VL-Nav: A Neuro-Symbolic Approach for Reasoning-based Vision-Language Navigation

Yi Du[1*], Taimeng Fu[1*], Zhipeng Zhao[1], Shaoshu Su[1], Zitong Zhan[1], Qiwei Du[1], Zhuoqun Chen[1], Bowen Li[2], Chen Wang[1]

[1]Spatial AI & Robotics Lab, University at Buffalo, USA

[2]Robotics Institute, Carnegie Mellon University, USA

{yidu, taimengf, cwx}@buffalo.edu

*Equal contribution.

Abstract

Navigating unseen, large-scale environments based on complex and abstract human instructions remains a formidable challenge for autonomous mobile robots. Addressing this requires robots to infer implicit semantics and efficiently explore large-scale task spaces. However, existing methods, ranging from end-to-end learning to foundation model-based modular architectures, often lack the capability to decompose complex tasks or employ efficient exploration strategies, leading to robot aimless wandering or target recognition failures. To address these limitations, we propose ***VL-Nav****, a neuro-symbolic (NeSy) vision-language navigation system. The proposed system intertwines neural reasoning with symbolic guidance through two core components: (1) a* ***NeSy task planner*** *that leverages a symbolic 3D scene graph and image memory system to enhance the vision-language models' (VLMs) neural reasoning capabilities for task decomposition and replanning; and (2) a* ***NeSy exploration system*** *that couples neural semantic cues with the symbolic heuristic function to efficiently gather the task-related information while minimizing unnecessary repeat travel during exploration. Validated on the DARPA TIAMAT Challenge navigation tasks, our system achieved an 83.4% success rate (SR) in indoor environments and 75% in outdoor scenarios. VL-Nav achieved an 86.3% SR in real-world experiments, including a challenging 483-meter run. Finally, we validate the system with complex instructions in a 3D multi-floor scenario. A preliminary version of this work is submitted to IROS 2026, and a more complete version is submitted to Science Robotics.*

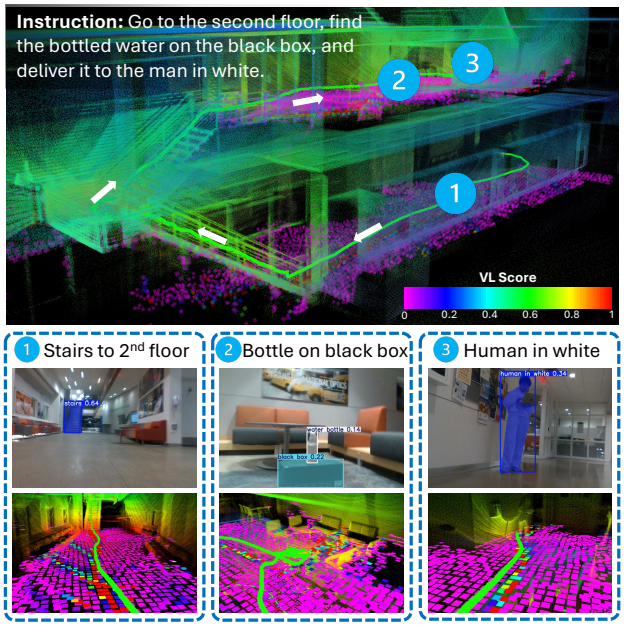

Figure 1. Given the complex instruction, VL-Nav autonomously navigates a university building. Top: The generated 3D global map and robot trajectory (green line). The projected value map (rainbow squares on the floor) visualizes the NeSy scoring policy, which prioritizes task-relevant areas to guide efficient exploration. Bottom: Key reasoning milestones where the system (1) identifies stairs for floor transition, (2) grounds the target object using spatial relation (on black box), and (3) verifies the recipient based on visual attributes.

1. Introduction

Efficient navigation following complex human instructions in completely unseen environments remains a formidable challenge for autonomous mobile robots. For example, consider a navigation task in the DARPA TIAMAT chal-

lenge [22]: "Today's weather report indicates rain. Help Rob find an umbrella, an appropriate jacket, and shoes." This marks a critical shift from naive command following vision-and-language navigaion [14] to reasoning-based vision-language navigation (VLN), where the robot must bridge massive logical gaps. The robot must first accurately understand the implicit semantics of the **abstract** instruction, reasoning out semantic inferences (e.g., "rain" implies waterproof gear) and performing object disambiguation to identify that the required targets are an umbrella, a rain jacket, and rain boots. This is essential to prevent the robot from looking for a normal jacket or sneakers as the target objects. Subsequently, it must efficiently explore the unknown environment to find all these required items within a limited task time.

However, a significant gap remains in addressing such complex VLN. Classical semantic-free approaches [9, 10, 33] lack the linguistic reasoning necessary to comprehend abstract human instructions. Conversely, end-to-end learning methods, such as RL-based [7, 19, 28] and VLA-based approaches [5, 37, 38], are notoriously data-hungry and struggle with sim-to-real transfer. While recent foundation model-based modular architectures [12, 35, 36, 39] integrate advanced VLMs for zero-shot single-object navigation, their tight coupling of target verification with exploration constrains inferential capabilities, frequently resulting in the identification of incorrect items (e.g., random shoes instead of rain boots). Furthermore, most of these methods over-rely on neural semantic cues while neglecting the potential of geometric frontiers, leading to aimless wandering within explored regions. Consequently, existing approaches fundamentally lack two critical capabilities: (1) *Accurate Task Reasoning* to parse and decompose abstract multi-target instructions; and (2) *Efficient Exploration Strategies* to rapidly locate multiple targets in large-scale environments.

To seamlessly bridge this gap, we propose a NeSy vision-language navigation system that fundamentally intertwines neural reasoning with symbolic guidance. Specifically, our VL-Nav architecture consists of two main modules: (1) **NeSy Task Planner:** This planner logically decomposes abstract, multi-target instructions into atomic subtasks, dynamically issuing either *exploration* subtasks to gather information or *go to* subtasks to navigate to and report target objects. Crucially, the planner grounds VLM's neural reasoning in a unified symbolic memory comprising a 3D scene graph and object-centric image memory, anchoring VLM decisions in precise spatial awareness and historical context. (2) **NeSy Exploration System:** Serving as the exploration backbone, this module couples neural semantic cues from lightweight VLMs with a symbolic heuristic function. This integration rapidly guides the robot through unknown environments to discover targets while minimizing unnecessary travel.

We comprehensively validate our proposed VL-Nav system in both simulations and extensive real-world robot environments. When evaluated in the navigation tasks in the DARPA TIAMAT challenge, our system demonstrated remarkable performance, achieving an 83.4% success rate (SR) in indoor settings and 75% in outdoor environments. Furthermore, real-world deployment across four diverse environments confirms the capability of generalizing from highly occluded, indoor domains to unstructured outdoor environments. Our real-world experiments achieved an 86.3% success rate, covering challenging scenarios such as a 483-meter long-range trajectory. Additionally, we showcased a complex multi-floor demonstration as shown in Fig. 1.

Our main contributions are as follows:

- We introduce VL-Nav, a NeSy system designed to resolve reasoning-based VLN tasks by intertwining neural semantic understanding with symbolic guidance.
- We introduce a robust NeSy Task Planner to guide VLM reasoning and replanning within a unified symbolic memory. When coupled with our NeSy Exploration System, this approach facilitates rapid multi-target discovery and minimizes unnecessary travel.
- VL-Nav achieves **83.4%** and **75%** SR in the indoor and outdoor settings on the navigation tasks of the DARPA TIAMAT Challenge, respectively. In real-world experiments, it attains an **86.3%** SR, including long-range runs of up to **483 meters** and demonstrations of complex tasks in **multi-floor** environments.

2. Related Works

Robot navigation methods can be broadly categorized into classical, end-to-end learning, and modular learning methods.

Classical Approaches: Classical methods for navigation, including Simultaneous Localization and Mapping (SLAM)-based techniques, have been extensively studied over the last three decades [9, 10, 31, 33]. These methods typically build geometric maps of an environment using depth sensors or monocular RGB cameras [8, 21, 27, 32], while simultaneously localizing the robot relative to the growing map. Exploration is often guided by heuristic strategies like frontier-based methods [11, 33], while analytical planners are employed for obstacle avoidance and path planning. Although effective for traditional navigation tasks, these classical methods lack the ability to integrate vision-language features, making them insufficient for human-collaboration tasks like vision-language navigation.

End-to-End Learning: End-to-end learning methods seek to directly map sensory inputs and instructions to navigation actions. Traditionally, these approaches relied on deep neural networks trained via imitation learning (IL) or rein-

forcement learning (RL) [7, 19, 26, 28, 30]. More recently, Vision-Language-Action (VLA) models [5, 37, 38] have emerged, attempting to unify vision, language, and continuous control into single massive transformer architectures. Despite their ability to inherently learn semantic priors for goal-directed exploration, these end-to-end approaches suffer from significant limitations when applied to real-world, multi-target VLN. They are notoriously data-hungry, computationally expensive to run in real-time, and lack interpretability. Most critically, they frequently fail to generalize beyond their training distributions, leading to severe performance degradation during sim2real transfer [13].

Modular Learning: Modular learning approaches seek to combine the advantages of classical and end-to-end methods by replacing specific components of the classical pipeline with learned modules. This modularity allows for the integration of semantic priors while maintaining system interpretability and efficiency [2, 3, 18, 20]. For example, semantic maps generated from RGB-D data have been used to guide navigation toward specific objects [4, 15, 24, 25]. Modular methods have shown promise in addressing Sim-to-Real transfer challenges by abstracting raw sensor data into higher-level representations, mitigating the impact of domain gaps [2, 3, 20]. However, existing modular approaches often rely heavily on real-world training data and struggle to utilize vision-language features.

Foundation Model-Based Modular Learning Methods: Recent advances in foundation models like large Vision-Language Models (VLMs) and Large Language Models (LLMs) have further enhanced semantic navigation by incorporating natural language understanding into modular learning approaches [12, 35, 36, 39]. For instance, the Vision-Language Frontier Maps (VLFM) method [36] leverages a pre-trained VLM to extract semantic values directly from RGB images, enabling zero-shot semantic prior understanding for frontier-based exploration. Other methods like ApexNav [39] and SG-Nav [35] build open-vocabulary representations to find novel categories. However, these systems inherently operate under the premise of naive instruction following—where the "target" is explicitly named (e.g., "go to the L-shape sofa"). They lack the complex logic required for Reasoning-based VLN needed in abstract and multi-target tasks where explicit targets are not provided (e.g., inferring that "rain" requires an "rain jacket" instead of a random coat). Furthermore, executing heavy foundation models all the time during the navigation process bottlenecks real-time operation on resource-constrained platforms.

3. Methodology

To address the limitations of existing methods in resolving abstract and multi-target tasks, we introduce **VL-Nav**, a neuro-symbolic vision-language navigation system that intertwines neural semantic understanding with symbolic precision. As shown in Fig. 2, the architecture consists:

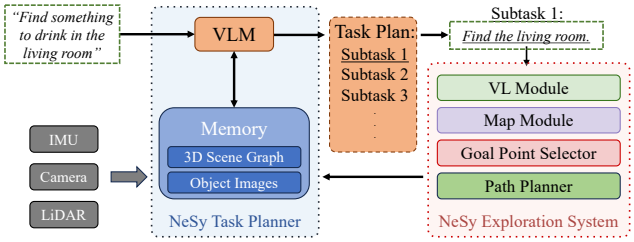

Figure 2. System pipeline overview.Complex tasks are decomposed into subtasks via VLM reasoning, guided by the unified memory system. The first subtask always serves as input to the NeSy exploration system, which either guides exploration or sets a direct goal point for robot navigation.

1. **NeSy Task Planner** logically decomposes complex, abstract instructions into atomic subtasks, *exploration* and *go to*, by guiding a VLM's neural reasoning with a unified symbolic memory comprising a 3D scene graph and an object-centric image memory for task decomposition, replanning, and completion determination.
2. **NeSy Exploration System** couples neural semantic cues from lightweight open-vocabulary detection models (OVDMs) with a symbolic heuristic function, enabling efficient task-guided exploration of unknown environments without the computational bottlenecks of pure VLM-in-the-loop approaches.

3.1. NeSy Task Planner

3.1.1. Unified Memory System

The Unified Memory System consists of a 3D scene graph and an object-centric image memory. First, we construct a 3D scene graph comprising two types of nodes: object nodes and room nodes, inspired by Hydra [16]. Room node masks are generated using our room segmentation algorithm, which relies on morphological operations. To label these room nodes, we utilize a Large Language Model (LLM) that infers labels based on the object nodes connected to each room node. The edges between object and room nodes are established through spatial inclusion; specifically, an edge is created if an object's centroid falls within a room's segmentation mask.

Object nodes are generated by an open-vocabulary detector. Each object node stores four key attributes: (1) the object centroid; (2) the highest-confidence detection score; (3) the robot's pose at the viewpoint of this detection; and (4) the corresponding best-viewpoint RGB image. This integration of the 3D scene graph with object-centric best-viewpoint image memory forms a *unified symbolic memory*, providing the spatial and relational grounding necessary for subsequent task decomposition and replanning.

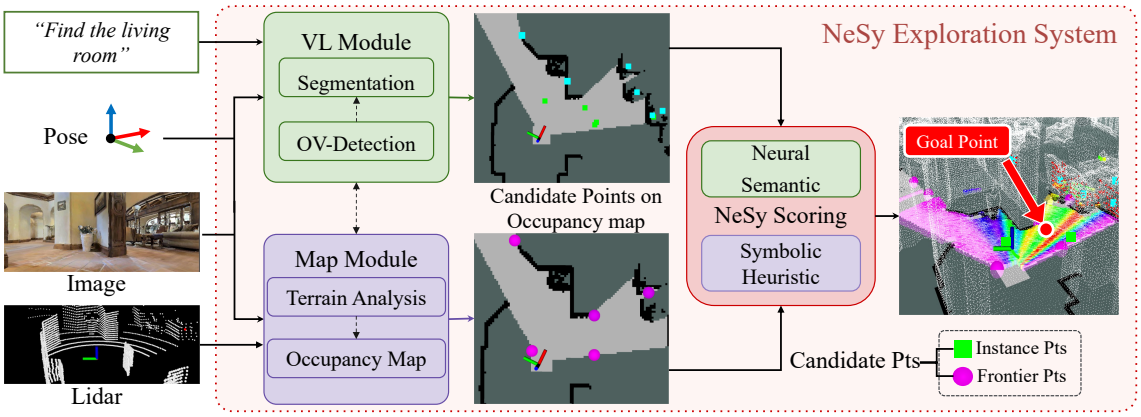

Figure 3. An overview of the NeSy exploration system. It processes inputs including prompts, RGB images, poses, and LiDAR scans. The Vision-Language (VL) module conducts open-vocabulary detection to identify objects related to the prompt, generating instance-based target points. Concurrently, the map module performs terrain analysis and manages a dynamic occupancy map. Frontier-based target points are then identified based on this occupancy map, along with the instance points, forming a candidate points pool. Finally, this system employs the NeSy scoring to select the most effective goal point.

3.1.2. Task Decomposition and Replanning

Leveraging Qwen3-VL [1] as the VLM backbone, the planner decomposes complex task descriptions into atomic "exploration" and "go to" subtasks. The VLM maintains a dynamic target object list, querying the memory module to track and register potential candidates. Upon the completion of each subtask, a replanning phase will be triggered, generating updated subtasks based on the updated symbolic memory.

For target acquisition, the system employs a coarse-to-fine verification strategy. For instance, when locating a specific item like a "rain jacket," the "go to" subtask is determined via a two-step process: **(1) Symbolic Filtering:** The 3D scene graph filters the available object nodes to propose the top-k (e.g., top-3) candidates based on detection confidence. **(2) Neural Verification:** The VLM performs fine-grained reasoning on the saved best-view images and the neighbor scene graph nodes of these top candidates to determine which object best semantically matches the abstract instruction. Once the target is chosen, the planner publishes the saved best-view pose as the navigation goal, directing the exploration system to navigate to and report the identified target.

3.2. NeSy Exploration System

Taking the subtask from the task planner, the **NeSy Exploration System** operationalizes high-level directives by fusing neural semantic cues from open-vocabulary detectors with symbolic geometric heuristics. The pipeline as shown in Fig. 3 proceeds from the generation of candidate goal points to the hybrid NeSy scoring policy, facilitating rapid, task-guided exploration and navigation in real-time.

3.2.1. Frontier-based Target Points

A cell (m_x, m_y) is a *frontier* if it is free, $\mathcal{G}(m_x, m_y) = 0$, and at least one neighbor is unknown, $\mathcal{G}(m_x + \Delta x, m_y + \Delta y) = -1$ for some $(\Delta x, \Delta y)$. After every grid update, only cells in the forward FoV wedge are tested:

$$\begin{aligned} \left|\mathrm{Angle}\big((m_x, m_y), (x_r, y_r)\big) - \theta_r\right| &\leq \frac{\mathrm{hfov}}{2}, \\ \text{and} \quad \|(m_x, m_y) - (x_r, y_r)\| &\leq R, \end{aligned} \tag{1}$$

where (x_r, y_r) and θ_r denote the robot's current position and heading, hfov represents the horizontal field of view, and R is the maximum effective sensor range.

A breadth-first search (BFS) clusters these frontier cells. Each cluster is represented by either a single centroid (for small clusters) or multiple sampled points (for large clusters) as the frontier-based target points.

3.2.2. Instance-Based Target Points (IBTP)

The vision-language detector shown in Fig. 3 VL Module periodically reports *candidate instance centers* in the form $(q_x,\ q_y,\ \text{confidence})$, where q_x, q_y denote the estimated global coordinates for a potential target instance, and confidence quantifies how likely this detection is to match the desired instance. If confidence is larger than the detection threshold τ_{det}, the candidate is retained; otherwise, it is discarded as too uncertain. Retained points will be downsampled via a voxel-grid filter if many candidates lie in close proximity.

This *instance-based* approach mimics human behavior when searching for an object: upon glimpsing something that *might* match the target, one naturally moves closer to confirm. VL-Nav likewise does not ignore intermediate detections. Instead, any candidate above a confidence threshold is treated as a valid *goal candidate*, allowing the robot

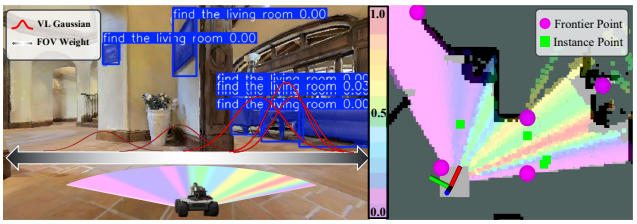

Figure 4. A brief illustration of VL Scoring. The pixel-wise open-vocabulary detection results are transferred into the spatial distibution via the Gaussian mixture model regularized by the FoV weighting (the gray arrows). Then the frontier-based and the instance-based target points will be assigned with VL score based on the distribution Equation (3).

to approach and verify whether it truly is an instance of interest. If the detection turns out to be incorrect, the robot continues exploring via frontiers or other instance cues.

3.2.3. NeSy Scoring Policy

Once frontier-based and instance-based target points are gathered as candidate goals, the system computes the NeSy score for each candidate goal $\mathbf{g}$. Let (x_r, y_r) be the robot position, θ_r the heading, and $d(\mathbf{x}_r, \mathbf{g})$ the Euclidean distance. The angular offset is:

$$\Delta\theta \;=\; \mathrm{Angle}\big(\mathbf{g}, (x_r, y_r)\big) \;-\; \theta_r. \tag{2}$$

VL Score: As shown in Fig. 4, a vision-language (VL) score $S_{\mathrm{VL}}(\mathbf{g})$ translates *fine-grained vision-language features* from the YOLO-World [6] and FastSAM [41] models into a Gaussian-mixture distribution over the robot's horizontal field of view (FoV). Suppose the open-vocabulary detection model identifies K *likely directions* or *detection regions*, each parameterized by $\big(\mu_k, \sigma_k, \alpha_k\big)$. Here, μ_k indicates the mean offset angle within the FoV, σ_k encodes the detection's angular uncertainty (in our implementation, a fixed constant of 0.1 for all detections), and α_k is a confidence-based weight that captures how important that region is. By accumulating each region's contribution in a mixture model, $S_{\mathrm{VL}}(\mathbf{g})$ naturally biases the robot toward goals in the direction of high-confidence detections that match the subtask prompt.

From a human perspective, this process resembles *observing* different areas that related to the target instance with varying levels of certainty. For instance, if someone is looking for a "red chair" in a large room, they might spot multiple *red silhouettes* at the right side of their vision, plus a clearer view of something red near the center. Although peripheral glimpses ("out of the corner of the eye") often carry lower confidence, they are not ignored. Instead, we naturally merge these partial observations into a mental sense of "the chair is probably in one of these directions." The system follows a similar approach, assigning each detection region a Gaussian shape around μ_k, scaled by σ_k and weighted by α_k. Inspired by VLFM [36], we then multiply by $C(\Delta\theta)$, a "view confidence" term that downweights detections with large angular offsets from the central field of view. The VL score is computed as following equations:

$$S_{\mathrm{VL}}(\mathbf{g}) \;=\; \sum_{k=1}^{K} \alpha_k \;\exp\!\Big(-\tfrac{1}{2}\Big(\tfrac{\Delta\theta-\mu_k}{\sigma_k}\Big)^2\Big) \;\cdot\; C\big(\Delta\theta\big), \tag{3}$$

$$C(\Delta\theta) = \cos^2\!\Big(\tfrac{\Delta\theta}{(\theta_{\mathrm{fov}}/2)} \cdot \tfrac{\pi}{2}\Big). \tag{4}$$

Finally, to ensure the metric remains bounded, the accumulated score $S_{\mathrm{VL}}(\mathbf{g})$ is clipped to the range $[0, 1]$.

Curiosity Cues: We add two curiosity terms to guide navigation toward larger unexplored areas and prevent the system from repeatedly selecting faraway goals that offer only a marginal increase in VL score—thus reducing unnecessary back-and-forth movement which is extremely important for the large-scale environments like the hallway and the outdoor environments in our experiments.

(1) Distance Weighting. We define:

$$S_{\mathrm{dist}}(\mathbf{g}) \;=\; \tfrac{1}{1+d(\mathbf{x}_r,\mathbf{g})}, \tag{5}$$

so that nearer goals receive slightly higher scores. This factor is especially important on a real robot, as shorter travel distances can significantly reduce energy consumption while preventing needless wandering. Although the distance term alone does not guarantee an optimal path, it helps prioritize goals that can be reached sooner.

(2) Unknown-Area Weighting. We further encourage curiosity-driven exploration by measuring how many unknown cells lie around $\mathbf{g}$. First, we define the ratio of unknown as:

$$\mathrm{ratio}(\mathbf{g}) \;=\; \tfrac{\#(\text{unknown cells})}{\#(\text{total visited cells})}, \tag{6}$$

where a local BFS radiates outward from $\mathbf{g}$, counting how many cells are unknown versus how many are reachable (i.e., not blocked by obstacles). Larger $\mathrm{ratio}(\mathbf{g})$ implies that moving to $\mathbf{g}$ may reveal significant unknown space and get more information. To translate this raw ratio into a normalized weighting score, we apply an exponential mapping:

$$S_{\mathrm{unknown}}(\mathbf{g}) \;=\; 1 \;-\; \exp\!\Big(-\,k\,\mathrm{ratio}(\mathbf{g})\Big), \tag{7}$$

where k is an adjustable parameter that controls how rapidly the score increases from 0 toward 1.

Combined NeSy Score: The final scores of frontier-based goals combine these three components:

$$S_{\mathrm{NeSy}}(\mathbf{g}) \;=\; \Big(w_{\mathrm{dist}}\, S_{\mathrm{dist}}(\mathbf{g}) \;+\; w_{\mathrm{VL}}\, S_{\mathrm{VL}}(\mathbf{g}) \;\cdot\; S_{\mathrm{unknown}}(\mathbf{g})\Big), \tag{8}$$

where w_{VL} and w_{dist} are scalar weights. Note that the VL score is assigned when a point is initially detected, whereas

the curiosity terms (distance and unknown-area weighting) are evaluated at goal selection time, when the current distance and unknown ratio can be calculated. For instance-based goals, these curiosity cues are omitted because their primary purpose is verification rather than exploration.

3.3. Goal Selection & Path Planning

After computing scores, the system selects the optimal goal point by prioritizing high-confidence semantic matches over general exploration. Specifically, we iterate through all available *instance-based target points* that exceed a minimum distance threshold δ_{reached}. If any valid instance targets exist, the one with the highest S_{VL} is selected immediately to drive verification. If no valid instance targets are available (or all are too close), the system falls back to *frontier-based target points*. The frontier with the highest combined S_{NeSy} score is chosen to maximize information gain. If no candidate exceeds δ_{reached}, the robot holds its position. Once a goal is selected, we employ the FAR Planner [34] to generate a collision-free path, guiding the robot to the target while dynamically avoiding real-time obstacles.

4. Experiments

4.1. Simulation Experiments: DARPA TIAMAT Challenge

We first validate VL-Nav in the high-fidelity simulation environment of the DARPA TIAMAT Challenge (Phase 1). This benchmark is designed to test robotic reasoning under highly abstract and complex human instructions.

Baselines: We compare VL-Nav against four well-known baselines and two ablation variants: closest frontier exploration, VLFM [36], SG-Nav [35], ApexNav [39], VL-Nav w/o IBTP (ablation), and VL-Nav w/o Curiosity (ablation).

Environments & Simulators: The evaluation covers four distinct scenarios simulated across two platforms:

- **HabitatSim (Indoor):** The *Apartment 1* and *Apartment 2* environments are rendered in HabitatSim [29] to evaluate navigation in daily residential settings.
- **IsaacSim (Outdoor):** The *Camping Site* and *Factory* environments are simulated in IsaacSim [23] to leverage its advanced physics engine for unstructured settings.

Task Settings: Except for interaction, each environment consists of **8 abstract navigation tasks** requiring multi-step reasoning. To ensure reliability, we conduct 3 trials for all tasks in each environment. Crucially, some tasks require the robot to locate multiple target objects distributed across widely separated positions within a large-scale map. Instructions involve implicit semantics (e.g., "it is raining" $\rightarrow$ find umbrella), hierarchical searching (e.g., "check the garage for tools"), and abstract social concepts (e.g., "prepare for a black-tie party" $\rightarrow$ find suit, shoes, and tie).

Robot and Sensor Setup: The DARPA TIAMAT tasks require a Boston Dynamics Spot robot, equipped with five RGB-D cameras positioned to provide a comprehensive surround view for visual reasoning.

4.2. Real-World Experiments

Baselines: Since SG-Nav [35] and ApexNav [39] cannot be deployed on the edge device, we evaluate our approach in real-robot experiments against five methods: (1) closet frontier-based exploration, (2) VLFM [36], (3) VL-Nav without instance-based target points, (4) VL-Nav without curiosity terms, and (5) the full VL-Nav configuration. Because the original VLFM relies on BLIP-2 [17], which is too computationally heavy for real-time edge deployment, we use the same YOLO-World [6] and FastSAM [41] models instead to generate per-observation similarity scores for VLFM. Each method is tested under the same conditions to ensure a fair comparison of performance.

Environments: We consider four distinct environments (shown in Fig. 5), each with a specific combination of semantic complexity (*High*, *Medium*, or *Low*) and size (*Big*, *Mid*, or *Small*). Concretely, we use a Hallway (*Medium & Big*), an Office (*High & Mid*), an Outdoor area (*Low & Big*), and an Apartment (*High & Small*).

Task Settings: We define nine distinct, uncommon human-described instances to serve as target objects or persons during navigation (e.g., "tall white trash bin," "there seems to be a man in white"). This results in 9 navigation tasks in total, and we conduct 3 to 5 trials for each task to robustly evaluate performance. This ensures that the robot must rely on NeSy reasoning to accurately locate these targets.

Real Robot Implementation: Experiments are conducted on a four-wheeled Rover, with a Unitree Go2 quadruped used for the multi-floor demonstration. Both robots are equipped with a Livox Mid-360 LiDAR (tilted 23°) and an Intel RealSense D455 camera (tilted 7°). State estimation is handled by Super Odometry [40], ensuring resilient robot state estimation throughout the mission. The Go2 is additionally equipped with a Unitree D1 arm and a RealSense D435 camera to enable object grasping. Computation is distributed: the real-time navigation stack runs onboard on an NVIDIA Jetson Orin NX, while the NeSy Task Planner runs on a remote laptop with an RTX 4090 GPU to host the Qwen3-VL-8B model [1]. This separation is effective as the task planner operates asynchronously, updating only when a new task is received or a subtask is completed.

4.3. Main Results

We validate the proposed VL-Nav system through rigorous testing in both the high-fidelity DARPA TIAMAT simulation and diverse real-world deployments. Our evaluation employs distinct metrics suited for each domain: for simulation, we report Success Rate (SR) and Max Time Usage

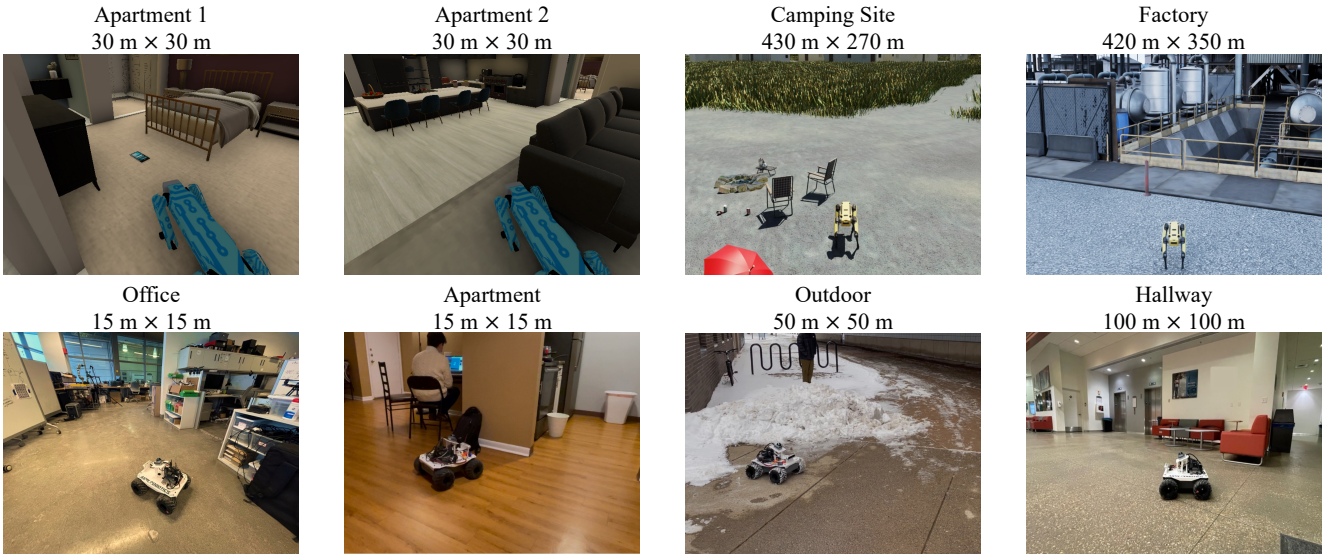

Figure 5. **Top:** Four simulation environments in the DARTPA TIAMAT Challenge. **Bottom:** Four real-world experiment environments.

Table 1. Simulation experiment results on the navigation tasks of DARPA TIAMAT challenge phase 1 in 4 different scenarios with metrics Success Rate (SR) and Max Time Usage Ratio (MTUR).

Method	**SR % (↑)**				**MTUR (↓)**			
	Apartment 1	Apartment 2	Camping Site	Factory	Apartment 1	Apartment 2	Camping Site	Factory
Frontier Exploration	8.3	8.3	0.0	0.0	0.958	0.884	1.000	1.000
VLFM [36]	8.3	8.3	4.2	8.3	0.931	0.864	0.953	0.859
SG-Nav [35]	0.0	4.2	0.0	8.3	1.000	0.973	1.000	0.901
ApexNav [39]	25.0	25.0	20.8	12.5	0.817	0.795	0.828	0.861
VL-Nav w/o IBTP	70.8	62.5	62.5	58.3	0.680	0.724	0.731	0.762
VL-Nav w/o Curiosity	79.1	75.0	58.3	66.7	0.612	0.635	0.793	0.735
VL-Nav	**87.5**	**79.2**	**75.0**	**75.0**	**0.562**	**0.591**	**0.647**	**0.679**

Ratio (MTUR), defined as the average execution time divided by the maximum time limit per task, to quantify time efficiency and completion reliability. For real-world experiments, we adopt SR and Success weighted by Path Length (SPL) following the VLFM convention [36] to evaluate efficiency.

Simulation Results (DARPA TIAMAT) As reported in Table 1, VL-Nav achieves dominant performance with SRs ranging from 75.0% to 87.5%, whereas baselines struggle.

- **Baseline Failures and High MTUR:** Standard approaches fail to resolve the abstract logic and multi-target sequencing required by the benchmark. Notably, **SG-Nav** [35] exhibits an MTUR near 1.0 across environments, frequently timing out. This is primarily because its reliance on LLM-based Chain-of-Thought reasoning at *every navigation step* incurs prohibitive latency [35]. Similarly, **ApexNav** [39] suffers from low success (SR $\approx$ 12.5%–25.0%) as it lacks the consistent symbolic memory to efficiently manage long-horizon and multiple goals located at different positions.
- **Qualitative Reasoning:** The system's ability to ground abstract concepts is visualized in Figures 6 and 7. In indoor settings (Fig. 6), VL-Nav successfully infers implicit semantics (e.g., "weather" $\rightarrow$ "rain jacket") and performs hierarchical decomposition. In unstructured outdoor settings (Fig. 7), the projected value maps demonstrate how the system targets large landmarks to guide the search for smaller, distributed objects (laptop, outfit, truck) without getting lost in open space.

Real-World Results Physical deployments (Table 2) confirm the system's robustness and efficiency.

- **Efficiency (SPL):** VL-Nav achieves significantly higher SPL scores (e.g., 0.812 in Office) compared to Frontier Exploration (0.317) and VLFM (0.556). This metric confirms that VL-Nav does not merely find targets by chance but navigates via efficient, non-redundant paths, verifying that the neuro-symbolic planning translates effectively to real-world dynamics.

Table 2. Vision-language navigation performance in 4 unseen environments (SR and SPL).

Method	SR % (↑)				SPL (↑)			
	Hallway	Office	Apartment	Outdoor	Hallway	Office	Apartment	Outdoor
Frontier Exploration	40.0	41.7	55.6	33.3	0.239	0.317	0.363	0.189
VLFM [36]	53.3	75.0	66.7	44.4	0.366	0.556	0.412	0.308
VL-Nav w/o IBTP	66.7	83.3	70.2	55.6	0.593	0.738	0.615	0.573
VL-Nav w/o curiosity	73.3	86.3	66.7	55.6	0.612	0.743	0.631	0.498
VL-Nav	**86.7**	**91.7**	**88.9**	**77.8**	**0.672**	**0.812**	**0.733**	**0.637**

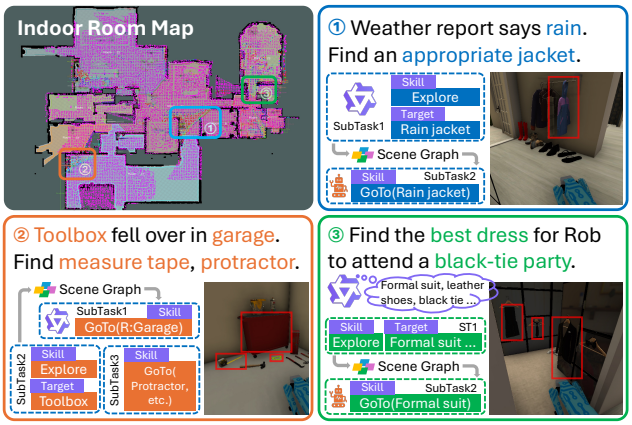

Figure 6. **Qualitative results in the DARPA TIAMAT Indoor-2 environment.** We demonstrate the system's capability to resolve three distinct types of complex instructions: (1) implicit semantic inference (weather → rain jacket); (2) hierarchical task decomposition (garage → tools); and (3) abstract concept grounding (black-tie party → formal suit).

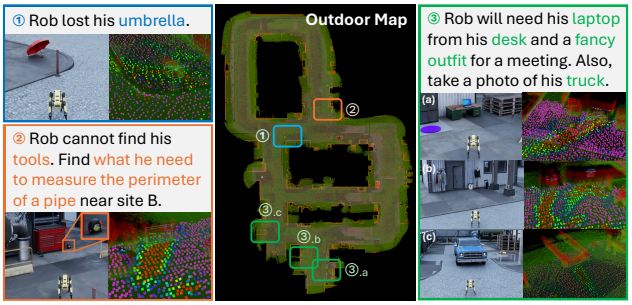

Figure 7. **Qualitative results in the DARPA TIAMAT outdoor factory environment.** We demonstrate robust navigation under unstructured conditions across three scenarios: (1) single-object retrieval; (2) implicit inference (measure the perimeter of a pipe → measuring tape); and (3) a challenging multi-objective task where the robot must locate three distinct targets scattered across different regions of the large-scale map. The colored point clouds in the insets visualize the value map guiding the exploration.

- **Sim-to-Real Transfer:** The consistent performance gap between VL-Nav and baselines across both simulation and real-world experiments highlights the system's strong generalization capabilities.

Ablation Analysis

- **Impact of IBTP (Verification):** The *w/o IBTP* variant suffers most in semantically complex and cluttered environments, such as the *Apartment* (Sim: 87.5% → 70.8%; Real: 88.9% → 70.2%). Without IBTP, the robot lacks the "short-cut" mechanism to verify tentative detections, leading to missed targets and lower SPL.
- **Impact of Curiosity (Exploration):** The *w/o Curiosity* variant degrades primarily in large-scale, open environments, such as the *Factory* and *Outdoor* scenes (Sim Factory: 75.0% → 66.7%; Real Outdoor: 77.8% → 55.6%). This confirms that distance and unknown-area weighting are critical for preventing local minima and aimless wandering in expansive maps.

Summary The combined results demonstrate that while baselines like SG-Nav and ApexNav stumble on the computational and logical demands of abstract multi-target tasks, VL-Nav's neuro-symbolic architecture provides the necessary reasoning depth and execution efficiency to succeed in both high-fidelity simulation and real-world environments.

5. Conclusions and Future Work

In this paper, we presented VL-Nav, a neuro-symbolic framework that bridges the gap between abstract human instructions and robotic execution. By intertwining a NeSy Task Planner with a NeSy Exploration System, our approach effectively addresses the logical and spatial reasoning challenges inherent in large-scale vision-language navigation. Extensive validation across the DARPA TIAMAT challenge and diverse real-world experiments demonstrates the system's robustness, achieving overall success rates of 79.2% and 86.3%, respectively. These results confirm VL-Nav's capability to complete complex tasks in large-scale unseen environments. Future work will focus on extending the symbolic memory to support temporal reasoning, enabling the robot to track moving targets and adapt to dynamic environmental changes. Additionally, we aim to incorporate lifelong learning capabilities, allowing the system to adapt to evolving environments over long-term deployments.

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. 4, 6

[2] D Chaplot, D Gandhi, A Gupta, and R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. arxiv 2020. *arXiv preprint arXiv:2007.00643*, 2007. 3

[3] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020. 3

[4] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020. 3

[5] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. *arXiv preprint arXiv:2412.04453*, 2024. 2, 3

[6] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16901–16911, 2024. 5, 6

[7] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4693–4700. IEEE, 2018. 2, 3

[8] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007. 2

[9] Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989. 2

[10] Claude Fennema and Allen R Hanson. Experiments in autonomous navigation. In *[1990] Proceedings. 10th International Conference on Pattern Recognition*, pages 24–31. IEEE, 1990. 2

[11] Luigi Freda and Giuseppe Oriolo. Frontier-based probabilistic strategies for sensor-based exploration. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3881–3887. IEEE, 2005. 2

[12] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23171–23181, 2023. 2, 3

[13] Theophile Gervet, Soumith Chintala, Dhruv Batra, Jitendra Malik, and Devendra Singh Chaplot. Navigating to objects in the real world. *Science Robotics*, 8(79):eadf6991, 2023. 3

[14] Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7606–7623, 2022. 2

[15] Meera Hahn, Devendra Singh Chaplot, Shubham Tulsiani, Mustafa Mukadam, James M Rehg, and Abhinav Gupta. No rl, no simulation: Learning to navigate without navigating. *Advances in Neural Information Processing Systems*, 34:26661–26673, 2021. 3

[16] Nathan Hughes, Yun Chang, Siyi Hu, Rajat Talak, Rumaia Abdulhai, Jared Strader, and Luca Carlone. Foundations of spatial perception for robotics: Hierarchical representations and real-time systems. *The International Journal of Robotics Research*, 43(10):1457–1505, 2024. 3

[17] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 6

[18] Rowan Thomas McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. International Joint Conferences on Artificial Intelligence, Inc., 2017. 3

[19] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673*, 2016. 2, 3

[20] Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, and Vladlen Koltun. Driving policy transfer via modularity and abstraction. *arXiv preprint arXiv:1804.09364*, 2018. 3

[21] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 2

[22] Erfaun Noorani, Zachary Serlin, Ben Price, and Alvaro Velasquez. From abstraction to reality: Darpa's vision for robust sim-to-real autonomy. *AI Magazine*, 46(2):e70015, 2025. 2

[23] NVIDIA. Isaac sim. https://github.com/isaac-sim/IsaacSim, 2025. 6

[24] Santhosh K Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Occupancy anticipation for efficient exploration and navigation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 400–418. Springer, 2020. 3

[25] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18890–18900, 2022. 3

[26] Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search

strategies from human demonstrations at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5173–5183, 2022. 3

[27] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8601–8610, 2018. 2

[28] Manolis Savva, Angel X Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. Minos: Multimodal indoor simulator for navigation in complex environments. *arXiv preprint arXiv:1712.03931*, 2017. 2, 3

[29] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019. 6

[30] Kunal Pratap Singh, Jordi Salvador, Luca Weihs, and Aniruddha Kembhavi. Scene graph contrastive learning for embodied navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10884–10894, 2023. 3

[31] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B Cremers, Frank Dellaert, Dieter Fox, Dirk Hahnel, Charles Rosenberg, Nicholas Roy, Jamieson Schulte, et al. Minerva: A second-generation museum tour-guide robot. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*. IEEE, 1999. 2

[32] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128(1-2):99–141, 2001. 2

[33] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, pages 146–151. IEEE, 1997. 2

[34] Fan Yang, Chao Cao, Hongbiao Zhu, Jean Oh, and Ji Zhang. Far planner: Fast, attemptable route planner using dynamic visibility update. In *2022 ieee/rsj international conference on intelligent robots and systems (iros)*, pages 9–16. IEEE, 2022. 6

[35] Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. *Advances in Neural Information Processing Systems*, 37:5285–5307, 2025. 2, 3, 6, 7

[36] 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)*, pages 42–48. IEEE, 2024. 2, 3, 5, 6, 7, 8

[37] 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. *arXiv preprint arXiv:2412.06224*, 2024. 2, 3

[38] 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*, 2025. 2, 3

[39] Mingjie Zhang, Yuheng Du, Chengkai Wu, Jinni Zhou, Zhenchao Qi, Jun Ma, and Boyu Zhou. Apexnav: An adaptive exploration strategy for zero-shot object navigation with target-centric semantic fusion. *IEEE Robotics and Automation Letters*, 2025. 2, 3, 6, 7

[40] Shibo Zhao, Sifan Zhou, Yuchen Zhang, Ji Zhang, Chen Wang, Wenshan Wang, and Sebastian Scherer. Resilient odometry via hierarchical adaptation. *Science Robotics*, 10 (109):eadv1818, 2025. 6

[41] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023. 5, 6