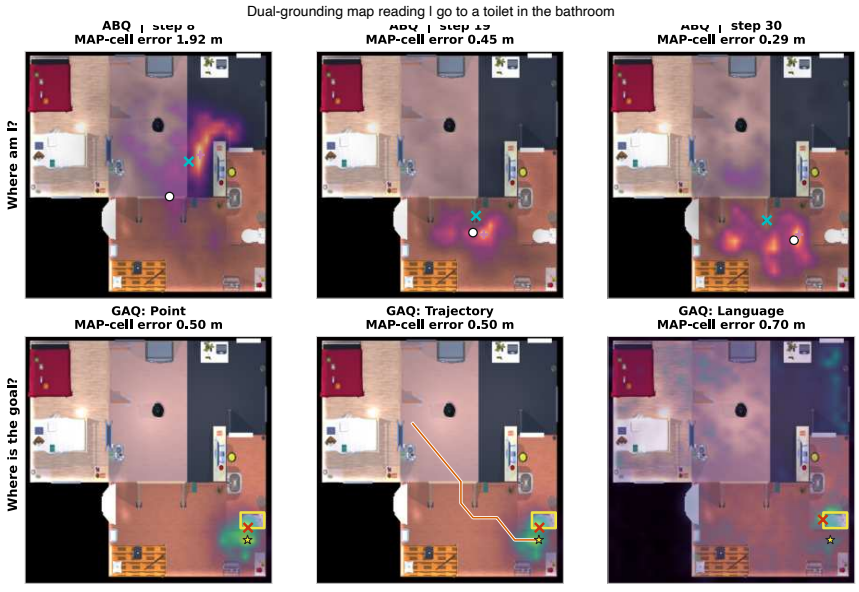


Figure 9: **Two spatial correspondences for map retrieval.** ABQ spatial belief marginals track agent motion (top); GAQ goal anchor fields localize the goal across cues (bottom).

C.1.2 LANGUAGE-GROUNDING FAILURE

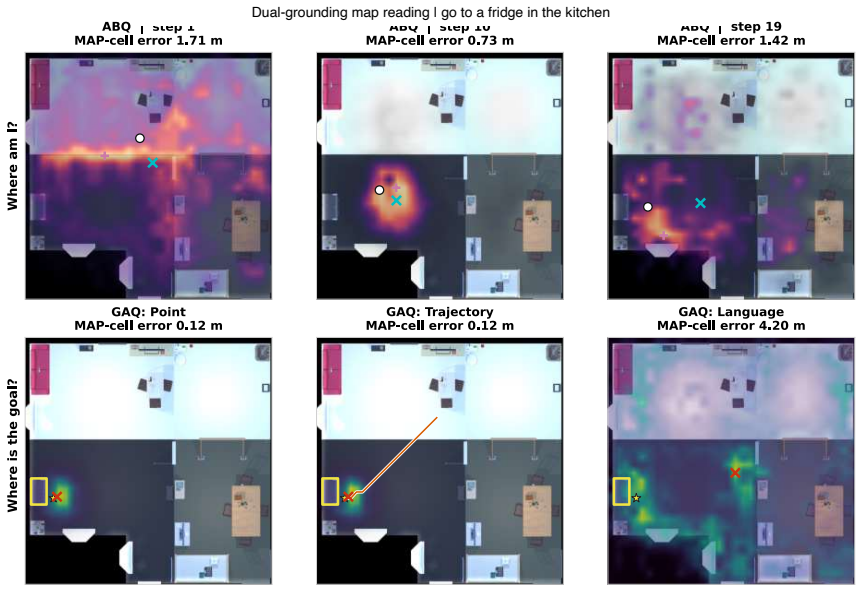


Figure 10: **Grounding failure.** Language mislocalizes the fridge goal, while Point and Trajectory localize it. ABQ retains multiple pose hypotheses.

C.2 GOAL-CUE ROLLOUT EXAMPLES

Figure 11 extends Figure 3 to three scenes under Point, Trajectory, Language, and Combined cues. Colored paths show policy rollouts; gray paths show reference routes. The displayed 1 m success labels also hold at 3 m. In scene h00041, the Point rollout stops 3.003 m from the goal (displayed as 3.00 m), just outside the 3 m success radius.



Figure 11: **Goal-cue effects on navigation.** Different cues yield distinct routes and stopping behavior toward the same target. (a) Toilet, h00041.



Figure 11: **Goal-cue effects on navigation (continued).** (b) TV stand, h00087.



Figure 11: **Goal-cue effects on navigation (continued).** (c) Coffee table, h00145.

C.3 Real-Robot Evaluation and Read-Weight Extraction

Deployment protocol. Using the platform in Section 5.5, inference runs on an NVIDIA GeForce RTX 4090 D server. Each method is adapted for six epochs on 45 trajectories annotated with Point, Trajectory, and Language goals. The final checkpoint is evaluated over three rounds totaling 60 trials across these conditions, with randomized initial positions and headings and a 1 m success radius. Success counts are summed over the three rounds.

Read-weight visualization. Figures 5 and 12 show deployment-test trajectories, distinct from adaptation data. We extract read weights offline using the deployment checkpoint, Language goals, and recorded RGB, actions, and odometry. ABQ weights are taken after spatial-belief gating and GAQ weights after goal-anchor biasing, then averaged over each branch's four queries. They are retrieval weights rather than the spatial distributions in Figure 9. Plots use a logarithmic scale from 10^{-5} to 10^{-1}; ROS localization and annotated goals provide reference markers. GAQ is shown once per trajectory because its map and goal inputs remain constant.

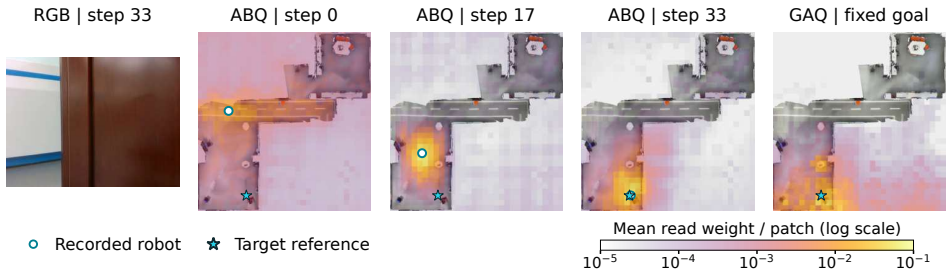


Figure 12: **Navigation to a table.** ABQ read weights track motion; GAQ read weights anchor the Language goal.

C.4 Real-Robot Visualizations

Figures 13–16 show four real-robot navigation executions.

Protocol note. Each recording shows four approximately evenly spaced instants, ordered left to right, with onboard views above and external views below. Views are aligned using content-based start offsets at original playback speeds.

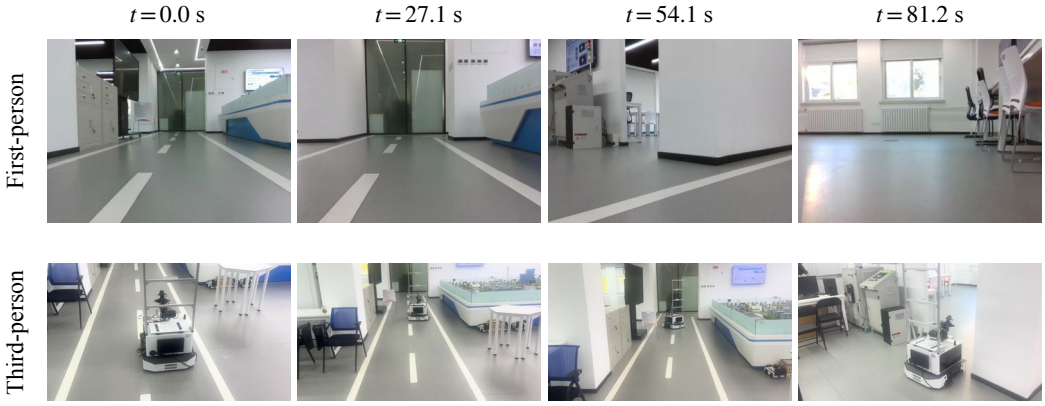


Figure 13: **Navigation to a doorway.** Language goal description: *"Go to the doorway of the room at the top right of the map."*

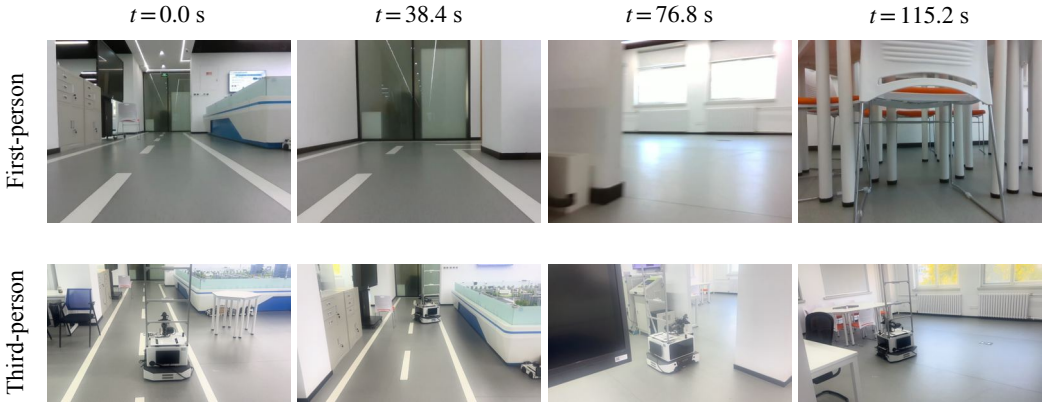


Figure 14: **Navigation to a white table.** Language goal description: *"Go to the white table in the room at the top right of the map."*

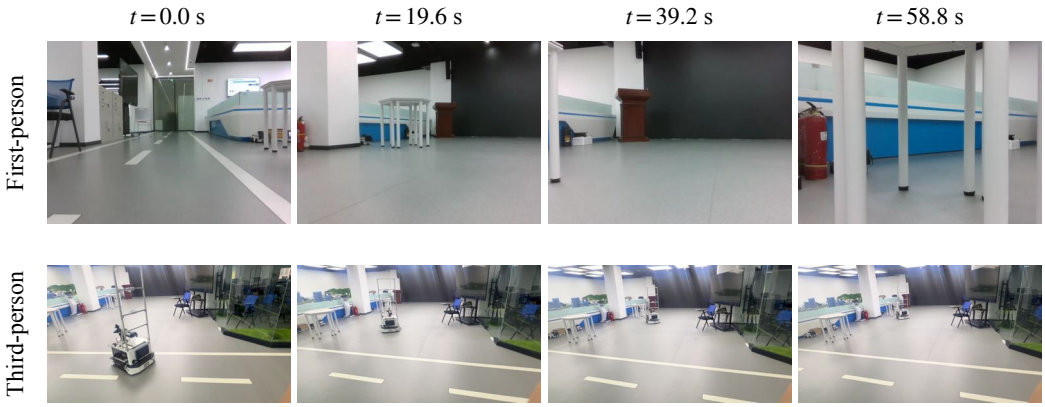


Figure 15: **Navigation to a centrally located table.** Language goal description: *"Go to the table in the center of the room at the bottom left of the map."*

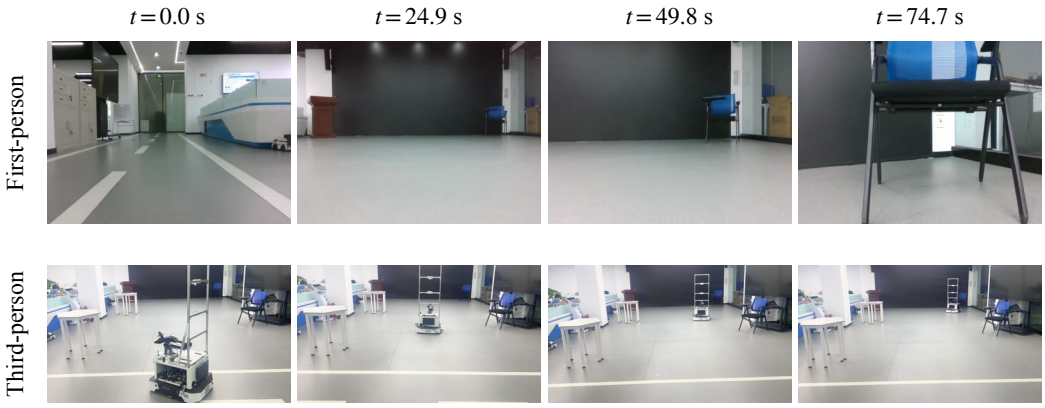
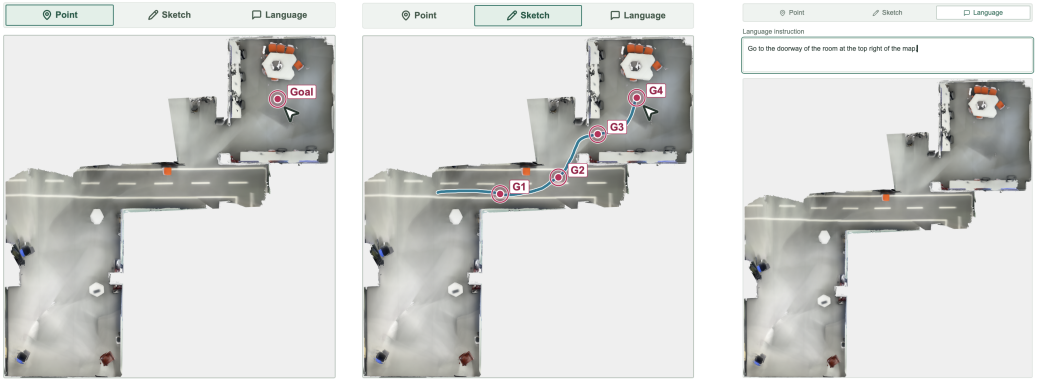


Figure 16: **Navigation to a stool in a room corner.** Language goal description: *"Go to the stool in the bottom-left corner of the room at the bottom left of the map."*



(a) Point (b) Trajectory (c) Language

Figure 17: **Map-centric goal specification for robot navigation.** (a) Point selects a destination on the map. (b) Trajectory uses a hand-drawn route with ordered anchors G1–G4 (the interface's Sketch mode). (c) Language specifies the destination relative to the map: *"Go to the doorway of the room at the top right of the map."*

D Implementation and Evaluation

Algorithm 1 summarizes the forward pass and training losses.

D.1 Observation and Goal Encodings

Pretrained representations. Frozen DINOv2-S/14 (Oquab et al., 2023) encodes observations and maps; frozen CLIP ViT-B/32 (Radford et al., 2021) provides language embeddings and future-feature targets.

Observation and action interface. The inputs are egocentric RGB history $O_{1:t} \in \mathbb{R}^{B\times T\times 3\times 98\times 98}$, overhead map $M \in \mathbb{R}^{B\times 3\times 448\times 448}$, start-relative odometry $U_{1:t} \in \mathbb{R}^{B\times T\times 4}$, previous actions, and the available goal cues. Here B is batch size and T is history length. The odometry vector is

$$u_t = \left[\mathrm{clip}\left(\frac{x_t - x_0}{20}, -1, 1\right), \mathrm{clip}\left(\frac{z_t - z_0}{20}, -1, 1\right), \sin(\theta_t - \theta_0), \cos(\theta_t - \theta_0)\right]. \tag{7}$$

Translation is relative to the episode start along map-aligned world axes, without rotation by $-\theta_0$; heading is relative to the initial heading. Absolute position and heading are not inputs. The six actions are Stop, a 0.2 m forward move, left/right rotations of 30°, and fine left/right rotations of 6°. The policy predicts five categorical actions, executes the first, and predicts again after observing the result.

Goal encoding. Point uses an MLP, and Trajectory uses learned anchor-order embeddings, a two-layer Transformer, and masked pooling. Each goal representation is projected to 384 dimensions, forming $G \in \mathbb{R}^{B\times 3\times 384}$ with a presence mask $m \in \{0,1\}^{B\times 3}$. Language uses CLIP's 512-dimensional text embedding with its standard 77-token context. A trainable linear projection and LayerNorm produce the 384-dimensional goal token.

D.2 Spatial Supervision and Belief Updates

Spatial supervision. Position and goal-anchor targets are normalized isotropic Gaussians on the 32×32 patch grid, centered at $(31u, 31v)$ for normalized coordinates (u, v), with standard deviation $\sigma = 1.5$ cells. The pose target multiplies this field by a normalized wrapped-angle Gaussian with $\sigma_\theta = 15^\circ$ over 12 heading bins spaced 30° apart. Goal-anchor supervision uses the same spatial width for every present goal cue.

Recurrent window and initialization. Each sample contains the episode's first frame, five preceding frames, and the current frame. ABQ excludes the global-first-frame slot and unrolls over the six contiguous slots, masking unavailable frames. The first valid posterior is initialized from visual scores; later steps apply action transport and gated visual updates. Gradients propagate through the window without detaching posteriors. Beliefs are reinitialized per training sample, and evaluation recomputes the window at each decision.

Belief transport. The deterministic prior follows the simulator action semantics. Forward shifts probability by 0.2 m along each heading using the metric map extent; a 30° turn rolls the 12-bin heading distribution by one bin; and a 6° fine turn mixes 0.8 of the current bin with 0.2 of the adjacent bin. Stop leaves the belief unchanged.

D.3 Policy and Semantic Foresight

Policy backbone. The backbone has four paired Visual and Action Blocks. Visual Blocks update observation tokens conditioned on odometry, previous actions, and goal/map context. Action Blocks mix five action queries and cross-attend to the updated visual tokens and context, comprising goal tokens and both groups of grounded map tokens.

Future-feature target. The next RGB observation is bicubically resized to 224×224, clamped to $[0, 1]$, and normalized with CLIP's channel statistics. Frozen CLIP ViT-B/32 produces a 7×7 patch grid; after removing CLS, the final LayerNorm and pretrained projection yield 512-dimensional patch features. Adaptive average pooling gives a detached 2×2 target, $Z^F_{t+1} \in \mathbb{R}^{B\times 2\times 2\times 512}$, supervising only the demonstrated first action a^*_t:

$$\mathcal{L}_{\text{future}} = \frac{1}{4}\sum_{s=1}^{4}\left[\lambda_{\cos}\left(1-\cos(\hat{z}_s, \text{sg}(z_s))\right) + \lambda_{\text{smooth}}\ell_{\text{smoothL1}}(\hat{z}_s, \text{sg}(z_s))\right]. \tag{8}$$

Here $\hat{z}_s = \hat{z}^F_{t+1,a^*_t,s}$ and $z_s = z^F_{t+1,s}$. All six branches receive action-loss gradients through β_t; the next observation is used only to construct the training target.

D.4 Forward Pass and Computational Cost

Computational cost. DGMR delivers eight map tokens to the policy. ABQ computes LHN heading–location scores for L history slots, $H = 12$ headings, and N map patches; GAQ and both four-query readers also access the full map. Retrieval therefore scales linearly with N at fixed feature dimension and history length. This excludes the DINOv2 encoder's global self-attention.

D.5 Inference Profiling Setup

Table 7 in the main text reports parameters, computation, and mean latency measured on one RTX 4090 D in FP32 with batch size one. The w/o foresight row disables both the future-feature predictor and semantic foresight scorer at inference, whereas Table 6 removes only the future-feature supervision during training.

Profiling setup. GPU-resident synthetic inputs comprise seven 98×98 RGB observations, one 448×448 map, odometry, and goals. GNM, ViNT, and NoMaD process 1,024 DINOv2 map

Algorithm 1 OmniMapNav forward pass and training objective.

Require: ego history $O_{1:t}$, map M, odometry $U_{1:t}$, previous actions $A_{1:t-1}$, present goals G, previous belief b_{t-1}

$Z^O, Z^M \leftarrow E_{\mathrm{DINO}}(O_{1:t}), E_{\mathrm{DINO}}(M)$
$E \leftarrow \mathrm{EgoEncode}(Z^O, \mathrm{StateEncode}(U_{1:t}, A_{1:t-1}))$; $e_t \leftarrow E[:, -1]$
$s_t \leftarrow \mathrm{CrossViewScore}(e_t, Z_t^O, Z^M)$
if first valid frame in the recurrent window **then**
    $b_t \leftarrow \mathrm{Softmax}(s_t/\tau)$
else
    $\bar{b}_t \leftarrow \mathrm{Transport}(b_{t-1}, A_{t-1}, s_M)$ ▷ previous-action transport
    $\eta_t \leftarrow \sigma(\mathrm{PriorGate}(e_t))$; $b_t \leftarrow \mathrm{Softmax}(\eta_t \log(\bar{b}_t + \epsilon) + s_t/\tau)$ ▷ gated visual correction
end if
$F^A \leftarrow \mathrm{GatedMapAttention}(Q_0^A + f_A(e_t, \hat{\theta}_t), Z^M, b_t)$
$G_{\mathrm{tok}}, m \leftarrow \mathrm{EncodeGoals}(G)$ ▷ three goal cues; any non-empty subset
$p^G \leftarrow \mathrm{SharedAnchorField}(G_{\mathrm{tok}}, m, Z^M)$
$F^G \leftarrow \mathrm{AnchorBiasedMapAttention}(Q_0^G + f_G(G_{\mathrm{tok}}, p^G), Z^M)$
$r_t, Z_t^V \leftarrow \mathrm{InterleavedPolicy}(Z_{1:t}^O, E, [G_{\mathrm{tok}}; F^A; F^G])$
$Y_t^{\mathrm{base}} \leftarrow \mathrm{ActionHead}(r_t)$
for $a \in \{1, \ldots, 6\}$ **do**
    $\hat{Z}_{t+1,a}^F \leftarrow P_\theta([Z_t^V; e_a])$ ▷ action-conditioned 2×2 CLIP grid
    $\beta_{t,a} \leftarrow \mathrm{Score}(W_q r_{t,1}, W_k \,\mathrm{Mean}(\hat{Z}_{t+1,a}^F))$
end for
$Y_t \leftarrow Y_t^{\mathrm{base}}$; $Y_{t,1,:} \leftarrow Y_{t,1,:}^{\mathrm{base}} + \beta_t$
if training **then**
    $Z_{t+1}^F \leftarrow \mathrm{sg}(E_{\mathrm{CLIP}}(O_{t+1}))$ ▷ detached 2×2 target only
    align only $\hat{Z}_{t+1,a_t^*}^F$ with Z_{t+1}^F in $\mathcal{L}_{\mathrm{future}}$
    compute $\mathcal{L}_{\mathrm{action}}, \mathcal{L}_{\mathrm{pose}}, \mathcal{L}_{\mathrm{anchor}}$, and optional $\mathcal{L}_{\mathrm{future}}$
    combine them using Eq. 6; never expose O_{t+1} or Z_{t+1}^F to policy inference
end if

Ensure: action logits Y_t and updated belief b_t

patches; FloNa retains its native map resizing to observation resolution. Timing includes visual/text encoding, map retrieval, and action prediction, excluding preprocessing, transfers, and environment execution. FLOP counts cover convolutions, matrix products, and attention, but not all elementwise, normalization, or softmax operations.

D.6 Training and Evaluation Protocol

Simulation evaluations use 924 episodes per round under each of Point, Trajectory, Language, and Combined. Scores average three rounds; Avg. denotes the equal-weight mean across these four goal conditions.

Baseline inputs and retained modules. All methods receive the episode's first RGB frame, five preceding frames, and the current frame, together with map-aligned odometry, a 448×448 overhead RGB map, and the available goal cues. GNM, ViNT, and NoMaD use the same DINOv2 map-token representation as OmniMapNav. FloNa instead retains its original EfficientNet map encoder and native localization and planning modules. All four baselines receive CLIP text features when Language is present. Trajectory is supplied to each baseline as the same ordered sequence of map coordinates, rather than a goal image.

Training details. All models and training ablations run for 10 epochs on eight GPUs with global batch size 6,144 and policy width 384. Per-GPU microbatch sizes (accumulation steps) are 96 (8) for GNM/ViNT, 256 (3) for NoMaD, 64 (12) for FloNa, and 384 (2) for OmniMapNav. AdamW uses learning rate 10^{-4}, weight decay 0.01, one warm-up epoch followed by cosine decay, mixed precision, and gradient clipping at 1.0. The action loss is geometrically weighted cross-entropy over five predicted actions. For OmniMapNav, $(\lambda_{\mathrm{act}}, \lambda_{\mathrm{pose}}, \lambda_{\mathrm{anchor}}, \lambda_{\mathrm{future}}) = (1, 0.05, 0.05, 0.05)$ and $(\lambda_{\cos}, \lambda_{\mathrm{smooth}}) = (1, 0.1)$. Reported simulation results use the final epoch-10 checkpoint.

Comparison controls. The future-feature ablation removes only its auxiliary loss, retaining the predictor and scorer. The spatially supervised dense reader uses a weight of 0.05 for each pose and goal-anchor loss.

Rollout and navigation metrics. Let $d_i^g(t)$ be the AI2-THOR NavMesh shortest-path distance from the pose after step t to episode i's executable target. A rollout ends on a predicted Stop or after 500 actions; reaching the action limit counts as failure. For $r \in \{0.5, 1, 2, 3\}$ m, GSR is

$$\mathrm{GSR}_i@r = \mathbf{1}\{\text{Stop predicted}\}\,\mathbf{1}\{d_i^g(T) < r\}. \tag{9}$$

Let $L_i = d_i^g(0)$ and P_i be the executed translational path length. We compute $\mathrm{SPL}@r = N^{-1}\sum_i \mathrm{GSR}_i@r\, L_i/\max(L_i, P_i)$ and $\mathrm{GOSR}_i@r = \mathbf{1}\{\min_{0\le t\le T} d_i^g(t) < r\}$. GOSR includes the initial pose and every pose after an atomic action, regardless of the final Stop decision. nDTW is $\exp[-\mathrm{DTW}(P,Q)/(3|Q|)]$ after 0.2 m arc-length resampling of executed/reference XZ paths, using Euclidean costs. sDTW multiplies nDTW by GSR@3m. Scores are dataset means expressed as percentages.