GROUNDING VISUAL EVIDENCE FOR LONG-RANGE LANGUAGE-CONDITIONED NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-range language-conditioned visual navigation (LCVN) requires embodied agents to follow natural-language instructions across environments spanning hundreds of meters. Topology-based LCVN relies on visual appearance for target retrieval, map construction, and navigation, but appearance becomes an unreliable proxy for task relevance at long range: similar landmarks confuse retrieval, visually redundant observations can be instruction-critical, and appearance variation causes waypoint errors to accumulate. We present **GRONav**, a unified grounded topology-based navigation framework that makes visual evidence decision-specific: retrieval is grounded in action and trajectory context, map compression in semantic composition, and navigation in geometric structure distilled from monocular depth. Experiments in CARLA and on a real quadrupedal robot demonstrate consistent gains in retrieval accuracy, navigation success, and robustness over strong baselines, while maintaining the lowest inference latency. GRONav achieves 91.7% target retrieval accuracy, reduces topology size by 68.0%, and improves real-world navigation performance by more than 25.0% on 60–200m tasks.

1 INTRODUCTION

Language-conditioned visual navigation (LCVN) requires embodied agents to understand natural-language instructions, ground them in visual observations, retrieve targets from a map, and execute image-goal-directed navigation in the physical world (Zhang & Ma, 2024; Dong et al., 2026; Jiang et al., 2025; Shah et al., 2023a). Unlike end-to-end vision-language navigation (VLN), which fuses language and vision within a single learned policy (Zhang et al., 2025b; Dong et al., 2026; Zhang et al., 2025a), LCVN explicitly relies on visual appearance as the common representation for language grounding, map construction, and navigation. This separation makes LCVN attractive for scalable and modular deployment, but also exposes a fundamental question: *what visual evidence should be retained to serve as a reliable relevance signal at each stage of long-range navigation?*

This question becomes critical as the navigation range grows, when agents must interpret free-form instructions and traverse environments spanning hundreds of meters, as required by service robots and autonomous delivery systems (Chen et al., 2025). At this scale, appearance alone is an unreliable proxy for task relevance, but *why* it fails differs across the LCVN pipeline. During *retrieval*, visually similar landmarks may occur at different locations, making isolated-view matching insufficient to identify the intended target (Figure 1A). During *mapping*, visually redundant observations may contain instruction-critical landmarks, making appearance-based compression discard information that matters for subsequent decisions (Figure 1C). During *navigation*, appearance variation can perturb waypoint prediction, and small local errors accumulate over long trajectories (Figure 1B).

These failures are also coupled: map compression determines which landmarks remain available for retrieval and which subgoals the policy must traverse; retrieval determines the route on which navigation operates; and errors introduced upstream cannot generally be recovered by local robustness downstream. Thus, the challenge is not simply to make appearance matching more robust, but to determine *which visual evidence is relevant to each decision*. This leads to our central principle: *visual appearance should be grounded according to the decision it supports, rather than treated as a universal similarity signal.*

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.

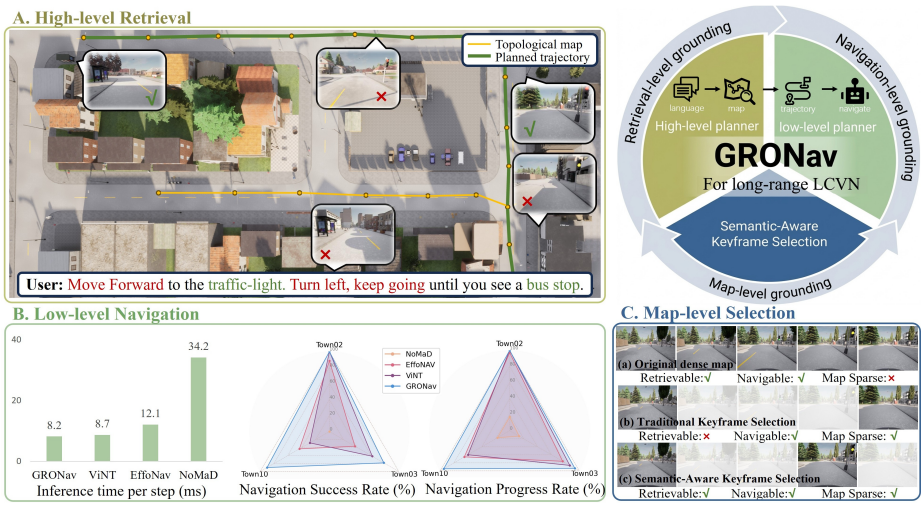


Figure 1: GRONav addresses long-range LCVN by grounding the task across three levels: (A) Action grounding disambiguates semantically similar targets by aligning instructions with candidate trajectories. (B) Spatially grounding navigation achieves higher success rates with lower inference latency than prior methods. (C) Semantic grounding guides keyframe selection to preserve nodes beneficial for both retrieval and navigation.

We instantiate this principle in **GRONav**, a unified **Gro**unded topology-based **Nav**igation framework that grounds visual evidence across the LCVN pipeline. GRONav introduces three complementary mechanisms. First, action-grounded retrieval decomposes instructions into semantic landmarks and action directives, and matches these directives against candidate trajectory signatures, allowing retrieval to distinguish visually similar targets using trajectory-level evidence rather than isolated views (Figure 1A). Second, semantically grounded map compression introduces Semantic-Grounded Keyframe Selection (SKS), which jointly considers semantic composition and visual change to preserve instruction-critical landmarks while substantially reducing topology size (Figure 1C). Third, spatially grounded navigation exposes spatial structure from visual observations through feature alignment and lightweight gated attention, enabling efficient waypoint prediction that is less sensitive to appearance variation (Figure 1B).

Together, these components turn visual appearance from a generic matching signal into a *decision-specific representation* shared across retrieval, mapping, and navigation. Unlike prior topology-based methods that focus on individual components (Shah et al., 2023c; Sridhar et al., 2023; Mei et al., 2025; Zeng et al., 2025), GRONav evaluates the complete pipeline across both simulation and real-world deployment. Experiments in CARLA and on a real-world quadrupedal robot demonstrate consistent improvements in retrieval accuracy, navigation performance, and efficiency. GRONav achieves 91.7% target retrieval accuracy while reducing map size by 68.0%, and improves real-world navigation performance over all baselines by more than 25.0% on 70–200 m tasks.

Our contributions are summarized as follows:

- We propose GRONav, a unified topology-based framework for long-range LCVN that grounds visual evidence according to the distinct decisions required for retrieval, mapping, and navigation.
- We introduce action-grounded retrieval and semantic-grounded keyframe selection, which exploit trajectory context and semantic composition to resolve ambiguous targets and preserve instruction-critical landmarks in compact topological maps.
- We develop a spatially grounded navigation policy that distills visual evidence from observations through feature alignment and gated attention, enabling efficient and robust long-horizon waypoint prediction under appearance variation.

2 Related Work

Topology-based language-conditioned visual navigation. LCVN requires an embodied agent to follow a natural-language instruction through an environment: the agent must understand the in-

struction, ground it in visual observations, retrieve the corresponding target from a map, and then execute image-goal navigation to reach it. Unlike end-to-end VLN, which fuses language and vision inside a single learned policy that predicts actions directly (Zhang et al., 2024; 2025b; Cheng et al., 2025; Zhang et al., 2025a; 2026; Team, 2025), LCVN separates the problem into an explicit map and a retrieval step, which keeps the pipeline more interpretable and more reliable in long-term navigation (Shah et al., 2026; Laina et al., 2025; Hirose et al., 2024). LM-Nav (Shah et al., 2023a) established this line by decomposing the instruction into an ordered list of landmarks with an LLM and matching each landmark to a graph node through CLIP similarity. Follow-up work explores generalization strategies at the levels of policy learning and graph construction (Hirose et al., 2024; Duan et al., 2026). Our work, by contrast, focuses on LCVN in long-scale tasks, where we find that visual appearance typically degrades along multiple perspectives; we therefore propose GRONav to address them systematically.

Visual appearance in long-range navigation. Visual appearance matters in topology-based LCVN, but its reliability degrades sharply as the map scale grows (Shah et al., 2023a). This degradation manifests at three levels. First, large scale gives rise to many semantically similar landmarks, which can confuse the retrieval process. Second, large scale also produces more graph nodes that must be compressed, whereas traditional compression methods rely only on visual difference and are therefore unsuited to LCVN tasks (Jha et al., 2025; Mur-Artal et al., 2015). Third, longer navigation process may amplify the accumulation of errors arising from visual information and easily leading to collapse (Guerrier et al., 2026). While prior work attempts to address some of these issues (Hirose et al., 2024; Duan et al., 2026; Busch et al., 2026), each solution targets a single aspect and lacks holistic awareness. GRONav therefore grounds visual evidence according to the decision each level supports: action directives for retrieval, semantic composition for map compression, and geometric structure for control.

3 METHOD

Framework Overview. GRONav consists of three key modules, each grounding a distinct signal at a different level of the navigation stack. The *high-level planner* (Section 3.1) interprets the instruction, decomposes it into semantic landmarks and action directives, and searches the topological graph to retrieve a sequence of subgoal nodes. The *low-level controller* (Section 3.2) sequentially navigates the robot between subgoal nodes using a lightweight policy model that aligns visual representations with monocular depth cues. Finally, *semantic-grounded keyframe selection* (Section 3.3) retains only visually distinctive or semantically informative frames, allowing the topological graph to remain compact while preserving nodes essential for retrieval and navigation.

Problem formulation. We represent the environment as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v \in \mathcal{V}$ stores an ego-centric RGB observation $\mathbf{o}_v$, and each edge $(u, v) \in \mathcal{E}$ indicates navigability between adjacent nodes. Given a natural language instruction $\mathcal{I}$, the high-level planner predicts a goal-node trajectory $\bar{\mathbf{v}} = (v_1, v_2, \ldots, v_n)$ over $\mathcal{G}$. The low-level controller then sequentially guides the robot toward each subgoal node v_i, using its associated observation $\mathbf{o}_{v_i}$ as a visual target.

3.1 ACTION-GROUNDED TARGET RETRIEVAL FOR HIGH-LEVEL PLANNER

Natural language instructions contain two complementary forms of information: semantic landmarks that specify *what* to look for and action directives that indicate *how* to reach them. Most existing LCVN methods primarily focus on semantic grounding while placing limited emphasis on action information. In contrast, GRONav explicitly grounds action directives onto graph trajectories.

We employ Qwen3 (Yang et al., 2025) to parse the instruction $\mathcal{I}$ into two aligned sequences $\bar{\mathbf{d}}$ and $\bar{\mathbf{a}}$ of length n:

$$\bar{\mathbf{d}} = (d_1, \ldots, d_n), \qquad \bar{\mathbf{a}} = (a_1, \ldots, a_n),$$

where d_i denotes the i-th semantic landmark description and a_i consists of $\{\texttt{move forward}, \texttt{turn left}, \texttt{turn right}, \texttt{stop}\}$ represents the action sequence preceding d_i. For instructions without explicit action directives, we set $a_i = \emptyset$.

For each candidate node $v \in \mathcal{V}$, we compute a grounded retrieval score that jointly incorporates semantic and action information. The semantic score is computed as the CLIP (Radford et al., 2021) cosine similarity between the node image $\mathbf{o}_v$ and the landmark description d_i. The action score

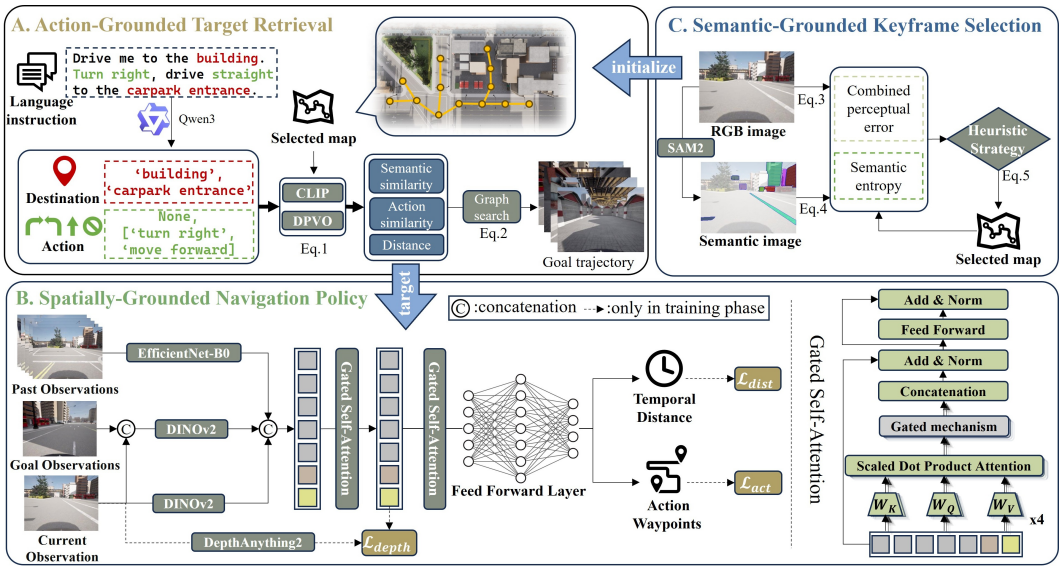


Figure 2: Overview of GRONav. **(A)** Action-grounded retrieval decomposes instructions into semantic and action components and performs graph-based trajectory search. **(B)** Semantic-grounded keyframe selection constructs a compact topological map by retaining frames with high visual change or semantic entropy. **(C)** The spatially grounded navigation policy sequentially executes navigation to retrieved goal nodes.

measures the consistency between the prescribed directive sequence a_i and the physical trajectory from the current robot position to node v. To obtain trajectory signatures, we apply DPVO (Teed et al., 2023) to image sequences along the graph path connecting the current node and candidate node v, producing a sequence of relative poses. Each inter-frame motion is classified according to its yaw variation using a fixed angular threshold, yielding a trajectory action sequence $\mathbf{s}(v)$. The retrieval score at phase i is defined as:

$$R(v,\, d_i,\, a_i) = \text{CLIP}(\mathbf{o}_v,\, d_i) + \frac{\text{LCS}\big(\mathbf{s}(v),\, a_i\big)}{\max\big(|\mathbf{s}(v)|,\, |a_i|\big)}, \tag{1}$$

where $\text{LCS}(\cdot,\cdot)$ denotes the Longest Common Subsequence length, which quantifies alignment between observed heading changes and prescribed action directives.

We recover the goal-node trajectory $\bar{\mathbf{v}} = (v_1, \ldots, v_n)$ via dynamic programming over $\mathcal{G}$. At phase i, we maintain a beam of k candidate nodes. Given the highest-scoring node from the previous phase $u^* = \arg\max_u Q^{(i-1)}(u)$, we evaluate all nodes reachable within a fixed radius and compute:

$$Q^{(i)}(v) = Q^{(i-1)}(u^*) + R(v,\, d_i,\, a_i) - \alpha \cdot D(v,\, u^*), \tag{2}$$

where $D(v, u^*)$ denotes graph distance and penalizes spatially inconsistent transitions, while α controls the regularization strength. We initialize $Q^{(0)}(\cdot) = 0$.

This formulation performs best-first search over a graph with instruction-conditioned edge weights (Dijkstra, 2022), yielding an interpretable retrieval process that explicitly reveals the landmark and action evidence underlying each node selection.

3.2 Spatially-Grounded Navigation Policy for Low-Level Controller

Following prior image-goal navigation approaches (Shah et al., 2023c; Shen et al., 2025; Sridhar et al., 2023), the low-level controller in GRONav receives the current observation $\mathbf{o}_t$, a history of $T = 5$ previous observations $\mathbf{o}_{t-T:t}$, and the retrieved goal image $\mathbf{o}_g$. The policy predicts both a temporal distance estimate $\hat{d}$ and a waypoint trajectory $\hat{\mathbf{w}} = (\mathbf{w}_t, \ldots, \mathbf{w}_{t+4})$ for the next five steps, where each waypoint $\mathbf{w}_i = (x_i, y_i, \cos\theta_i, \sin\theta_i)$ represents a predicted pose relative to the current robot frame.

We encode $\mathbf{o}_t$ and $\mathbf{o}_g$ using DINOv2 (Oquab et al., 2023), leveraging its strong semantic and spatial representations (Guerrier et al., 2026), while the observation history $\mathbf{o}_{t-T:t}$ is encoded using Effi-

cientNet (Tan & Le, 2019) for computational efficiency. The resulting token sequence is processed by a gated self-attention network followed by an MLP prediction head:

$$\hat{\mathbf{w}},\, \hat{d} = f_\theta\big(\text{Concat}(\mathbf{o}_{t-T:t},\ \mathbf{o}_t,\ \mathbf{o}_g)\big). \tag{3}$$

Visual encoders trained with appearance-centric objectives often exhibit diffuse attention patterns, commonly referred to as *attention sinks*, resulting in limited spatial discriminability (Kang et al., 2025). To improve geometric awareness, we introduce a spatial alignment objective that transfers geometric priors from a frozen depth estimator, DepthAnythingV2 (Yang et al., 2024), to the navigation policy.

Let $\mathbf{f}_{\text{depth}} \in \mathbb{R}^D$ denote the pooled depth feature extracted from the current observation, and let $\mathbf{f}_c \in \mathbb{R}^D$ denote the corresponding intermediate visual feature from the transformer decoder. We align their feature distributions by minimizing the KL divergence:

$$\mathcal{L}_{\text{depth}} = \mathbb{E}_b\left[\text{KL}\left(\text{softmax}\left(\mathbf{f}_{\text{depth}}^{(b)}\right)\ \middle\|\ \text{softmax}\left(\text{MLP}\left(\mathbf{f}_c^{(b)}\right)\right)\right)\right], \tag{4}$$

where b indexes samples within the training batch.

We further introduce a learned gating mechanism after each attention layer to suppress redundant attention heads (Qiu et al., 2025). Interestingly, when combined with spatial alignment, the mode achieves strong performance using only two transformer layers. Further analysis is provided in Section 4.3.

Training objective. Following Shah et al. (2023b), waypoint coordinates are normalized according to the maximum speed of each robotic platform. The overall training objective is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{dist}} + \lambda_1\, \mathcal{L}_{\text{act}} + \lambda_2\, \mathcal{L}_{\text{depth}}, \tag{5}$$

where $\mathcal{L}_{\text{dist}} = \text{MSE}(\hat{d},\, \tilde{d})$ and $\mathcal{L}_{\text{act}} = \text{MSE}(\hat{\mathbf{w}},\, \tilde{\mathbf{w}})$ denote the temporal-distance prediction loss and waypoint regression loss, respectively. $\tilde{d}$ and $\tilde{\mathbf{w}}$ represent the ground-truth temporal distance and waypoint trajectory, while λ_1 and λ_2 are balancing coefficients specified in Appendix C.1.

3.3 Semantic-Grounded Keyframe Selection for Map Compression

Topological graph construction for long-range LCVN presents a fundamental trade-off: graphs built from densely sampled observations become prohibitively large in terms of storage and retrieval cost, whereas aggressive subsampling or generic graph compression techniques often remove frames that are crucial for downstream node retrieval and navigation. To address this issue, we propose *Semantic-Grounded Keyframe Selection (SKS)*, an online strategy that grounds semantic richness into map compression by jointly considering visual variation and object-level semantic information.

Visual-change criterion. Scene variation is quantified using a combined perceptual metric (Jha et al., 2025):

$$e_{\text{vis}}(\mathbf{I}_{\text{c}},\, \mathbf{I}_{\text{l}}) = \beta\, e_{\text{photo}}(\mathbf{I}_{\text{c}},\, \mathbf{I}_{\text{l}}) + (1-\beta)\, e_{\text{ssim}}(\mathbf{I}_{\text{c}},\, \mathbf{I}_{\text{l}}), \tag{6}$$

where e_{photo} denotes the mean absolute pixel-intensity difference and $e_{\text{ssim}} = 1 - \text{SSIM}(\mathbf{I}_{\text{c}}, \mathbf{I}_{\text{l}})$ measures structural dissimilarity (Wang et al., 2004). $\mathbf{I}_{\text{c}}$ and $\mathbf{I}_{\text{l}}$ represent the current frame and the most recently retained keyframe, respectively, while β balances the contribution of the two terms. To account for scene-dependent variation, we employ an adaptive threshold based on the running mean μ_t and standard deviation σ_t of e_{vis} over a sliding window. A frame is selected under this criterion if:

$$e_{\text{vis}} \geq \mu_t + \theta_\Delta \sigma_t.$$

This adaptive formulation improves robustness across diverse environments without requiring manual threshold tuning.

Semantic-entropy criterion. Visual change alone is insufficient for LCVN, since successful navigation requires preserving frames that contain semantically informative landmarks for future retrieval. We therefore evaluate each frame according to its semantic entropy, which measures the information content of object-level scene composition. We employ SAM 2 (Ravi et al., 2024) to obtain pixel-wise segmentation with m detected object categories. The semantic entropy is defined as:

$$H_{\text{sem}}(\mathbf{I}_{\text{c}}) = -\frac{1}{m+1}\left[\sum_{j=1}^{m} p_j \log p_j + \left(1 - \textstyle\sum_{j=1}^{m} p_j\right) \log\left(1 - \textstyle\sum_{j=1}^{m} p_j\right)\right], \tag{7}$$

Table 1: Landmark retrieval accuracy (%) on dense topological maps across three scenes.

Method	CLIP	CLIP + Qwen3	GRONav
Town02	50.0	25.0	**100.0**
Town03	37.5	50.0	**87.5**
Town10	37.5	62.5	**87.5**
Average	41.7	45.8	**91.7**

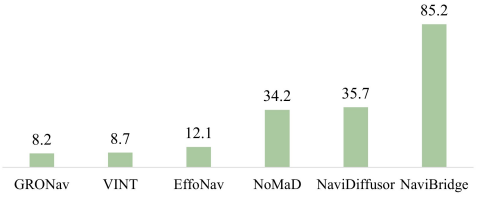


Figure 3: Inference time per step (ms)

where p_j denotes the proportion of pixels assigned to category j, and the second term accounts for background regions not associated with any object category. A high semantic entropy value H_{sem} indicates a scene containing diverse semantic content and object coverage, making it a strong candidate for a landmark node even under limited visual variation.

A frame $\mathbf{I}_c$ is retained as a keyframe if either criterion is satisfied:

$$\text{select}(\mathbf{I}_c) = \begin{cases} \text{True}, & e_{\text{vis}} \geq \mu_t + \theta_t \, \sigma_t, \\ \text{True}, & H_{\text{sem}}(\mathbf{I}_c) \geq \theta_s, \\ \text{False}, & \text{otherwise}, \end{cases} \tag{8}$$

subject to minimum and maximum inter-keyframe spacing constraints that prevent degenerate graph structures. SKS operates online during map construction with negligible computational overhead. The thresholds θ_t and θ_s determine the trade-off between graph compression and information retention, and their sensitivity is analyzed in Appendix A.

4 Experiments

4.1 Experimental Setups

Model Training: GRONav and all baselines are trained using four real-world navigation datasets - RECON (Shah et al., 2021), GoStanford (Hirose et al., 2019), SCAND (Karnan et al., 2022), and SACSoN (Hirose et al., 2023).

Simulation Experiments: We first evaluate retrieval and navigation independently in the CARLA simulator (Dosovitskiy et al., 2017), and then assess the complete system in real-world environments. Task lists are shown in Appendix B.2, we prioritize landmarks that appear multiple times in the topological graph as task destinations. We use three CARLA scenes with increasing complexity: Town02 (average trajectory length: 94.6 m), Town03 (111.3 m), and Town10 (174.4 m). Simulation experiments are conducted on an NVIDIA RTX5090. Following prior work (Shah et al., 2023c; Wang & Chen, 2025), we fine-tune GRONav and all baselines using 30 minutes of driving data collected in Town01 to alleviate failures caused by sharp turns.

Real-world Deployment: We additionally collect 30 minutes of onboard data to adapt the models to gait-induced camera motion from the quadruped platform. We then evaluate all methods in real-world scenarios. Real-world deployment runs on an NVIDIA RTX3080 laptop mounted onboard a Unitree GO2 quadruped equipped with an Intel RealSense D435i camera.

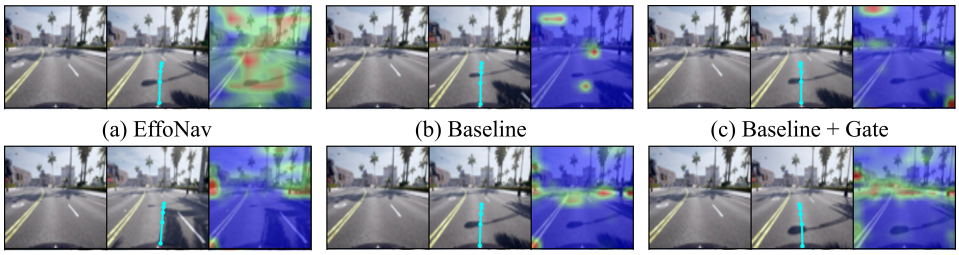
Evaluation Metric: We report Retrieval Accuracy (Acc) for goal retrieval, Success Rate (SR) and Progress Rate (PR) for navigation performance, and Compression Ratio (CR) for keyframe selection.

4.2 Target Retrieval Performance

We construct 24 tasks across the three CARLA scenes using densely sampled maps with node spacing of approximately 0.5 m. The tasks range from simple single-target instructions (e.g., *"Move forward to the gas station"*) to multi-step navigation commands with directional constraints (e.g., *"Move forward then turn left at the museum, and go to the park with the sculpture"*). We compare against two retrieval baselines. The first, **CLIP-only**, selects the graph node with the highest image–text similarity score. The second, **CLIP+Qwen3**, provides candidate node images together with the navigation instruction to Qwen3 and uses its confidence score for ranking. Both methods rely solely on semantic information and do not explicitly model action directives.

Table 2: Navigation performance (%) across CARLA scenes.

Method	Avg.	ViNT		EffoNAV		NoMaD		NaviBridge		NaviDiffusor		**GRONav**	
Metric	len (m)	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
Town02	94.6	100.0	100.0	88.3	97.1	0.0	17.2	0.0	19.8	0.0	22.3	**100.0**	**100.0**
Town03	111.3	61.7	86.4	36.7	76.2	0.0	13.6	0.0	13.0	0.0	13.6	**78.3**	**93.7**
Town10	174.4	28.3	59.9	43.3	65.2	0.0	17.5	0.0	29.2	0.0	18.1	**90.0**	**94.8**
Avg.	126.8	63.3	82.1	56.1	79.5	0.0	16.1	0.0	20.6	0.0	18.0	**89.4**	**96.2**



(a) EffoNav (b) Baseline (c) Baseline + Gate

(d) Baseline + Depth Align (e) GRONav (dual) (f) GRONav

Figure 4: Attention heatmaps for the navigation policy. Each column shows the target observation, the predicted waypoint trajectory, and the attention heatmap.

Table 1 reports retrieval accuracy across all scenes. GRONav achieves an average retrieval accuracy of 91.7%, substantially outperforming other baselines. The gains remain consistent across all environments, suggesting that grounding action directives onto candidate trajectories provides a strong disambiguation signal beyond semantic similarity alone. Interestingly, CLIP+Qwen3 occasionally performs worse than CLIP-only (e.g., Town02: 25.0% vs. 50.0%). This observation suggests that naively using a language model as a retrieval confidence estimator may introduce additional errors when its interpretation of the instruction becomes inconsistent with the agent's visual context.

4.3 Image-Goal Navigation Performance

We compare GRONav against two categories of image-goal navigation methods: (i) transformer-based approaches, including ViNT (Shah et al., 2023c) and EffoNAV (Shen et al., 2025), and (ii) diffusion-based methods, including NoMaD (Sridhar et al., 2023), NaviBridge (Ren et al., 2025), and NaviDiffusor (Zeng et al., 2025). Note that previous LCVN methods such as Lm-Nav (Shah et al., 2023a) and LeLaN (Hirose et al., 2024) using directly navigation methods belong to these two approaches (i.e., ViNT and NoMaD respectively).

We evaluate 12 long-range navigation tasks across Town02, Town03, and Town10, with trajectory lengths ranging from 50–300 m. Each task is repeated 15 times and averaged over trials. To isolate low-level control performance, all methods use the same goal nodes provided by GRONav's high-level planner. The policy predicts a sequence of five waypoints in the robot frame, each represented as $(x_i, y_i, \cos\theta_i, \sin\theta_i)$; following standard practice (Shen et al., 2025; Shah et al., 2023c), we execute the third waypoint and convert it into velocity commands, capped at 1.5 m/s and 0.8 rad/s.

Table 2 shows that GRONav achieves 89.4% success rate and 96.2% progress rate, outperforming the strongest baseline (ViNT) by 26.1% percentage points in SR. Diffusion-based methods fail entirely on these long-range tasks, which we attribute to their high inference latency (Figure 3). In contrast, GRONav runs at 8.2 ms per step, approximately $10\times$ faster than NaviBridge, making it more suitable for real-time deployment.

4.4 Ablation Study on Navigation

Component analysis. Table 3 isolates the contribution of each component. Both the spatial alignment objective and gated attention improve performance, with complementary gains when combined. The main improvement comes from spatial alignment: as shown in Figure 4, EffoNAV without alignment exhibits diffuse attention with weak spatial structure, whereas GRONav focuses on obstacles and navigable boundaries without explicit supervision. This suggests that depth alignment provides an implicit spatial grounding signal with improved interpretability.

Table 3: Ablation study on navigation policy components.

Method	Avg.	Baseline		+ Gate		+ Spatial Align		GRONav(dual)		GRONav	
Metric	len (m)	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
Town02	94.6	63.3	77.2	66.7	81.1	71.7	82.6	86.7	92.4	**100.0**	**100.0**
Town03	111.3	0.0	29.3	42.5	49.7	57.5	82.0	23.3	57.9	**78.3**	**93.7**
Town10	174.4	45.0	84.4	60.0	84.9	53.3	75.9	51.7	73.9	**90.0**	**94.8**
Avg.	126.8	36.1	63.6	56.4	71.9	60.8	80.2	53.9	74.7	**89.4**	**96.2**



(a) Baseline + Gate (b) GRONav (dual) (c) GRONav

Figure 5: Gate scores across the four attention layers of the navigation policy.

Figure 4 further contrasts attention patterns. EffoNAV spreads attention broadly, offering little discriminative signal. GRONav without alignment over-concentrates on isolated salient points, producing spurious saliency. With alignment, GRONav selectively attends to obstacles, curbs, and passable boundaries—structures directly relevant to navigation—confirming that alignment induces task-relevant and interpretable attention without region-level labels. The gating mechanism reduces model complexity. Prior methods (Shah et al., 2023c; Shen et al., 2025) typically use four-layer transformers, but with depth alignment, gate activations concentrate in the first two layers (Figure 5). Without alignment, gate patterns are less structured, suggesting that grounded geometry enables shallower, more efficient policies.

Dual-Token Spatial Alignment. In Section 3.2, spatial alignment is applied only to the current-frame token o_t. Here we evaluate GRONav(dual), which applies spatial alignment to both the current-frame token o_t and the history tokens $o_{t-T:t}$. As shown in Table 3, extending alignment to all input tokens yields only 53.9% average SR, substantially worse than current-frame-only alignment (89.4%). This issue relates to the gate score (as demonstrated in Figure 5) and shown as single-head collapse, where one gate value is significantly larger than the others. The reason is that aligning EfficientNet-encoded history tokens to depth features produced by a different encoder introduces new inconsistencies that the transformer cannot resolve, directly motivating GRONav's selective-alignment design.

4.5 Topological Map Compression Evaluation

Retrieval performance under map compression. To ensure a fair comparison, we compare SKS against three keyframe-selection baselines at a matched compression ratio of approximately 67% (retaining ∼33% of graph nodes): (i) uniform temporal sampling, (ii) ORB-based keyframe selection from ORB-SLAM (Mur-Artal et al., 2015; Mur-Artal & Tardós, 2017), and (iii) Adaptive Keyframe Selection (AKS) (Jha et al., 2025), which relies solely on visual variation. Table 4 shows that SKS achieves the highest average retrieval accuracy (83.3%) while maintaining a comparable compression ratio. Relative to the dense-map setting in Table 1, only 2 of 24 targets are lost despite a 67% reduction in map size. These results suggest that visual-change criteria alone are insufficient

Table 4: Retrieval Acc (%) under different keyframe selection methods at matched compression rate (%).

Method	**AKS**		**ORB**		**Uniform**		**SKS (ours)**	
Metric	CR	Acc	CR	Acc	CR	Acc	CR	Acc
Town02	69.6	62.5	65.9	62.5	65.9	**87.5**	69.9	**87.5**
Town03	67.5	75.0	67.7	50.0	66.0	62.5	65.4	**87.5**
Town10	66.7	**87.5**	65.6	75.0	65.8	75.0	68.7	75.0
Avg.	67.9	75.0	66.4	62.5	65.9	75.0	68.0	**83.3**

Table 5: Performance (SR, %) on maps compressed by different keyframe selection methods.

Scene	AKS			ORB			Uniform			**SKS (ours)**		
	ViNT	EffoNAV	GRONav	ViNT	EffoNAV	GRONav	ViNT	EffoNAV	GRONav	ViNT	EffoNAV	GRONav
Town02	91.7	50.0	75.0	95.0	83.3	75.0	93.3	76.7	55.0	95.0	83.3	85.0
Town03	65.0	46.7	73.3	68.3	33.3	77.7	60.0	40.0	73.3	58.3	50.0	81.7
Town10	21.7	16.7	53.3	20.0	50.0	53.3	25.0	33.3	48.3	28.3	36.7	73.3
Avg.	59.5	37.8	67.2	61.1	55.5	68.7	59.4	50.0	58.9	60.5	56.7	80.0



Figure 6: Left: qualitative results of different methods in the real-world experiment. Right: challenges in real-world environment, each column shows the subgoal and the predicted waypoint.

for LCVN and highlight the importance of semantic information for preserving retrieval-critical landmarks under aggressive map compression.

Navigation performance under map compression. Table 5 evaluates navigation performance on compressed maps produced by different keyframe selection strategies across three navigation models. SKS achieves the highest SR in most of cases, indicating that it preserves nodes critical for both retrieval and downstream navigation. Compared with the dense setting in Table 2, the degradation is also moderate in GRONav. Importantly, SKS is not coupled to GRONav: all evaluated models, including GRONav, ViNT, and EffoNAV. This demonstrates that SKS is a general topology compression strategy that improves a wide range of navigation systems.

4.6 Real-World Deployment on a Quadrupedal Robot

To evaluate real-world performance, we design 12 long-range navigation tasks across indoor and outdoor environments and compare ViNT, EffoNAV, and GRONav. Trajectory lengths range from 60–200 m, and each task is repeated three times. Figure 6 shows a representative trajectory. Compared with simulation, real-world deployment introduces additional challenges, including illumination variation, uneven terrain, and obstacles including flower terraces and pedestrians. Table 6 shows that GRONav achieves an average SR of 55.6%, substantially outperforming ViNT and EffoNAV. Notably, GRONav attains a considerably higher PR than SR, suggesting that the robot completes most of the prescribed route before failure. This indicates that failures are primarily "last-mile" errors near the goal, rather than the early trajectory divergence frequently observed in ViNT and EffoNAV.

Table 6: Navigation performance (%) in real-world environment.

Method	Avg.	ViNT		EffoNAV		**GRONav**	
Metric	len (m)	SR	PR	SR	PR	SR	PR
Indoor	98.2	33.3	61.5	5.6	33.9	**66.7**	**82.6**
Outdoor	132.4	27.8	61.8	0.0	20.6	**44.4**	**74.2**
Avg.	115.3	30.6	61.6	2.8	27.2	**55.6**	**78.4**

5 Conclusion

We propose GRONav, a topological framework for long-range LCVN that unifies the navigation stack under the grounding principle. By grounding action in retrieval, semantics in map construction, and geometry in control, GRONav provides an interpretable pipeline and consistently improves both target retrieval accuracy and overall navigation success over strong baselines. Looking ahead, we aim to extend GRONav to scalable urban environments and diverse navigation agents, further demonstrating the generalizability of the tri-grounded design in real-world embodied systems.

AI USE STATEMENT

In this work, we have not used generative AI tools for design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. Generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses are not applicable to this work. Additionally, we used generative AI tools for edit a research paper to improve readability and propose a title or keywords for a research paper. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. ETPNav: Evolving topological planning for vision-language navigation in continuous environments. *arXiv preprint arXiv:2304.03047*, 2024.

Finn Lukas Busch, Matti Vahs, Quantao Yang, Jesús Gerardo Ortega Peimbert, Yixi Cai, Jana Tumova, and Olov Andersson. Learning to localize reference trajectories in image-space for visual navigation, 2026. URL https://arxiv.org/abs/2602.18803.

Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee K. 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*, pp. 9796–9810, 2024.

Sheng Chen, Peiyu He, Jiaxin Hu, Ziyang Liu, Yansheng Wang, Tao Xu, Chi Zhang, Chongchong Zhang, Chao An, Shiyu Cai, et al. Astra: Toward general-purpose mobile robots via hierarchical multimodal learning. *arXiv preprint arXiv:2506.06205*, 2025.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. In *RSS*, 2025.

Edsger W Dijkstra. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: his life, work, and legacy*, pp. 287–290. 2022.

Yifei Dong, Fengyi Wu, Yilong Dai, Lingdong Kong, Guangyu Chen, Xu Zhu, Qiyu Hu, Tianyu Wang, Johnalbert Garnica, Feng Liu, et al. Language-conditioned world modeling for visual navigation. *arXiv preprint arXiv:2603.26741*, 2026.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

Hongbo Duan, Shangyi Luo, Zhiyuan Deng, Yanbo Chen, Yuanhao Chiang, Yi Liu, Fangming Liu, and Xueqian Wang. Causalnav: A long-term embodied navigation system for autonomous mobile robots in dynamic outdoor scenarios. *IEEE Robotics and Automation Letters*, 2026.

Maeva Guerrier, Karthik Soma, Jana Pavlasek, and Giovanni Beltrame. Can vision foundation models navigate? zero-shot real-world evaluation and lessons learned, 2026. URL https://arxiv.org/abs/2603.25937.

Noriaki Hirose, Fei Xia, Roberto Martín-Martín, Amir Sadeghian, and Silvio Savarese. Deep visual mpc-policy learning for navigation. *IEEE Robotics and Automation Letters*, 4(4):3184–3191, 2019.

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

Noriaki Hirose, Catherine Glossop, Ajay Sridhar, Dhruv Shah, Oier Mees, and Sergey Levine. Lelan: Learning a language-conditioned navigation policy from in-the-wild videos. *arXiv preprint arXiv:2410.03603*, 2024.

Raman Jha, Yang Zhou, and Giuseppe Loianno. Adaptive keyframe selection for scalable 3d scene reconstruction in dynamic environments. *arXiv preprint arXiv:2510.23928*, 2025.

Kaixuan Jiang, Yang Liu, Weixing Chen, Jingzhou Luo, Ziliang Chen, Ling Pan, Guanbin Li, and Liang Lin. Beyond the destination: A novel benchmark for exploration-aware embodied question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9091–9101, 2025.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. *arXiv preprint arXiv:2503.03321*, 2025.

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

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.

Sebastián Barbas Laina, Simon Boche, Sotiris Papatheodorou, Simon Schaefer, Jaehyung Jung, Helen Oleynikova, and Stefan Leutenegger. Findanything: Open-vocabulary and object-centric mapping for robot exploration in any environment. *arXiv preprint arXiv:2504.08603*, 2025.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Yanghong Mei, Yirong Yang, Longteng Guo, Qunbo Wang, Ming-Ming Yu, Xingjian He, Wenjun Wu, and Jing Liu. Urbannav: Learning language-guided urban navigation from web-scale human trajectories. *arXiv preprint arXiv:2512.09607*, 2025.

Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

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.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *arXiv preprint arXiv:2505.06708*, 2025.

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.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

Hao Ren, Yiming Zeng, Zetong Bi, Zhaoliang Wan, Junlong Huang, and Hui Cheng. Prior does matter: Visual navigation via denoising diffusion bridge models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12100–12110, 2025.

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

Dhruv Shah, Błażej Osiński, Sergey Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on robot learning*, pp. 492–504. pmlr, 2023a.

Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. GNM: A General Navigation Model to Drive Any Robot. In *International Conference on Robotics and Automation (ICRA)*, 2023b. URL https://arxiv.org/abs/2210.03370.

Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. ViNT: A foundation model for visual navigation. In *7th Annual Conference on Robot Learning*, 2023c. URL https://arxiv.org/abs/2306.14846.

Hardik Shah, Erica Tevere, Deegan Atha, Marcel Kaufmann, Shehryar Khattak, Manthan Patel, Marco Hutter, Jonas Frey, and Patrick Spieler. Wildos: Open-vocabulary object search in the wild. *arXiv preprint arXiv:2602.19308*, 2026.

Wangtian Shen, Pengfei Gu, Haijian Qin, and Ziyang Meng. Effonav: An effective foundation-model-based visual navigation approach in challenging environment. *IEEE Robotics and Automation Letters*, 2025.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL https://arxiv.org/abs/2508.10104.

Xinshuai Song, Weixing Chen, Yang Liu, Weikai Chen, Guanbin Li, and Liang Lin. Towards long-horizon vision-language navigation: Platform, benchmark and method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12078–12088, 2025.

Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration. *arXiv pre-print*, 2023. URL https://arxiv.org/abs/2310.xxxx.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

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

Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *Advances in Neural Information Processing Systems*, 2023.

Jiahui Wang and Changhao Chen. Dynanav: Dynamic feature and layer selection for efficient visual navigation. *arXiv preprint arXiv:2509.21930*, 2025.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024.

Yiming Zeng, Hao Ren, Shuhang Wang, Junlong Huang, and Hui Cheng. Navidiffusor: Cost-guided diffusion model for visual navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11994–12001, 2025.

Hai Zhang, Siqi Liang, Li Chen, Yuxian Li, Yukuan Xu, Yichao Zhong, Fu Zhang, and Hongyang Li. Sparse video generation propels real-world beyond-the-view vision-language navigation. *arXiv preprint arXiv:2602.05827*, 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. *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*, 2025a.

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*, 2025b.

Junbo Zhang and Kaisheng Ma. Mg-vln: Benchmarking multi-goal and long-horizon vision-language navigation with language enhanced memory map. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7750–7757. IEEE, 2024.

A Implementation Details

High-level Retrieval. The initial candidate pool size is $k = 20$ nodes per phase, selected by CLIP cosine similarity. The beam width for the dynamic programming search (Equation 2) is 5 nodes per phase. The spatial-jump penalty coefficient is $\alpha = 0.1$. Heading-change classification uses a yaw threshold of $15°$: changes below this are labeled `Move forward`, left turns above this threshold are labeled `Turn left`, and right turns above this threshold are labeled `Turn right`.

Low-level Navigation. History length is $T = 5$ observations. The DINOv2 encoder uses the ViT-B/14 variant; EfficientNet uses the B0 configuration. Feature dimension is $D = 1024$ after linear projection. Loss coefficients are $\lambda_1 = 0.01$ and $\lambda_2 = 0.1$. Training uses AdamW with learning rate 10^{-4}, batch size 256, for 30 epochs.

Semantic-Aware Keyframe Selection. The perceptual error weight $\beta = 0.7$ in Equation 6 follows Jha et al. (2025). The adaptive threshold uses a sliding window of size $W = 30$ frames. The deviation multiplier $\theta_t = 0.75$ and the semantic entropy threshold $\theta_s = 0.3$ are selected by grid search on a held-out map segment from Town01. To prevent degenerate graphs, we enforce a minimum inter-keyframe interval of 2 frames and a maximum interval of 10 frames; these constraints are applied after both selection criteria. Sensitivity analysis over $\theta_s \in [0.1, 0.5]$ shows that retrieval accuracy varies by at most 4% across this range, confirming robustness to moderate threshold variation.

B Experiment Setting

B.1 Data Collection

The training data of GRONav is collected from a remote-controlled car, whose egocentric view differs noticeably from both the CARLA vehicle camera and the onboard view of a quadruped robot. Validation in CARLA and in real-world scenarios therefore requires a short fine-tuning stage to ensure reliable long-range LCVN.

In the simulator, we collect roughly 30 min of driving data in Town01 for fine-tuning and evaluate on Town02, Town03, and Town10. As shown in Figure 8, the fine-tuning and evaluation scenes differ significantly in both weather conditions and object composition. For real-world validation, we collect an onboard dataset using a Unitree GO2 quadruped equipped with a RealSense D435i and an RTK module (Figure 7). Figure 9 highlights the area used for fine-tuning (green) and the area used for validation (yellow); the fine-tuning data contains only outdoor scenes, while validation tasks cover both outdoor and indoor scenes. The real-world fine-tuning set also lasts about 30 min.

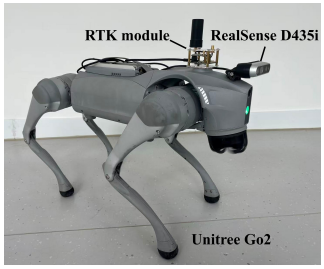


Figure 7: Quadruped robot setup used for real-world validation.

We report **Success Rate (SR)**: the fraction of tasks where the agent reaches the goal and stop within a distance threshold (5 m in this paper); **Progress Rate (PR)**: the fraction of the prescribed trajectory completed before failure; **Retrieval Accuracy (Acc)**: the fraction of goal nodes correctly identified from the topological map; and **Compression Ratio (CR)**: the fraction of original map nodes pruned after keyframe selection.

B.2 Task Lists

In this section we list all LCVN tasks used in our experiments: the retrieval task set (Table 7), the CARLA navigation task set (Table 8), and the real-world navigation task set (Table 9). When selecting tasks, priority is given to landmarks that appear multiple times in the topological graph.



Figure 8: CARLA data used in our experiments. Fine-tuning data is collected in Town01, while evaluation is performed on Town02, Town03, and Town10, which differ in weather and scene composition.

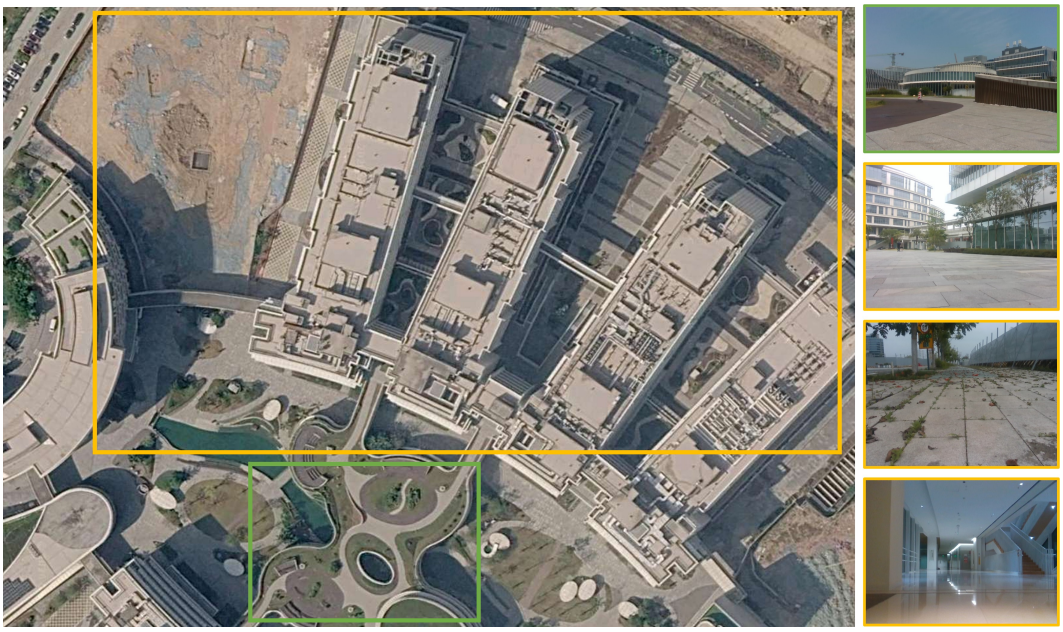


Figure 9: Real-world data collection. Green regions denote the area used for onboard fine-tuning; yellow regions denote the indoor and outdoor areas used for validation.

Table 7: Full task list for retrieval evaluation.

Task ID	Scene	Language Instruction
1	Town02	Move forward to the gas station.
2		Move forward then turn left, go straight until you see a bus station.
3		Move forward to the traffic light. Then turn left, keep going until you see a bus station.
4		Move forward then turn left, guide me to the church.
5		Move forward then turn left, keep going until you see a red vending machine on your right side.
6		Take me to the entrance of the park.
7		Find the blue mailbox in the housing estate.
8		Guide me to the nearest traffic light.
9	Town03	Move forward then turn left, keep going then drive into the park with a left turn.
10		Move forward then turn right, go to the fountain in the building on your left hand side.
11		Drive me to the gas station.
12		Turn right and send me to the red housing estates.
13		Move forward then turn right, keep going until you see red blocks.
14		Drive to the building. Turn right, drive straight to the car park entrance.
15		Move forward then turn right, walk along the road until you see a bus station.
16		Move forward then turn left, keep going then drive into the park with a right turn.
17	Town10	Move forward and guide me to the park.
18		I want to go to the museum.
19		Walk along the road then turn left, take me to the hotel.
20		Move forward then turn left, you will find a car park entrance on your right side.
21		Move forward then turn left, keep going until you see a bus station.
22		Move forward then turn left at the museum, go to the park with sculpture.
23		Move forward then turn right, after that turn left and stop in front of the crossroad.
24		Take me to the nearest bus stop.

Table 8: Navigation task list in CARLA simulator. Each task is characterized by its scene, trajectory length, and language instruction.

Task ID	Scene	Length (m)	Language Instruction
1	Town02	54	Move forward to the gas station.
2		147	Take me to the church.
3		95	Move forward then turn left, go straight until you see a bus station. Keep moving then turn left, go straight until you see a bus station.
4		47	Find the blue mailbox in the housing estate.
5	Town03	101	Drive me to the park.
6		107	Move forward then turn right, go to the fountain in the building on your left hand side. Then drive me to the gas station.
7		133	Take me to the red stalls.
8		105	Move forward then turn left, you will see red stalls. Turn left then turn right, keep going to the car park entrance.
9	Town10	111	I want to go to the museum.
10		141	Guide me to the car park entrance.
11		152	Walk along the road then turn left, take me to the hotel.
12		294	Walk along the road then turn left to the sculpture. After that drive me to the park.

Table 9: Navigation task list in real-world validation. Each task is characterized by its scene, trajectory length, and language instruction.

Task ID	Scene	Length (m)	Language Instruction
1	Indoor	126	Move forward then turn right, keep going until you see a refrigerator on your left side.
2		75	Move forward then turn left, stop when you see a fire extinguisher.
3		102	Go straight ahead and turn left; you'll see a printer. Keep going until you see the second vending-machine.
4		63	Move forward then turn left, stop when you see a plant.
5		106	Move forward then turn right, stop at the stairs.
6		138	Move forward then turn right, keep going until you see a plant.
7	Outdoor	120	move forward then turn right, stop when you see the traffic sign.
8		197	Guide me to the sculpture. Then turn left, keep going until you see an escalator.
9		204	Move forward to the sculpture, turn right then walk straight to the fire extinguisher.
10		71	Move forward to the fire extinguisher. Keep going until you see another fire extinguisher.
11		112	Walk along the road to the bench. Keep going then turn left, turn left again, stop at the stair.
12		91	Move forward the traffic sign, then guide me to stairs.

Table 10: Navigation performance (%) under different values of the depth-alignment coefficient λ_2. SR is reported over 60 trials (4 tasks $\times$ 15 runs) per scene; λ_1 is fixed at 0.01.

Method	Avg.	$\lambda_2 = 0.01$		$\lambda_2 = 0.05$		$\lambda_2 = 0.1$		$\lambda_2 = 0.5$		$\lambda_2 = 1$	
Metric	len (m)	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
Town02	94.6	95.0	97.2	100.0	100.0	100.0	100.0	91.7	96.0	80.0	90.5
Town03	111.3	71.7	86.3	73.3	90.1	78.3	93.7	70.0	87.5	46.7	74.2
Town10	174.4	81.7	88.5	85.0	91.4	90.0	94.8	78.3	89.2	76.7	83.2
Avg.	126.8	82.8	90.7	86.1	93.8	89.4	96.2	80.0	90.9	68.1	82.6

C Additional Experimental Results

C.1 Ablation on the depth-alignment coefficient

In Equation 5, the two hyper-parameters λ_1 and λ_2 balance the waypoint loss and the depth-alignment loss against the distance loss. We follow prior work (Shah et al., 2023c; Sridhar et al., 2023; Wang & Chen, 2025; Shen et al., 2025) and fix $\lambda_1 = 0.01$; the sensitivity of λ_2, which controls the strength of spatial grounding, is reported in Table 10. Moderate values of $\lambda_2 \in [0.05, 0.5]$ yield comparable performance (within 10 SR points of the optimum), indicating that GRONav is robust to the choice of depth-alignment strength. When λ_2 is too small (λ_2=0.01) the depth signal is too weak to anchor visual tokens and SR drops to 69.4%; when λ_2 is too large (λ_2=1) the depth loss dominates the main distance and waypoint objectives, degrading SR to 61.7%. The best result is obtained at λ_2=0.1, which we adopt as the default.

In the main experiments and visualizations, we evaluate GRONav using DepthAnything2 Yang et al. (2024) and DINOv2 Oquab et al. (2023) to ensure a fair comparison, as baselines such as EffoNav also adopt the same encoder. Here, we additionally provide results for GRONav trained with DepthAnything3 Lin et al. (2025) and DINOv3 Siméoni et al. (2025), as shown in Table 11. The original model in the main text is renamed GRONav-v2, while the newer one is denoted GRONav-v3. The experimental setting is identical to that in Table 2. As shown in the table, after replacing the encoder block, the average performance improves from 89.4% to 93.9%, demonstrating that GRONav adapts well to different encoders and that employing a stronger encoder indeed enhances model performance.

Table 11: Navigation performance (%) of GRONav-v2 and GRONav-v3 across CARLA scenes.

Method	Town02		Town03		Town10		Avg.	
	SR	SPL	SR	SPL	SR	SPL	SR	SPL
GRONav-v2	100.0	100.0	78.3	93.7	90.0	94.8	89.4	96.2
GRONav-v3	100.0	100.0	88.3	96.1	93.3	98.9	93.9	98.3

C.2 Supplementary results

Table 12 provides the full version of Table 5 in the main paper, reporting both success rate (SR) and progress rate (PR) for every keyframe selection method and navigation model.

Table 12: Navigation performance (SR and PR, %) on maps compressed by different keyframe selection methods, evaluated across three navigation models. **Bold** and *italic* denote the best and second-best results per navigation model.

Method	**AKS**						**ORB**					
	ViNT		EffoNAV		GRONav		ViNT		EffoNAV		GRONav	
Metric	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
Town02	91.7	98.8	50.0	83.6	75.0	91.9	95.0	98.1	83.3	95.5	75.0	90.4
Town03	65.0	88.0	46.7	75.2	73.3	92.3	68.3	83.2	33.3	75.8	77.7	92.7
Town10	21.7	54.3	16.7	55.6	53.3	86.4	20.0	55.8	50.0	75.9	53.3	87.7
Avg.	59.5	*80.4*	37.8	71.5	67.2	90.2	**61.1**	79.0	*55.5*	**82.4**	*68.7*	*90.3*
Method	**Uniform**						**SKS (ours)**					
	ViNT		EffoNAV		GRONav		ViNT		EffoNAV		GRONav	
Scene	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
Town02	93.3	99.3	76.7	86.2	55.0	80.0	95.0	99.1	83.3	95.9	85.0	97.4
Town03	60.0	79.3	40.0	77.9	73.3	93.0	58.3	92.7	50.0	84.0	81.7	91.9
Town10	25.0	56.0	33.3	70.7	48.3	79.9	28.3	59.0	36.7	61.5	73.3	90.4
Avg.	59.4	78.2	50.0	78.3	58.9	84.3	*60.5*	**83.6**	**56.7**	*80.5*	**80.0**	**93.2**

Following Wang & Chen (2025), we additionally report offline evaluation metrics on the four training datasets in Table 13 for reference, restricting the comparison to transformer-based methods. $\mathrm{Sim}(a_t, a_t^{gt})$ and $\mathrm{Sim}(w_t, w_t^{gt})$ measure the similarity between the predicted action/waypoints and the ground truth (↑); $\mathcal{L}_{\text{action}}$ and $\mathcal{L}_{\text{dist}}$ report the corresponding losses (↓).

Table 13: Offline quantitative comparison on four training datasets: RECON, Stanford, SACSoN, and SCAND. Higher Sim and lower losses are better.

Dataset	**Method**	$\mathrm{Sim}(a_t, a_t^{gt})$	$\mathrm{Sim}(w_t, w_t^{gt})$	$\mathcal{L}_{\text{action}}$	$\mathcal{L}_{\text{dist}}$	**Dataset**	**Method**	$\mathrm{Sim}(a_t, a_t^{gt})$	$\mathrm{Sim}(w_t, w_t^{gt})$	$\mathcal{L}_{\text{action}}$	$\mathcal{L}_{\text{dist}}$
RECON	ViNT	94.49	96.20	0.285	6.94	SACSoN	ViNT	89.66	93.16	0.686	10.95
	DynaNav	94.92	96.53	0.191	**6.26**		DynaNav	90.54	93.72	0.493	9.62
	EffoNAV	95.87	97.02	**0.120**	6.40		EffoNAV	92.28	93.36	**0.285**	**8.25**
	GRONav	**95.89**	**97.09**	0.123	6.30		**GRONav**	**93.36**	**95.22**	0.270	8.47
Stanford	ViNT	88.50	93.47	0.530	15.80	SCAND	ViNT	95.43	96.89	0.140	16.08
	DynaNav	89.07	93.66	0.450	14.23		DynaNav	96.85	97.03	0.130	14.41
	EffoNAV	89.23	94.93	0.320	12.30		EffoNAV	97.29	97.31	**0.100**	**11.45**
	GRONav	**91.26**	**95.77**	**0.290**	**11.89**		**GRONav**	**97.46**	**97.37**	0.110	11.64

For qualitative analysis, we provide additional attention heatmaps in Figure 10, showing that the policy consistently concentrates on obstacles and path boundaries when predicting waypoints.

Figure 11 visualizes the effect of SKS: blue nodes mark the trajectory under the dense map, and red nodes mark the trajectory under the compressed map. The compressed trajectory is dense at turns and sparse on straight sections, consistent with a compression scheme driven by visual change. Because SKS also uses semantic information, retained nodes remain dense in semantically rich regions even when the visual change is modest.

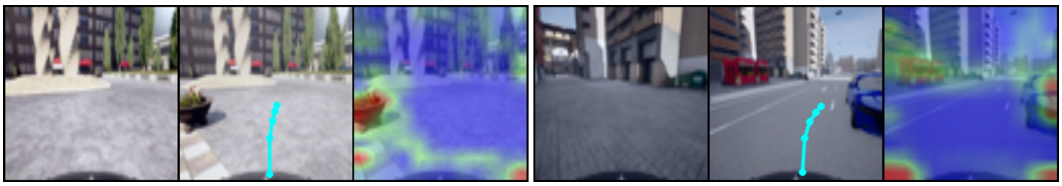


Figure 10: Additional attention-heatmap visualizations. Each row shows the target observation, the predicted waypoint trajectory, and the attention heatmap.

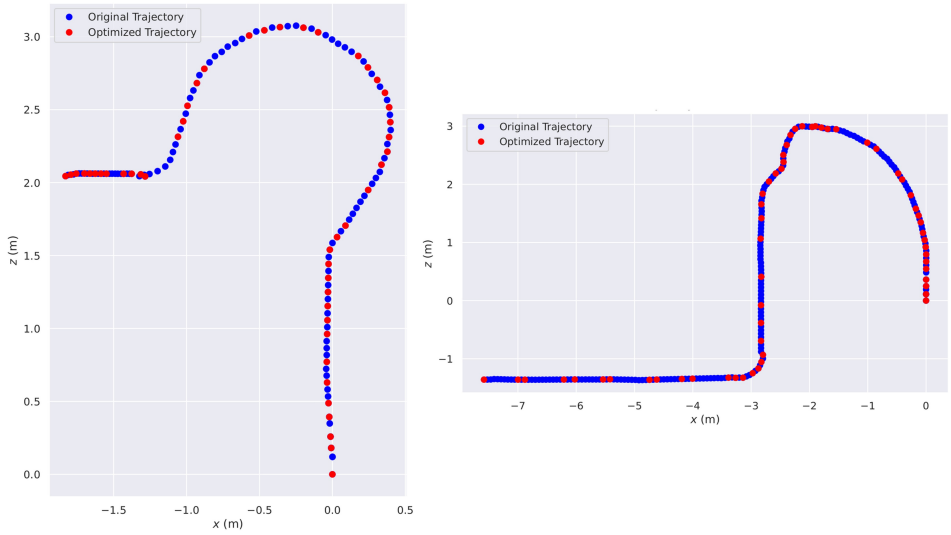


Figure 11: Visualization of SKS. Blue nodes denote the trajectory under the dense map; red nodes denote the trajectory under the compressed map.

D Relation to Standard VLN Benchmarks

Standard vision-language navigation (VLN) benchmarks on vision language navigation and the methods built on them are often suggested as evaluation targets for language-conditioned navigation. We do not adopt them, because the quantities they measure are defined by a map representation and an action space that our task does not have. Table 14 lists the differences that block a direct comparison.

Table 14: Why standard VLN benchmarks do not transfer to our setting.

	VLN tasks (R2R-CE, RxR-CE, LHPR-VLN, etc.)	Our work
Map	Prebuilt on indoor Matterport scans and given to the agent; nodes are panoramic viewpoints on a dense, fixed graph	Built online from a robot trajectory and then compressed; each node stores one egocentric frame
Action	Discrete viewpoint selection, or short continuous steps inside the same scanned scene	A continuous low-level policy executed over tens to hundreds of meters, indoors and outdoors
Instruction	Fine-grained, step-by-step directions tied to successive viewpoints	One long instruction grounded onto a sparse sequence of visually similar landmarks
Success	Reaching a designated viewpoint, or a nearby pose, on the given graph	Landmark disambiguation after compression, plus collision-free execution of the retrieved subgoals

Continuous variants do not remove the mismatch. R2R-CE (Krantz et al., 2020) replaces viewpoint teleportation with continuous control, and LHPR-VLN (Song et al., 2025) extends the horizon, but both still navigate a prebuilt indoor scan whose graph, scenes, and episodes are given in advance. Planners such as ETPNav (An et al., 2024) and MapGPT (Chen et al., 2024) are likewise tied to that graph. Our failure modes are different: retrieval among visually similar landmarks after aggressive compression, and collision-free control from a single forward-facing camera. Their episode splits, navigation graphs, and metrics therefore cannot be reused, and we instead follow the protocol of topological visual navigation (Shah et al., 2023c;a).

E LIMITATIONS

Most real-world failures occur in the "last-mile" phase near the goal, indicating limited fine-grained obstacle avoidance in cluttered environments; integrating a local reactive planner may alleviate this issue. In addition, GRONav assumes a pre-built topological graph and does not support dynamic map updates (e.g., new obstacles or structural changes). Extending the framework to incremental graph and semantic updates remains an open problem. Our real-world experiments reveal that most failures occur in the "last-mile" phase near the goal, suggesting that the current policy lacks fine-grained obstacle avoidance capability in cluttered spaces—integrating local planners or reactive controllers could address this. Finally, the current framework assumes a pre-collected topological graph and does not handle dynamic map updates (e.g., new construction or blocked paths); extending GRONav to incrementally update both the graph structure and its semantic annotations remains an open challenge.