2026 IEEE International Conference on Robotics and Automation (ICRA)
June 1-5, 2026. Vienna, Austria

2026 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3315-8160-2/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICRA57385.2026.11697371

TopoNav: Topological Graphs as a Key Enabler for Advanced Object Navigation

Peiran Liu[1,2,*], Qiang Zhang[1,2,*], Daojie Peng[1,*], Lingfeng Zhang[3,*],
Yihao Qin[1], Hang Zhou[1], Jun Ma[1,†], Renjing Xu[1,†], Yiding Ji[1,†]

***Abstract*— Object Navigation (ObjectNav) has made great progress with large language models (LLMs), but still faces challenges in memory management, especially in long-horizon tasks and dynamic scenes. To address this, we propose TopoNav, a new framework that leverages topological structures as spatial memory. By building and updating a topological graph that captures scene connections, adjacency, and semantic meaning, TopoNav helps agents accumulate spatial knowledge over time, retrieve key information, and reason effectively toward distant goals. Our experiments show that TopoNav achieves state-of-the-art performance on benchmark ObjectNav datasets, with higher success rates and more efficient paths. It particularly excels in diverse and complex environments, as it connects temporary visual inputs with lasting spatial understanding.**

***Index Terms*— object navigation, vision language model, topological memory graph, spatial reasoning, point cloud**

I. Introduction

ObjectNav is a pivotal task in embodied AI and robotic interaction, requiring seamless integration of visual perception, natural language understanding, and spatial reasoning to enable agents to navigate in unknown environment to find specifed target object. In recent years, we have witnessed remarkable advancements in ObjectNav, largely driven by the rapid evolution of LLMs, which have significantly enhanced capabilities in instruction parsing, cross-modal alignment, and context-aware reasoning. State-of-the-art frameworks leveraging pre-trained LLMs and vision-language encoders have achieved impressive results in simple or constrained scenarios, underscoring ObjectNav's potential as a cornerstone for intelligent agent-environment interaction.

Despite these progresses, we observe that current ObjectNav methods still face critical bottlenecks, particularly in complex, dynamic, or large-scale environments. A key limitation lies in the lack of robust spatial memory mechanisms: most approaches rely heavily on ephemeral visual observations, struggling to accumulate and retain long-term structural knowledge of environments. This deficiency leads to fragmented memory of environmental structures, often resulting in goal confusion, redundant paths, or even complete navigation failure during long-horizon tasks—hindering the generalization of navigation systems to real-world, unconstrained scenarios.

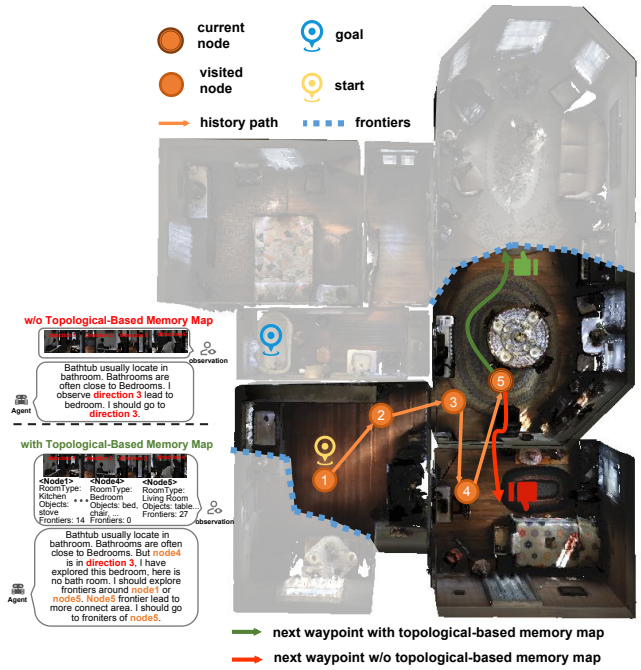

Fig. 1: Our memorization technique allows backtracking to key nodes when exploration directions are suboptimal, enhancing navigation efficiency via historical trajectory recall.

Our key insight is that topological structures inherently operate as a form of compact and durable spatial memory. Unlike pixel-level visual details being transient and data-heavy, topological information captures the essence of environmental structure in a stable, decision-supporting format. Such properties make topology uniquely suited to address navigation's memory deficits. Although topological maps have long been used in traditional robotic navigation, their potential as dynamic memory carriers in ObjectNav—especially in conjunction with language-driven spatial reasoning—remains largely untapped, creating a critical research gap.

To address this gap, we introduce **TopoNav**, a novel framework that constructs and maintains a dynamic topological memory graph as the core of its navigation system. To-

*Equal contribution; †Corresponding author.
[1]The authors are with the Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. (emails: {pliu868, qzhang749, dpeng108, yqin637, hzhou269}@connect.hkust-gz.edu.cn, {eejma, renjingxu, jiyiding}@hkust-gz.edu.cn.)
[2]The authors are with Beijing Innovation Center of Humanoid Robotics Co., Ltd, Beijing, China. (email: jony.zhang@x-humanoid.com)
[3]The author is with Shenzhen International Graduate School, Tsinghua University, Shenzhen, China. (email: zlf25@mails.tsinghua.edu.cn)
This work is supported in part by National Natural Science Foundation of China grants 62303389 and 62373289; Guangdong Basic and Applied Basic Research Funding grant 2024A1515012586; Guangdong Scientific Research Platform and Project Scheme grant 2024KTSCX039; Youth Talent Support Program of Guangdong Association for Science and Technology grant SKXRC2025463 and Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things (No.2023B1212010007).

Authorized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 01,2026 at 02:41:19 UTC from IEEE Xplore. Restrictions apply.

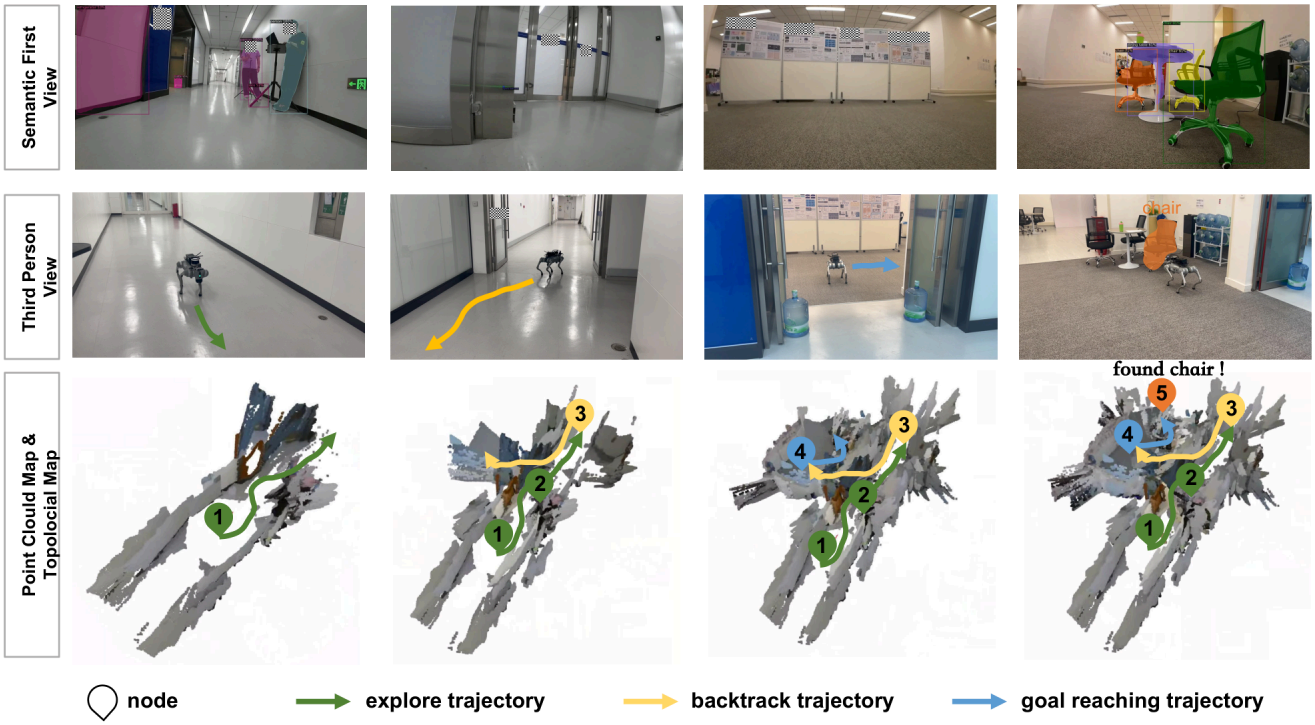

Fig. 2: **Real Robot Experiment Demonstration**: When prolonged hallway exploration is detected, the robot backtracks to a node associated with a room and then starts a procedure inside the room to search for chairs.

poNav's key innovation is modeling environmental topology as actionable spatial memory: critical regions (nodes) and their connectivity (edges) are encoded in a graph structure that evolves in real time as the agent navigates. We design this topological memory graph to be tightly integrated with LLM-based instruction parsing and visual perception modules, enabling the agent to accumulate, update, and retrieve spatial knowledge dynamically—thus supporting coherent reasoning about long-range navigation goals and correcting deviations from optimal paths. Our main contributions are threefold and summarized below:

1) We establish the theoretical connection between topological structures and spatial memory in ObjectNav, proposing a novel paradigm for enhancing navigation robustness.
2) We introduce **TopoNav**, a novel framework that implements dynamic topological memory graphs to bridge transient visual inputs and persistent spatial understanding.
3) We demonstrate that TopoNav achieves state-of-the-art performance on benchmark ObjectNav datasets, outperforming existing methods in success rate and path efficiency, with its performance further validated through extensive real-world experiments.

II. Related Work

A. Object Navigation

Existing approaches to object navigation can be broadly categorized into two paradigms: the first relies on learning-based methods, including reinforcement learning [1], [2], imitation learning [3], [4], and predictive models generating bird's-eye view maps [5], [6], [7], [8]. Although the second employs zero-shot methods that leverage open-vocabulary scene understanding, eliminating the need for task-specific training—these zero-shot approaches are further divided into image-based methods projecting target objects into visual embedding spaces for matching [9], [10], and map-based methods utilizing frontier-based exploration combined with LLMs for semantic reasoning and decision-making [11], [12], [13], [14], [15], [16]. However, current approaches lack explicit spatial memory mechanisms, leading to fragmented reasoning in complex scenarios. These limitations highlight the need for architectures that integrate LLMs with enduring spatial representations, such as topological memory graphs, to support coherent long-term planning.

B. Memory Mechanisms in Navigation

To address the memory limitations of LLM-based Vision Language Navigation (VLN), recent research has explored hierarchical memory systems and continual learning paradigms. The Dual Memory Networks propose a hybrid architecture combining static memory and dynamic memory [17]. This approach mitigates catastrophic forgetting but remains constrained by predefined memory structures, failing to encode dynamic spatial relationships.

In the realm of continual learning, the CVLN paradigm introduces a sequential training framework for VLN agents, enabling them to adapt to new environments while retaining knowledge of previously learned scenes [18]. However, CVLN focuses on incremental task adaptation rather than structural memory consolidation, making it ill-suited for environments requiring persistent topological understanding. Meanwhile, Mem4Nav presents a hierarchical spatial-cognition system that fuses sparse octrees with semantic topology graphs, achieving gains in task completion rates

Authorized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 01,2026 at 02:41:19 UTC from IEEE Xplore. Restrictions apply.

on large-scale urban datasets [19]. This work underscores the importance of multiscale memory encoding but relies on predefined topological priors, limiting its applicability to unseen environments. A critical gap persists in current memory mechanisms: most methods store data-level observations rather than structure-level relationships. TopoNav addresses this by modeling topology as actionable spatial memory, dynamically updating graph nodes and edges to reflect real-time environmental changes [20].

C. Topological Structures for Spatial Navigation

Topological maps have gained renewed attention in VLN, particularly for encoding environmental connectivity and long-term spatial relationships. The ETPNav framework introduces an online topological planning system that self-organizes waypoints into dynamic graphs [21]. However, ETPNav's topological maps are constructed based on pre-computed waypoint graphs, lacking real-time adaptability to unseen environmental changes [21]. Similarly, Revind combines offline reinforcement learning with topological graphs to enable long-horizon navigation in real-world scenarios [22], but its reliance on predefined graph structures limits generalization to novel environments.

Recent studies have also explored GNN-based topological reasoning to enhance spatial understanding. For example, Su et al. analyze the topology awareness of GNNs and demonstrate their ability to preserve structural information in graph representations [23]. However, these methods primarily focus on node classification tasks and have not been adapted for VLN's multimodal reasoning requirements. In contrast, TopoNav integrates topological graphs with LLM-driven instruction parsing, enabling semantic-aware topological updates that dynamically align with linguistic cues [20].

Crucially, TopoNav distinguishes itself from other frameworks by treating topological structures as evolving memory carriers rather than static maps. Unlike prior work that uses topology for coarse path planning [21], [22], TopoNav's dynamic topological memory graph continuously refines spatial knowledge through visual-linguistic interactions, supporting adaptive deviation correction and long-range goal reasoning in real time [20].

III. Methodology

A. Problem Formulation

The ObjectNav task [24] requires autonomous agents to locate and approach predefined target object categories in previously unexplored environments. Formally, given a set of target object categories $\mathcal{T} = \{T_1, T_2, \ldots, T_N\}$ (e.g., "chair", "bed", etc,.), an episode initiates by instantiating an agent at a randomized starting pose within an unseen scene $\mathcal{S}$. The agent is assigned a specific target category $T_i \in \mathcal{T}$ as its navigation objective. At each discrete timestep t, the agent perceives an observation tuple $O_t = (V_t, P_t)$, where V_t represents egocentric visual inputs comprising RGB and depth images, while P_t denotes the agent's proprioceptive pose information (position and orientation). Based on the sensory inputs, the agent must navigate to a specified object T_i in the unknown environment $\mathcal{S}$.

B. Overview

Figure 3 shows our framework. At each timestep, the system receives an RGB-D image V_t and agent pose P_t. We build a semantic point cloud map as in Section III-C and maintain a text-based topological memory map as in Section III-D. A fused policy then combines both maps with Vision Language Model (VLM) guidance to select waypoint candidates, following the strategy in Section III-E.

C. Semantic Point Cloud Map Construction

The mapping module constructs a point-cloud representation using visual observations V_t and pose information P_t from Section III-A. It contains five components: a *scene point cloud* $\mathcal{M}_{scene}$, a semantically segregated *object point clouds* $\mathcal{M}_{obj}$, a filtered *navigable point cloud* $\mathcal{M}_{nav}$, an *obstacle point cloud* $\mathcal{M}_{obs}$, and a *frontier point cloud* $\mathcal{M}_{fro}$. The map is updated incrementally to preserve geometric and semantic consistency across observations.

At each timestep t, the system processes RGB images $I_t \in \mathbb{R}^{H\times W\times 3}$ and depth maps $D_t \in \mathbb{R}^{H\times W}$ from the observation V_t. The agent's pose $P_t = (\mathbf{p}_t, \mathbf{q}_t)$ provides position $\mathbf{p}_t \in \mathbb{R}^3$ and orientation $\mathbf{q}_t$ (represented as quaternion). Using the known camera intrinsics matrix $\mathbf{K} \in \mathbb{R}^{3\times 3}$, each valid depth pixel (u, v) with depth value $d = D_t(u, v)$ is transformed to camera coordinates:

$$\mathbf{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = d \cdot \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \tag{1}$$

The corresponding RGB value $\mathbf{c} = I_t(u, v)$ is associated with each generated 3D point. These camera-space points $\mathbf{x}_c$ are then transformed to the global coordinate frame via rigid body transformation:

$$\mathbf{x}_w = \mathbf{R}(\mathbf{q}_t)\mathbf{x}_c + \mathbf{p}_t, \tag{2}$$

where $\mathbf{R}(\mathbf{q}_t) \in \mathbb{R}^{3\times 3}$ is the rotation matrix derived from the orientation quaternion $\mathbf{q}_t$.

The scene point cloud $\mathcal{M}_{scene}$ aggregates all observed points $\mathbf{x}_w$ with their associated color attributes $\mathbf{c}$. To reduce measurement noise, a voxel downsampling operation is applied after incorporating new points:

$$\mathcal{M}_{scene} \leftarrow \text{VoxelDownsample}\left(\mathcal{M}_{scene} \cup \{\mathbf{x}_w^{(i)}\}, r_{pcd}\right),$$

where r_{pcd} specifies the voxel grid resolution. This operation replaces points within each cubic voxel of size r_{pcd} with their centroid, ensuring uniform point density while preserving geometric features.

Semantic segmentation is applied to the RGB image I_t to identify regions corresponding to interesting object categories in $\mathcal{T}$. The resulting object masks M_{obj} select depth pixels belonging to recognized objects. For each object class $c \in \mathcal{T}$, the corresponding points in the depth map D_t are projected and transformed using the same process outlined in Equations 1 and 2. These class-specific points are accumulated in separate object point clouds:

Authorized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 01,2026 at 02:41:19 UTC from IEEE Xplore. Restrictions apply.

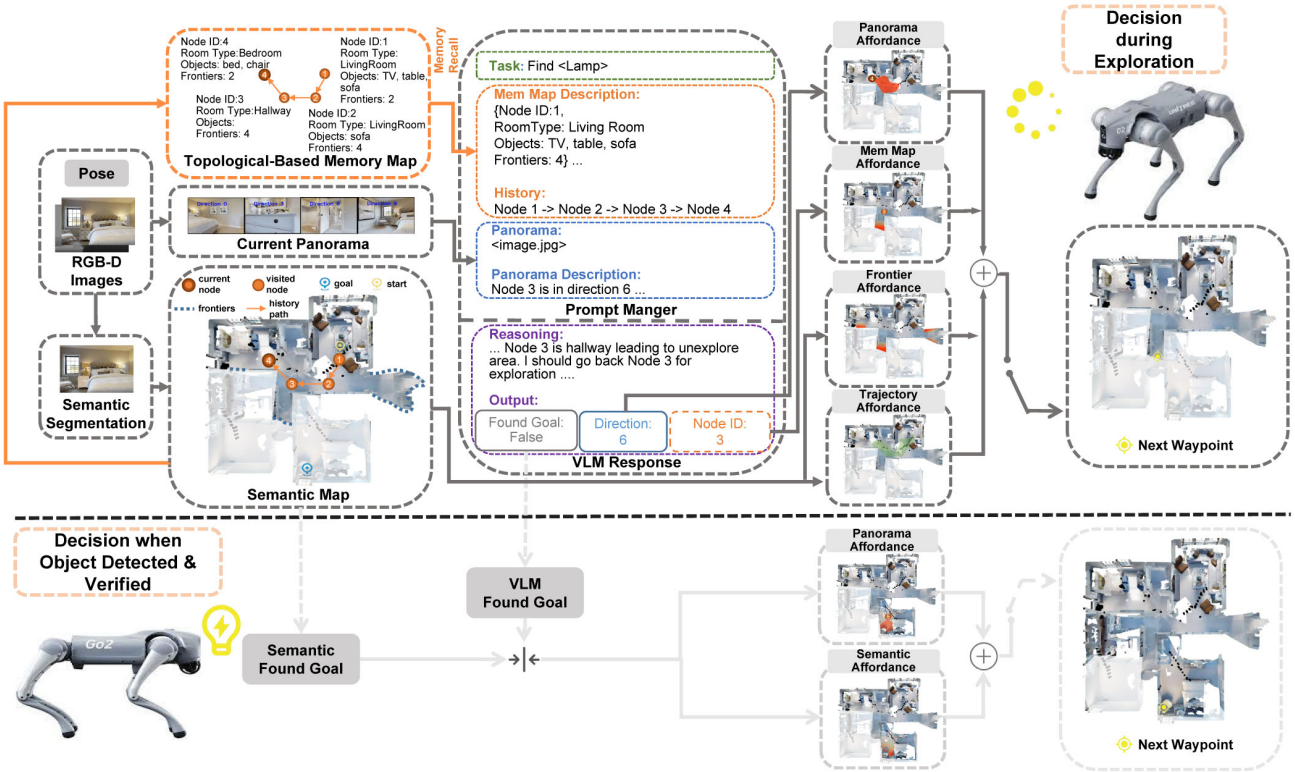

Fig. 3: **Framework Overview**. Our approach builds a semantic point cloud map alongside a topological memory map during navigation. A prompt manager integrates current observations with these representations, while an affordance-driven waypoint selection strategy dynamically balances exploration and target acquisition

$$\mathcal{M}_{obj}^{(c)} \leftarrow \text{VoxelDownsample}\left(\mathcal{M}_{obj}^{(c)} \cup \{\mathbf{x}_w^{(c,j)}\}, r_{pcd}\right),$$

where $\mathbf{x}_w^{(c,j)}$ denotes the j-th point belonging to object class c. These class-specific representations enable efficient spatial querying of potential target objects T_i during navigation.

The navigable point cloud $\mathcal{M}_{nav}$ is constructed through a sequential geometric and semantic refinement process to identify traversable regions. Initial candidate points are extracted based on height constraints relative to the estimated floor level, formalized as:

$$\mathcal{M}_{nav} = \{\mathbf{x}_w \in \mathcal{M}_{scene} : z_{floor} - \delta \leq z_w \leq z_{floor} + \delta\},$$

where z_{floor} represents the ground plane elevation, and δ accommodates floor irregularities. This geometrically filtered set is then augmented with semantically navigable structures: Points belonging to object classes in $\mathcal{C}_{nav} \subset \mathcal{T}$ (such as stairs) are incorporated regardless of height, yielding:

$$\mathcal{M}_{nav} \leftarrow \mathcal{M}_{nav} \cup \left(\bigcup_{c \in \mathcal{C}_{nav}} \mathcal{M}_{obj}^{(c)}\right).$$

To ensure spatial connectivity in sparse observation conditions, linear interpolation bridges gaps between the agent's standing position $\mathbf{p}_{stand} = [x_t, y_t, z_{floor}]$ and observed navigable points. For each $\mathbf{x}_i \in \mathcal{M}_{nav}$ with $\|\mathbf{x}_i - \mathbf{p}_{stand}\|_2 > \Delta_{step}$, intermediate points are generated:

$$\mathbf{p}_k = \mathbf{p}_{stand} + \frac{k \cdot \Delta_{step}}{\|\mathbf{x}_i - \mathbf{p}_{stand}\|_2}(\mathbf{x}_i - \mathbf{p}_{stand})$$
$$\text{for} \quad k = 1, 2, \ldots, \lfloor \|\mathbf{x}_i - \mathbf{p}_{stand}\|_2 / \Delta_{step} \rfloor,$$

retaining only those within the navigable height band $|z_k - z_{floor}| < \delta$. This interpolation establishes continuous traversable surfaces even with partial observations. We then downsample to uniform resolution:

$$\mathcal{M}_{nav} \leftarrow \text{VoxelDownSample}(\mathcal{M}_{nav}, r_{pcd}).$$

The obstacle point cloud $\mathcal{M}_{obs}$ is constructed by selecting all points located above the floor elevation. Formally:

$$\mathcal{M}_{obs} = \{\mathbf{x}_w \in \mathcal{M}_{scene} : z_w > z_{floor} + \delta\}.$$

The frontier point cloud $\mathcal{M}_{fro}$ identifies boundary regions between explored navigable areas and unknown space. This representation is constructed through geometric analysis of the navigable and obstacle point clouds within a discretized 2D grid with width N_x and height N_y, where

$$N_x = \left\lceil \frac{x_{max} - x_{min}}{r_g} \right\rceil, \quad N_y = \left\lceil \frac{y_{max} - y_{min}}{r_g} \right\rceil.$$

Each grid cell $\mathcal{G}(i,j)$ is populated by projecting points from $\mathcal{M}_{nav}$ and $\mathcal{M}_{obs}$:

$$\mathcal{G}(i,j) = \begin{cases} 1 & \text{if } \exists \mathbf{p} \in \mathcal{M}_{nav} : \phi(\mathbf{p}) = (i,j) \\ -1 & \text{if } \exists \mathbf{p} \in \mathcal{M}_{obs} : \phi(\mathbf{p}) = (i,j) \\ 0 & \text{otherwise,} \end{cases}$$

where $\phi(\mathbf{p}) = \left(\left\lfloor \frac{p_x - x_{min}}{r_g} \right\rfloor, \left\lfloor \frac{p_y - y_{min}}{r_g} \right\rfloor\right)$ with r_g represents the grid resolution.

Then we select frontier candidates points $\mathcal{F}_{candidate}$ using [25]. Finally, we project 2D grid to 3D at floor height with

Authorized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 01,2026 at 02:41:19 UTC from IEEE Xplore. Restrictions apply.

$$\mathbf{x}_{fro} = \begin{bmatrix} i_k \cdot r_g + x_{min} \\ j_k \cdot r_g + y_{min} \\ z_{floor} \end{bmatrix} \quad \forall (i_k, j_k) \in \mathcal{F}_{candidate}.$$

The resulting point cloud $\mathcal{M}_{fro}$ provides target locations for exploration while maintaining safe clearance from obstacles.

D. Topological Memory Map Construction

Complementing the geometric representation defined in Section III-C, we maintain a hierarchical topological map $\mathcal{M}_{topo} = \{\mathcal{N}_k\}_{k=1}^{K}$ as shown in Fig. 4. This abstraction transforms continuous environmental geometry into semantically meaningful nodes, enabling high-level reasoning essential for long-horizon navigation tasks. Each topological node $\mathcal{N}_k = \langle \text{id}_k, \mathbf{n}_k, \mathcal{O}_k, R_k, f_k \rangle$ contains five navigation-critical attributes:

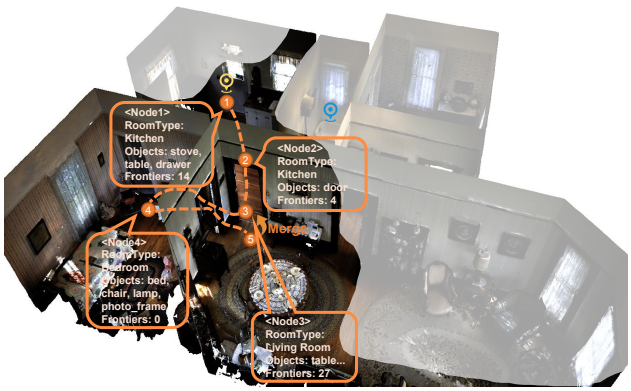

Fig. 4: The text-based topological map enables VLMs to perform high-level reasoning and memory-based decisions.

1. Node ID: a unique ID id_k to distinguish nodes.
2. Position: $\mathbf{n}_k = (x_k, y_k)$ in world coordinates (z-coordinate omitted as topological reasoning operates primarily in 2D space).
3. Surrounding object classes: $\mathcal{O}_k \subset \mathcal{T}$ identified within a r_{topo} radius of $\mathbf{n}_k$. This semantic context informs object search strategies during target-driven navigation.
4. Room type: R_k derived from VLM analysis of panoramic RGB observations $V_t^{panorama}$. This high-level semantic classification enables room-aware navigation policies.
5. Frontier count: $f_k = |\mathcal{F}_{\mathbf{n}_k}|$ where $\mathcal{F}_{\mathbf{n}_k} = \{\mathbf{x}_w \in \mathcal{M}_{fro} : \|\mathbf{x}_w - \mathbf{n}_k\|_2 < r_{topo}\}$.

The topological map initializes at episode start with $\mathcal{N}_1$ at the agent's starting position $\mathbf{p}_0$. New nodes are incrementally created when the agent arrive a new waypoint (determined by the strategy in Section III-E).

Node creation involves:

$$\mathcal{N}_{new} = \langle \text{id}_k, \mathbf{p}_k, \mathcal{O}_k, R_k, f_k \rangle$$

where:

$$\mathcal{O}_k = \left\{ c \in \mathcal{T} : \exists \mathbf{x}_w \in \mathcal{M}_{obj}^{(c)} \text{ s.t. } \|\mathbf{x}_w - \mathbf{n}_k\|_2 < r_{topo} \right\}$$
$$R_k = \Psi_{\text{VLM}}(V_t^{panorama})$$
$$f_k = |\{\mathbf{x}_w \in \mathcal{M}_{fro} : \|\mathbf{x}_w - \mathbf{n}_k\|_2 < r_{topo}\}|$$

with Ψ_{VLM} classifying room types (e.g., "kitchen", "bedroom") from panoramic imagery.

Node merging occurs when spatial proximity and traversability conditions are satisfied, preventing redundant representation of connected spaces:

$$\|\mathbf{n}_i - \mathbf{n}_j\|_2 < d_{merge} \wedge$$
$$\nexists \mathbf{x}_w \in \mathcal{M}_{obs} \cap (\mathbf{x}_w \in \mathcal{L}(\mathbf{n}_i, \mathbf{n}_j)),$$

where $\mathcal{L}(\mathbf{n}_i, \mathbf{n}_j)$ denotes the straight-line path between nodes, and d_{merge} represent merge distance. Merged nodes retain the earliest node ID and room type R for temporal consistency, with updated attributes:

$$\mathbf{n}' = \frac{\mathbf{n}_i + \mathbf{n}_j}{2}$$
$$\mathcal{O}'_t = \left\{ c \in \mathcal{T} : \exists \mathbf{x}_w \in \mathcal{M}_{obj}^{(c)} \text{ s.t. } \|\mathbf{x}_w - \mathbf{n}'\|_2 < r_{topo} \right\}$$
$$f'_k = |\{\mathbf{x}_w \in \mathcal{M}_{fro} : \|\mathbf{x}_w - \mathbf{n}\|_2 < r_{topo}\}|.$$

The merged node position is the centroid of the two merging nodes, with surrounding objects $\mathcal{O}'$ and frontier count f' recomputed from $\mathcal{M}_{obj}$ and $\mathcal{M}_{fro}$ respectively.

The topological map critically enhances navigation through three integrated capabilities: *Semantic Goal Prioritization* leverages room-object correlations to guide target search (e.g., locating beds in bedrooms by aligning T_i with R_k); *Exploration Optimization* directs agents toward frontier-rich nodes to minimize redundant coverage; and *History-Aware Planning* utilizes the navigation history sequence $\mathcal{H} = (\mathcal{N}_1 \rightarrow \mathcal{N}_2 \rightarrow \cdots)$ to avoid recent revisits while permitting strategic returns to key nodes like hallways.

This hierarchical representation serves as a cognitive map bridging perception and action, enabling: efficient long-horizon planning, context-aware navigation through room-object associations, adaptive exploration based on frontier distributions, and memory-efficient environment modeling.

E. Waypoint Selection Strategy

The waypoint selection strategy integrates VLM guidance to determine optimal navigation targets. The VLM processes multimodal inputs including: 1) Target object category T_{target}, 2) Textual topological memory map representation $\mathcal{M}_{topo}$, 3) Navigation history $\mathcal{H}$, and 4) Current panoramic observation $V_t^{panorama}$. Based on these inputs, the VLM outputs three critical navigation parameters: 1) Next topological node ID k', 2) Preferred direction d_{vlm} within the panoramic view $V_t^{panorama}$, and 3) Binary target object detection indicator $g_{found} \in \{0, 1\}$.

The VLM helps waypoint selection through an affordance-based framework operating over the navigable point cloud $\mathcal{M}_{nav}$. For each candidate point $\mathbf{p}_{nav}^i \in \mathcal{M}_{nav}$, we compute a composite affordance value $A(\mathbf{p}_{nav}^i)$ that integrates multiple navigation objectives. The strategy operates in two distinct phases based on target detection status.

The navigation affordances are computed using a unified framework. For any point $\mathbf{p}_i \in \mathcal{M}_{nav}$ and target point cloud S, we define:

$$d_i^S = \min_{\mathbf{q} \in S} \|\mathbf{p}_i - \mathbf{q}\|_2, \quad d_{\min}^S = \min_i d_i^S, \quad d_{\max}^S = \max_i d_i^S.$$

The normalized affordance is:

$$N(d_i^S) = 1 - \frac{d_i^S - d_{\min}^S}{d_{\max}^S - d_{\min}^S + \epsilon} \quad .$$

Authorized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 01,2026 at 02:41:19 UTC from IEEE Xplore. Restrictions apply.

This normalization ensures that the affordance values range from 0 and 1, where higher values indicate more favorable positions.

1) Exploration Phase

When $g_{found} = 0$ (target not visually confirmed by VLM) and $T_{target} \notin \mathcal{O}$ (target object not in detected object list), the agent enters exploration mode. This phase employs four complementary affordance components to guide efficient exploration:

VLM Direction Affordance (A_{dir}): Incorporates the VLM's directional guidance by attracting the agent toward points aligned with the preferred direction:

$$A_{dir}(\mathbf{p}^i_{nav}) = N\left(d_i^{S_{dir}}\right),$$

where S_{dir} is the point set along the VLM-selected direction d_{vlm} in the panoramic view. This component leverages the VLM's visual reasoning capability to identify promising directions that potentially lead to the target object.

Node Affordance (A_{node}): Focuses exploration around frontiers near the VLM-selected topological node:

$$A_{node}(\mathbf{p}^i_{nav}) = N\left(d_i^{S_{node}}\right),$$

with S_{node} representing frontier points associated with the VLM-selected node $\mathcal{N}_{k'}$. This affordance utilizes the VLM's topological reasoning ability to prioritize high-potential areas while avoiding repeated visits to the same regions.

Frontier Affordance (A_{front}): Encourages comprehensive exploration of unmapped regions:

$$A_{front}(\mathbf{p}^i_{nav}) = N\left(d_i^{\mathcal{M}_{fro}}\right).$$

This component provides broad exploration pressure complementary to the more targeted VLM guidance.

History Avoidance Affordance (A_{hist}): Prevents redundant revisits to recently explored areas:

$$A_{hist}(\mathbf{p}^i_{nav}) = 1 - N\left(d_i^{S_{hist}}\right),$$

where S_{hist} contains trajectory history points. This mechanism promotes efficient exploration by encouraging novelty-seeking behavior.

2) Target Acquisition Phase

When $g_{found} = 1$ and target objects are confirmed by the detection model ($T_{target} \in \mathcal{O}$), the strategy transitions to target approach mode. This phase employs two key affordances:

VLM Direction Affordance (A_{dir}): Maintains continuity with exploration by retaining the VLM's directional guidance, ensuring consistent navigation toward the target area.

Semantic Proximity Affordance (A_{sem}): Directs the agent toward visually confirmed target instances:

$$A_{sem}(\mathbf{p}^i_{nav}) = N\left(d_i^{S_{sem}}\right),$$

where S_{sem} is the set of points classified as the target object T_{target} in $\mathcal{M}_{obj}$. This affordance enables precise approach to identified targets.

3) Safety Integration and Waypoint Selection

The composite affordance combines phase-specific components through summation:

$$A(\mathbf{p}^i_{nav}) = \begin{cases} \sum A_{\{dir,node,front,hist\}} & \text{exploration} \\ A_{dir} + A_{sem} & \text{target acquisition,} \end{cases}$$

Obstacle avoidance is implemented by zeroing out affordances near obstacles:

$$A_{obs}(\mathbf{p}^i_{nav}) = N\left(d_i^{\mathcal{M}_{obs}}\right)$$

$$A(\mathbf{p}^i_{nav}) = A(\mathbf{p}^i_{nav}) \cdot \mathbb{I}(A_{obs}(\mathbf{p}^i_{nav}) > \sigma),$$

where σ is a safety threshold parameter determining the minimum safe distance from obstacles. This operation eliminates navigation options that are too close to potential hazards.

The target waypoint is selected through global maximization over the navigable space:

$$\mathbf{p}_{target} = \underset{\mathbf{p}^i \in \mathcal{M}_{nav}}{\arg\max}\, A(\mathbf{p}^i_{nav}).$$

This optimization ensures the agent moves toward positions that best balance exploration efficiency, target approach, and safety considerations.

Method	Training Free	HM3D			MP3D		
		SR↑	SPL↑	DTG↓	SR↑	SPL↑	DTG↓
SemExp[6]	✗	-	-	-	0.144	0.360	6.73
ZSON[9]	✗	0.255	0.126	-	0.153	0.048	-
PONI[7]	✗	-	-	-	0.318	0.121	5.1
PixNav[3]	✗	0.379	0.205	-	-	-	-
SPNet[26]	✗	0.312	0.101	-	0.163	0.048	-
SGM[27]	✗	**0.602**	0.308	-	0.377	0.147	4.93
CoW[10]	✓	-	-	-	0.074	0.037	-
ESC[28]	✓	0.392	0.223	-	0.287	0.142	-
VLFM[12]	✓	0.525	0.304	-	0.364	**0.175**	-
VoroNav[29]	✓	0.420	0.260	-	-	-	-
L3MVN[11]	✓	0.504	0.231	4.43	-	-	-
TriHelper[13]	✓	0.565	0.253	3.87	-	-	-
InstructNav[15]	✓	0.510	0.187	<u>2.89</u>	0.420	0.161	<u>4.26</u>
GAMap[30]	✓	0.531	0.260	-	-	-	-
UniGoal[31]	✓	0.545	0.251	-	0.410	0.164	-
WMNav[32]	✓	0.581	<u>0.312</u>	-	<u>0.454</u>	<u>0.172</u>	-
Ours	✓	<u>0.601</u>	**0.346**	**2.49**	**0.455**	0.168	**4.21**

TABLE I: Comparison with methods on HM3D and MP3D object goal navigation. We use **bold** and <u>underline</u> to denote the first and second best performance respectively.

IV. Experiments

A. Experiment Setup

We evaluate our method in the Habitat Simulator [33] using HM3D and MP3D datasets. HM3D [34] comprises 2000 episodes across 20 scenes with 6 target objects. MP3D [35] contains 2195 episodes spanning 11 scenes and 21 target objects. Our point cloud map construction builds upon the codebase of InstructNav [15]. We employ GPT-4o [36] as the VLM and GLEE [37] as the vision foundation model for semantic segmentation. Although the simulation environment has limited target objects, GLEE's open-vocabulary capability enables generalization to open-vocabulary object navigation tasks. Experiments were conducted on a server with an RTX3090 GPU to support efficient detection model inference and parallel point cloud processing.

Authorized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 01,2026 at 02:41:19 UTC from IEEE Xplore. Restrictions apply.

B. Metrics

We adopt three established metrics [38]: Success Rate (SR), Success weighted by Path Length (SPL), and Distance to Goal (DTG). Higher SR and SPL values indicate better performance. SPL combines task completion and path efficiency. DTG measures the final distance between agent and target at episode termination, where lower values are desirable.

C. Evaluation

Table I reports comparative results in Habitat. Our method achieves state-of-the-art (SOTA) performance for training-free ObjectNav on both datasets. On HM3D, our 0.601 SR exceeds the best training-free baseline WMNav by +2.0%, while SPL improves by +10.9% over VLFM. On MP3D, our SR surpasses WMNav by +0.2%, and our 4.21 DTG sets a new benchmark. This training-free approach also matches or exceeds leading training-based methods: our HM3D 0.601 SR is close to SGM's training-based SOTA (0.602), while on MP3D we outperform SGM's 0.377 SR by +20.7%. These results show that our method narrows the gap between training-free and training-based paradigms. As shown in Fig. 5, TopoNav consistently outperforms InstructNav across all navigation targets, including bed, chair, and potted plant.

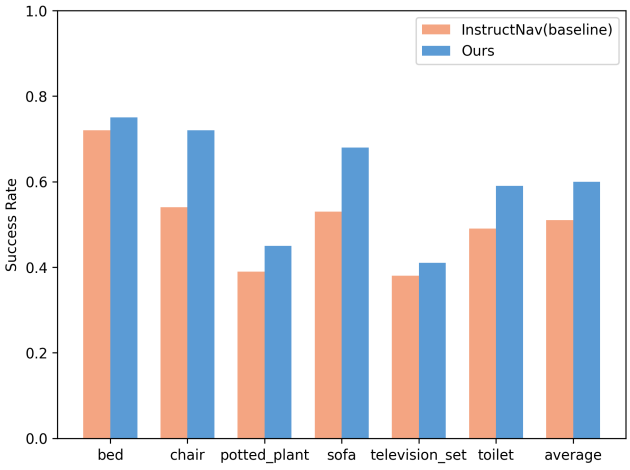

Fig. 5: Comparison with baseline (InstructNav[15]) on various objects.

D. Ablation Study

VLM and GLEE. Both VLM and GLEE are quite important for better performance. The combination of VLM + GLEE demonstrates a higher SR and SPL compared to the individual methods only with VLM and only with GLEE (Table. II). This suggests that the integration of these two components improves the overall success and efficiency of goal detection in terms of these metrics. Moreover, a smaller DTG of the VLM + GLEE method indicates that the combination also improves the spatial performance, enabling the robot to stop closer to the goal.

Overall, the results consistently validate that the integration of VLM and GLEE yields a more robust and efficient goal detection framework for object navigation tasks, outperforming the use of either component in isolation in all evaluated metrics. These results highlight the complex interactions between different components in detection of targests for object navigation.

Method	Object Nav. on HM3D		
	SR↑	SPL↑	DTG↓
VLM + GLEE	0.601	0.346	2.49
VLM only	0.472	0.233	3.58
GLEE only	0.445	0.223	3.39

TABLE II: Goal detection ablation.

Node Information. We also highlight the importance of topological node information. As shown in Table III, comparing with the topological node with all information, omitting any of these topological map node information types (Frontiers, Room Type, Object Type) causes a decline in SR and SPL, and an increase in DTG. This ablation study underscores the importance of topological map node information for robust object navigation.

Method	Object Nav. on HM3D		
	SR↑	SPL↑	DTG↓
Current Memory	0.601	0.346	2.49
w/o Frontiers	0.544	0.267	2.70
w/o Room Type	0.510	0.236	3.12
w/o Objects Type	0.503	0.243	2.94

TABLE III: Topological map node information ablation.

E. Real Robot Implementation

We conduct real-robot experiments on a Unitree Go2 quadruped, with the full TopoNav system running on a Jetson Orin. We use an Intel Realsense D435i for point-cloud construction and a HESAI XT-6 lidar for SLAM. For semantic segmentation, we use GLEE [37], a SOTA detection and segmentation model, and employ GPT-4o [36] to maintain topological nodes and determine navigation directions. To build the semantic point cloud, we capture 9 RGBD panorama observations in real-world experiments (12 in simulation). As shown in Fig. 2, Go2 explores a hallway, backtracks to the correct node, and enters a room to find the target object. These experiments show that TopoNav supports adaptive exploration by adjusting paths in real time while building topological nodes as memory anchors. Balancing revisits to known regions with exploration of new areas reduces redundancy, improves coverage, and enhances navigation efficiency.

V. Conclusion

In this work, we present TopoNav, a zero-shot ObjectNav framework powered by topological mapping. By integrating a semantic point cloud with a graph-structured memory map, the system dynamically builds and updates spatial representations in real time, without prior training or scene-specific tuning. Nodes represent semantic regions and edges encode navigational connectivity, enabling efficient knowledge accumulation, path correction, and robust decision-making in novel environments. TopoNav achieves convincing naviga-

Authorized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 01,2026 at 02:41:19 UTC from IEEE Xplore. Restrictions apply.

tion performance on benchmark datasets and demonstrates strong practicality in real-world experiments. For future work, we plan to incorporate memory mechanisms into end-to-end methods, which have shown superior performance to zero-shot approaches on the VLN task.

REFERENCES

[1] X. Ye and Y. Yang, "Efficient robotic object search via hiem: Hierarchical policy learning with intrinsic-extrinsic modeling," *IEEE robotics and automation letters*, vol. 6, no. 3, pp. 4425–4432, 2021.

[2] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, K. Ehsani, J. Salvador, W. Han, E. Kolve, A. Kembhavi, and R. Mottaghi, "Procthor: Large-scale embodied ai using procedural generation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5982–5994, 2022.

[3] W. Cai, S. Huang, G. Cheng, Y. Long, P. Gao, C. Sun, and H. Dong, "Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5228–5234, 2024.

[4] Y. Wang, Y. Fang, T. Wang, Y. Feng, Y. Tan, S. Zhang, P. Liu, Y. Ji, and R. Xu, "Dreamnav: A trajectory-based imaginative framework for zero-shot vision-and-language navigation," in *IEEE International Conference on Robotics and Automation*, 2026.

[5] H. Luo, A. Yue, Z.-W. Hong, and P. Agrawal, "Stubborn: A strong baseline for indoor object navigation," in *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3287–3293.

[6] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4247–4258, 2020.

[7] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "Poni: Potential functions for objectgoal navigation with interaction-free learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18890–18900, 2022.

[8] A. J. Zhai and S. Wang, "Peanut: Predicting and navigating to unseen targets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10926–10935, 2023.

[9] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, "Zson: Zero-shot object-goal navigation using multimodal goal embeddings," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32340–32352, 2022.

[10] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. 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*, pp. 23171–23181, 2023.

[11] B. Yu, H. Kasaei, and M. Cao, "L3mvn: Leveraging large language models for visual target navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3554–3560, 2023.

[12] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *IEEE International Conf. on Robotics and Automation*, pp. 42–48, 2024.

[13] L. Zhang, Q. Zhang, H. Wang, E. Xiao, Z. Jiang, H. Chen, and R. Xu, "Trihelper: Zero-shot object navigation with dynamic assistance," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10035–10042, 2024.

[14] L. Zhang, H. Wang, E. Xiao, X. Zhang, Q. Zhang, Z. Jiang, and R. Xu, "Multi-floor zero-shot object navigation policy," in *IEEE International Conference on Robotics and Automation*, pp. 6416–6422, 2025.

[15] Y. Long, W. Cai, H. Wang, G. Zhan, and H. Dong, "Instructnav: Zero-shot system for generic instruction navigation in unexplored environment," in *Conference on Robot Learning*, pp. 2049–2060, 2025.

[16] M. Zhang, Y. Du, C. Wu, J. Zhou, Z. Qi, J. Ma, and B. Zhou, "Apexnav: An adaptive exploration strategy for zero-shot object navigation with target-centric semantic fusion," *IEEE Robotics and Automation Letters*, vol. 10, no. 11, pp. 11530 – 11537, 2025.

[17] Y. Zhang, W. Zhu, H. Tang, Z. Ma, K. Zhou, and L. Zhang, "Dual memory networks: A versatile adaptation approach for vision-language models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 28718–28728, 2024.

[18] M. Jeong, Z. Li, H. Wang, Z. Li, and Y. Lv, "Continual vision-and-language navigation," in *36th British Machine Vision Conference*, 2025.

[19] J. Liu, T. Liu, J. Yu, R. Fu, Y. Pan, and Y. Zhang, "Mem4nav: Boosting vision-and-language navigation with a hierarchical spatial-cognition long-short memory system," *arXiv preprint arXiv:2506.19433*, 2025.

[20] J. Wei, B. Liu, Y. Wu, and Z. Liu, "Exploring spatial representation to enhance llm reasoning in aerial vision-and-language navigation," *arXiv preprint arXiv:2410.08500*, 2024.

[21] D. An, H. Wang, W. Wang, Z. Wang, Y. Huang, K. He, and L. Wang, "Etpnav: Evolving topological planning for vision-language navigation in continuous environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[22] D. Shah, A. Bhorkar, H. Leen, I. Kostrikov, N. Rhinehart, and S. Levine, "Offline reinforcement learning for visual navigation," in *Conference on Robot Learning (CoRL)*, 2022.

[23] J. Su and C. Wu, "On the topology awareness and generalization performance of graph neural networks," in *European Conference on Computer Vision*, pp. 73–89, Springer, 2024.

[24] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," *arXiv preprint arXiv:2006.13171*, 2020.

[25] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 146–151, 1997.

[26] Q. Zhao, L. Zhang, B. He, and Z. Liu, "Semantic policy network for zero-shot object goal visual navigation," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7655–7662, 2023.

[27] S. Zhang, X. Yu, X. Song, X. Wang, and S. Jiang, "Imagine before go: Self-supervised generative map for object goal navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16414–16425, 2024.

[28] K. Zhou, K. Zheng, C. Pryor, Y. Shen, H. Jin, L. Getoor, and X. E. Wang, "Esc: Exploration with soft commonsense constraints for zero-shot object navigation," in *International Conference on Machine Learning*, pp. 42829–42842, PMLR, 2023.

[29] P. Wu, Y. Mu, B. Wu, Y. Hou, J. Ma, S. Zhang, and C. Liu, "Voronav: Voronoi-based zero-shot object navigation with large language model," in *41st International Conference on Machine Learning*, 2024.

[30] H. Huang, Y. Hao, C. Wen, A. Tzes, Y. Fang, *et al.*, "Gamap: Zero-shot object goal navigation with multi-scale geometric-affordance guidance," *Advances in Neural Information Processing Systems*, vol. 37, pp. 39386–39408, 2024.

[31] H. Yin, X. Xu, L. Zhao, Z. Wang, J. Zhou, and J. Lu, "Unigoal: Towards universal zero-shot goal-oriented navigation," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19057–19066, 2025.

[32] D. Nie, X. Guo, Y. Duan, R. Zhang, and L. Chen, "Wmnav: Integrating vision-language models into world models for object goal navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2392–2399, 2025.

[33] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra, "Habitat 2.0: Training home assistants to rearrange their habitat," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[34] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. M. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, and D. Batra, "Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[35] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *International Conf. on 3D Vision*, 2017.

[36] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[37] J. Wu, Y. Jiang, Q. Liu, Z. Yuan, X. Bai, and S. Bai, "General object foundation model for images and videos at scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3783–3795, 2024.

[38] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.

Authorized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 01,2026 at 02:41:19 UTC from IEEE Xplore. Restrictions apply.