Semantic-Aware Path Planning with SLAM-Based Simulation for Vision-Language Navigation

Usama Mohiuddin
Department of Computer Science
Wilfrid Laurier University
Waterloo, Canada
mohi0340@mylaurier.ca

***Abstract*—Vision-Language Navigation (VLN) focuses on enabling autonomous agents to follow natural language instructions in complex indoor environments. This paper presents a study of a semantic-aware path planning approach for VLN that emphasizes clarity and system-level understanding. The method incorporates semantic constraints into a graph-based planning pipeline and evaluates navigation behavior using a SLAM-inspired simulation framework. A prototype was implemented in Python and tested in simulated indoor environments based on standard VLN datasets. The results show that semantic cost modeling affects route selection and introduces trade-offs between path efficiency and computational cost. Although the evaluation is limited to simulation, the study highlights the importance of classical path planning as a reliable component in hybrid semantic navigation systems and points toward future real-world deployment.**

***Index Terms*—Vision-Language Navigation, Semantic-Aware Path Planning, Graph-Based Navigation, SLAM-Inspired Simulation, Autonomous Navigation**

I. Introduction

Autonomous navigation is a core problem in robotics and artificial intelligence. For a robot or agent to move independently, it must be able to perceive its surroundings, understand where it is, and decide how to reach a goal. In real environments, this task is challenging because scenes are visually complex and navigation instructions are often given in natural language rather than exact coordinates.

Vision-Language Navigation (VLN) addresses this challenge by studying how agents can follow spoken or written instructions while navigating through an environment. Instead of being given precise locations, an agent may receive instructions such as moving to a nearby room or stopping close to a particular object. Successfully completing these tasks requires linking language, visual observations, and navigation decisions in a meaningful and reliable way.

Traditional navigation systems often rely on *geometric path planning*. In this setting, the environment is treated as a geometric space, and the goal is to find the shortest or least-cost path between two locations while avoiding obstacles. While geometric planning is efficient and well understood, it does not account for higher-level meaning in the environment. For example, a purely geometric planner may choose a path that passes through an undesirable or restricted area simply because it is shorter.

In contrast, *semantic path planning* incorporates information about objects, regions, and their meaning into the navigation process. By using semantic knowledge, an agent can prefer paths that align better with human instructions and environmental context, even if those paths are not the shortest in terms of distance. For example, a semantic-aware planner may avoid certain rooms or favor paths that pass through meaningful landmarks described in an instruction. This added reasoning improves interpretability and makes navigation decisions easier to understand.

Recent work in VLN has shown strong performance using large neural networks trained on extensive datasets. Although these approaches perform well on benchmarks, they often require significant computational resources and are difficult to deploy on embedded robotic platforms. In addition, many end-to-end learning-based systems offer limited insight into how navigation decisions are made, which makes debugging and failure analysis challenging.

Semantic information offers a practical way to improve navigation behavior without relying entirely on end-to-end learning. By explicitly modeling semantic constraints and integrating them into classical planning methods, it is possible to balance efficiency, interpretability, and robustness. However, adding semantic reasoning also introduces additional computational cost and design complexity, especially when combined with path planning and localization.

This paper studies a semantic-aware path planning approach for Vision-Language Navigation. Rather than proposing a new end-to-end learning model, the goal is to examine how semantic constraints influence navigation behavior within a structured and interpretable planning pipeline. By focusing on the comparison between geometric and semantic planning under controlled conditions, this work aims to highlight the trade-offs involved in integrating semantic reasoning into navigation systems.

II. Literature Review

Vision-Language Navigation (VLN) became a well-defined research problem with the introduction of benchmark datasets such as Room-to-Room (R2R), which enabled consistent evaluation of navigation agents in realistic indoor environments [1]. Early work by Anderson et al. introduced sequence-to-sequence models that map visual observations and natural language instructions directly to navigation actions. This work established a baseline for learning-based VLN systems and highlighted the difficulty of grounding language in visual environments.

Following this, research efforts focused on improving how instructions are grounded during navigation. Fried et al. proposed speaker-follower models that generate auxiliary language descriptions to strengthen the connection between vision and language [2]. Attention mechanisms were also introduced to help agents focus on instruction-relevant visual cues. While these methods improved benchmark performance, they often required large models and extensive computation, limiting their practicality for real-time or embedded robotic systems.

To address these challenges, several works explored the use of explicit spatial representations and classical planning. Gupta et al. introduced cognitive mapping approaches that combine learned visual features with traditional planning algorithms, allowing agents to reason more explicitly about spatial structure [3]. Chaplot et al. further demonstrated that integrating semantic representations with SLAM-based exploration improves navigation efficiency and interpretability, particularly in complex environments [4]. These approaches highlight the benefits of combining learned perception with structured planning.

Simulation platforms have also played an important role in VLN research. The Habitat framework provides photorealistic environments and standardized evaluation tools, enabling large-scale experimentation without physical robots [5]. While simulation platforms accelerate research, they often assume access to significant computational resources and do not fully capture the constraints of real-world robotic deployment.

More recent work has explored the use of large language models (LLMs) for navigation reasoning. NavGPT introduces explicit reasoning by using LLMs to decompose instructions into interpretable navigation steps [6]. MapGPT extends this idea by incorporating map-guided prompting and adaptive path planning to improve long-horizon navigation performance [7]. Similarly, VLN-Game proposes a zero-shot semantic navigation framework that balances visual and linguistic signals through an equilibrium-based search process, demonstrating strong generalization to unseen environments [8]. Although these methods show promising results, they rely heavily on complex models and high computational cost.

In contrast to these approaches, this work focuses on a lightweight and modular navigation framework that emphasizes semantic-aware path planning using classical graph-based methods. Rather than relying on end-to-end learning or large language models during execution, the proposed system integrates semantic constraints directly into the planning process. This design prioritizes interpretability, feasibility, and system-level understanding, making it well-suited for analyzing the role of semantic reasoning under practical computational constraints.

III. System Architecture

The proposed system follows a modular architecture for Vision-Language Navigation that integrates perception, instruction abstraction, path planning, and simulation-based evaluation. A high-level overview of the complete framework is shown in Fig. 1. The modular design allows individual components to be developed, tested, and analyzed independently, while still supporting end-to-end navigation behavior.

The perception module processes visual observations and provides high-level semantic information about the environment, such as scene context and object-related cues. Rather than performing dense geometric reconstruction or real-time sensor fusion, this module supplies semantic signals that support reasoning and planning. In this work, perception is treated as a supporting component and does not directly influence low-level motion execution.

The instruction and goal abstraction module converts natural language navigation instructions into structured navigation goals. Instead of performing full language parsing or end-to-end instruction grounding, instructions are simplified into a start viewpoint and a goal viewpoint that align with the Room-to-Room task formulation. This abstraction reduces system complexity while preserving compatibility with standard vision-language navigation pipelines.

The path planning and routing module is the primary focus of this work. It operates on a graph-based representation constructed from R2R connectivity files, where nodes represent navigable viewpoints and edges represent valid transitions between viewpoints. Given a start and goal viewpoint, the planner computes a feasible route using shortest-path planning. Unlike purely geometric planning, which optimizes only distance or traversal cost, this module incorporates semantic costs into the planning process. These semantic costs penalize transitions that pass through undesirable or restricted regions, allowing the planner to balance geometric efficiency with semantic compliance. This explicit cost formulation enables interpretable analysis of how semantic reasoning influences navigation decisions.

The simulation and evaluation module executes the planned route using a deterministic pose propagation model and a grid-based map representation. While inspired by SLAM principles, this module does not perform probabilistic localization, sensor fusion, or loop closure. Instead, it provides a lightweight mechanism for visualizing navigation trajectories and evaluating path feasibility in a controlled simulation setting.

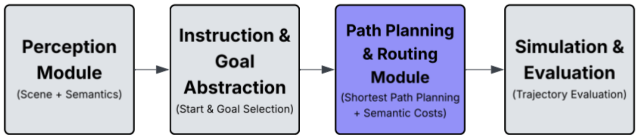

Fig. 1. High-level system architecture for vision-language navigation. The framework consists of perception, instruction and goal abstraction, path planning, and simulation-based evaluation modules. This paper focuses on the semantic-aware path planning and routing component.

Overall, the architecture is influenced by recent vision-language navigation frameworks that emphasize structured representations and multi-step planning. However, unlike approaches that rely on large language models or map-based prompting during execution, the proposed system integrates semantic reasoning directly into a classical planning pipeline. This design prioritizes interpretability, simplicity, and feasibility under computational constraints, making it well suited for system-level analysis of semantic-aware navigation.

IV. Data

This study uses data derived from the Room-to-Room (R2R) Vision-Language Navigation dataset along with its associated connectivity files and additional scene-level metadata. The R2R dataset is a widely used benchmark in VLN research and is built on top of the Matterport3D indoor environment. It provides human-written navigation instructions paired with discrete viewpoints that represent realistic indoor spaces such as homes, offices, and hallways.

Each navigation task in R2R consists of a start viewpoint, a goal viewpoint, and a natural language instruction describing how to reach the goal. Instead of relying on continuous sensor streams, the environment is represented as a discrete graph of viewpoints. This representation makes the dataset particularly suitable for studying graph-based navigation and path planning, which closely aligns with the goals of this work.

Table I summarizes the datasets used in this study and their roles within the navigation pipeline.

TABLE I
Datasets Used in This Study

Dataset	Purpose	Count
R2R Dataset	Navigation tasks and instructions	7,189 paths
R2R Connectivity Files	Viewpoint graph construction	90 scenes
MapGPT-72 Scenes	Simulation environments	72 scenes

Connectivity information is obtained from the official R2R connectivity files. These files specify which viewpoints within a scene are navigable and how they are connected. Using this information, each scene is converted into a graph where nodes represent viewpoints and edges represent valid transitions. These graphs serve as the core representation used by the path planning module and enable feasibility checking before any route is generated. By operating directly on the connectivity graph, the system ensures that all planned paths are physically navigable within the environment.

In addition to the R2R dataset, scene-level information from the MapGPT-72 dataset is used to support simulation and visualization. MapGPT-72 provides a collection of indoor scenes with structured spatial layouts, allowing navigation behavior to be evaluated across a broader range of environments. Although MapGPT is designed for language-guided navigation using large language models, in this work the dataset is used primarily to supply diverse scene layouts rather than to drive language-based decision making.

All experiments are conducted offline using pre-recorded data. No live sensor input or real-time perception is used during evaluation. This design choice allows for controlled analysis of semantic-aware path planning behavior while keeping system complexity low. While this limits realism, it provides a stable foundation for understanding how semantic constraints influence navigation decisions.

V. Methodology

This section describes the methodology used to design and evaluate the proposed semantic-aware navigation framework. The goal is not to build a fully autonomous robot system, but to study how semantic constraints influence path planning behavior in a structured and interpretable setting. The overall pipeline consists of environment representation, goal specification, path planning, and simulated execution.

The navigation process begins with representing the environment in a form suitable for planning. Rather than relying on raw sensor data, the environment is modeled using a grid-based structure derived from scene metadata and connectivity information. Semantic information is incorporated by assigning labels or constraints to specific regions of the map. In this work, these semantic labels are predefined or manually specified to isolate the effects of semantic reasoning without introducing additional perception-related uncertainty. This choice allows controlled experimentation while keeping the system lightweight.

Goal specification is performed by mapping navigation instructions to target locations within the environment. Instead of applying full natural language understanding, instructions are simplified into start and goal viewpoints that align with the Room-to-Room task formulation. This abstraction removes dependence on specific language models while remaining consistent with vision-language navigation scenarios. As a result, the system focuses on evaluating navigation decisions rather than instruction parsing.

Path planning is the core component of the proposed framework. Planning is performed using a grid-based search strategy that evaluates neighboring cells based on accumulated traversal cost. The planner maintains an open set of candidate cells and a cost map that records the cost of reaching each location. At each step, the cell with the lowest estimated cost is selected for expansion, and its neighbors are evaluated. The transition cost between cells is determined by spatial distance and additional semantic penalties. These penalties discourage traversal through regions that violate semantic constraints, such as restricted or undesirable areas. By adjusting these costs, the planner balances path length with semantic compliance. Although this planner is not optimized for real-time execution, it provides a transparent and interpretable mechanism for analyzing how semantic information affects route selection.

An overview of the execution flow for the semantic-aware planning process is shown in Fig. 2, including graph construction, feasibility checking, and simulated navigation.

To simulate navigation execution, a deterministic pose update model is used. The robot pose at time step t is represented as

$$\mathbf{p}_t = [x_t, y_t, \theta_t]^T, \tag{1}$$

where x_t and y_t denote the robot's position on the grid and θ_t represents its heading angle. Given an incremental motion

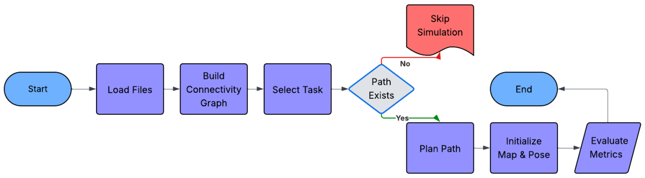

Fig. 2. Execution flow of the semantic-aware path planning pipeline, including graph construction, feasibility checking, and simulation-based evaluation.

command $\Delta\mathbf{p} = [\Delta x, \Delta y, \Delta\theta]^T$ derived from the planned path, the pose is updated according to

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \Delta\mathbf{p}. \tag{2}$$

The orientation is normalized to maintain a valid angular range:

$$\theta_{t+1} = (\theta_t + \Delta\theta) \bmod 360. \tag{3}$$

Following pose propagation, the environment map is updated using a simple grid-based occupancy representation. Each grid cell $M(u, v)$ is assigned a value based on the robot's interaction with the environment:

$$M(u,v) = \begin{cases} 1.0 & \text{if the cell is visited by the robot,} \\ 0.5 & \text{if an obstacle is detected nearby,} \\ 0.0 & \text{otherwise.} \end{cases} \tag{4}$$

Grid indices are computed from the continuous robot position as

$$(u, v) = \left(\left\lfloor \frac{x}{r} \right\rfloor, \left\lfloor \frac{y}{r} \right\rfloor \right), \tag{5}$$

where r denotes the grid resolution.

It is important to emphasize that this formulation does not represent a full probabilistic SLAM system. The robot pose is assumed to be known during simulation, and no uncertainty modeling, sensor fusion, or loop closure is performed. Instead, this SLAM-inspired approach provides a simplified mechanism for pose propagation and map updates, enabling qualitative analysis of navigation behavior without introducing additional computational complexity.

VI. Results

The experimental results show that incorporating semantic constraints has a clear effect on navigation behavior. In relatively simple indoor environments, the proposed planner consistently generated feasible paths that respected viewpoint connectivity while avoiding semantically restricted regions. Compared to purely geometric shortest-path planning, the inclusion of semantic costs encouraged more meaningful route selection, especially when multiple feasible paths were available.

Fig. 3 shows a representative navigation episode visualized on the Room-to-Room connectivity graph. In this example, a natural language instruction directs the agent to leave an office, move through intermediate rooms, and stop near the doorway of a sitting room. The instruction is abstracted into a valid start viewpoint and a goal viewpoint within the connectivity graph, resulting in a well-defined planning problem.

Given the specified start and goal, the planner generated a sequence of intermediate viewpoints forming a connected path through the environment. The resulting trajectory consisted of seven navigable viewpoints with a total path length of approximately 14.2 meters. Each waypoint corresponds to a valid node in the R2R graph, ensuring that all transitions satisfy connectivity constraints. This result demonstrates how the planner operates directly on the graph structure while balancing traversal distance with semantic cost considerations.

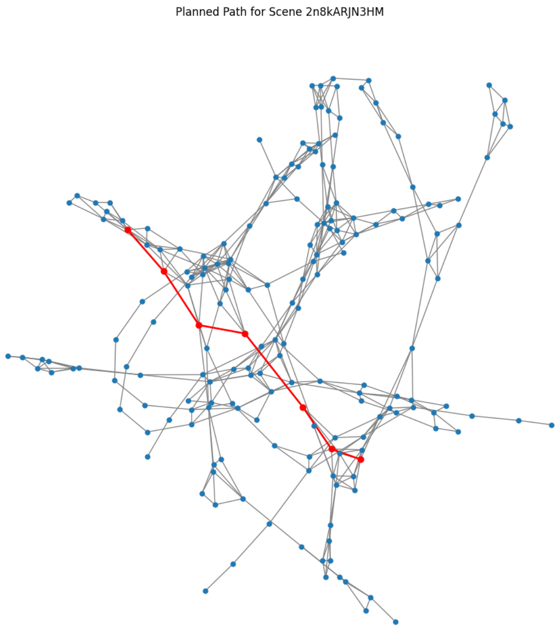

Fig. 3. Planned navigation path over the Room-to-Room connectivity graph for a representative scene. Nodes represent navigable viewpoints and edges denote valid transitions. The highlighted path corresponds to the route selected by the planner.

As environment complexity increased, both computation time and path length tended to increase. In denser connectivity graphs, the planner occasionally selected longer or less direct routes, reflecting the trade-off between semantic compliance and geometric optimality. These observations highlight the additional computational overhead introduced when semantic reasoning is incorporated into the planning process.

Fig. 4 presents the corresponding simulated execution trajectory for the same navigation episode. The 3D visualization illustrates pose propagation along the planned sequence of viewpoints, showing the progression from the start location to the goal. Although the trajectory is generated using a deterministic pose update model rather than a full probabilistic SLAM system, it provides useful insight into the spatial structure of the planned path and the effect of waypoint selection on execution behavior.

For this representative navigation episode, the system successfully reached the target location with a navigation error of

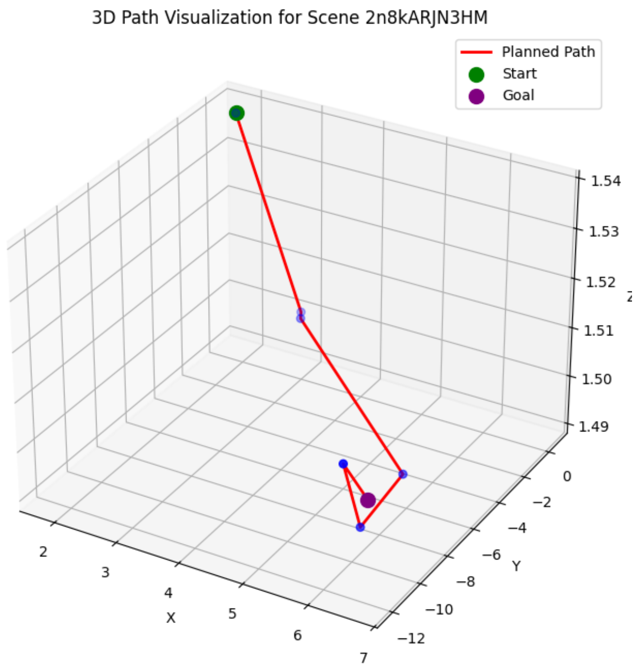

Fig. 4. 3D visualization of the simulated execution trajectory for the planned path. The start and goal viewpoints are highlighted, and the trajectory illustrates pose propagation over the selected waypoint sequence.

0.00 meters and a success weighted by path length (SPL) of 0.583. While these results are limited in scale and primarily qualitative, they confirm that the planner can generate feasible and goal-directed paths under semantic constraints.

In addition to standalone evaluation, the path planning module was integrated into a larger vision-language navigation pipeline that included semantic perception and language-based decision making. During execution, higher-level components occasionally failed due to missing dependencies, invalid language model outputs, or timeout conditions. In such cases, the system relied on the path planning module as a deterministic fallback to maintain navigation progress.

Despite interruptions in semantic reasoning, the integrated system successfully completed navigation episodes. This behavior highlights the robustness of classical planning methods when embedded within more complex semantic architectures. The planner consistently produced valid routes and enabled recovery from upstream failures without manual intervention, reinforcing its role as a reliable core component in hybrid navigation systems.

VII. Limitations and Future Work

A. Limitations

This study has several limitations, mainly due to its reliance on simulation and pre-recorded data. First, the proposed framework operates on offline datasets rather than real-time sensor input. Visual observations are obtained from the Matterport3D and R2R datasets, and navigation is performed using predefined connectivity graphs. As a result, the system does not model real-world sensing noise, moving obstacles, or changes in the environment that commonly occur during physical robot operation.

Second, navigation is simulated using discrete waypoint transitions instead of continuous robot motion. The robot pose is assumed to be known at all times and is updated deterministically during execution. While this abstraction allows for focused analysis of semantic-aware path planning, it does not capture localization uncertainty, actuator noise, or control errors that arise in real robotic systems.

Finally, computational constraints were not strictly enforced during evaluation. Although the framework was designed to remain lightweight, experiments were conducted on a development machine with sufficient processing power and memory. The performance of the system under the strict latency, memory, and power constraints of embedded robotic platforms was therefore not fully assessed.

B. Future Work

An important direction for future work is deploying the proposed framework on a physical robot for real-time semantic navigation. Initial attempts were made to implement the full system on a ROSMaster X3 Plus robot equipped with an NVIDIA Jetson Orin NX, a LiDAR sensor, and a depth camera. This hardware setup provides the necessary sensing capabilities for real-time localization, obstacle detection, and environment mapping.

Moving from a laptop-based simulation pipeline to a physical robot requires several key changes. In simulation, the system processes offline image folders, navigates using predefined connectivity graphs, and simulates movement through viewpoint transitions. On the robot, these components must be replaced with live camera feeds published over ROS topics, real-time localization using odometry or SLAM, and continuous motion control through velocity commands such as `/cmd_vel`. Graph-based planning would need to be replaced or combined with planning over SLAM-generated occupancy maps built from LiDAR and depth sensor data.

Deploying the complete architecture on embedded hardware also introduces challenges related to computational efficiency and real-time responsiveness. Future work will explore optimizing perception pipelines, selectively enabling semantic modules, and leveraging hardware acceleration available on the Jetson platform. Integrating classical ROS navigation components with semantic-aware planning remains a promising approach for achieving robust real-world navigation.

Overall, extending the framework to real-time robot deployment would allow evaluation under realistic conditions and provide deeper insight into how semantic reasoning, perception, and control interact in embodied navigation systems.

VIII. Conclusion

This paper presented a semantic-aware path planning framework for Vision-Language Navigation, with a focus on interpretability, modularity, and system-level understanding. Rather than aiming for end-to-end learning or benchmark optimization, the work examined how semantic information can be integrated into a classical graph-based planning pipeline and how this integration affects navigation behavior in indoor environments.

Traditional geometric path planning methods focus primarily on minimizing distance or travel cost, often without considering the meaning of the environment. In contrast, semantic-aware planning incorporates high-level information about rooms, objects, and regions, allowing navigation decisions to better align with human instructions and intent. The results of this study show that even simple semantic cost modeling can influence route selection, encouraging more meaningful paths when multiple geometric solutions are available.

Using standard VLN datasets and connectivity graphs, the proposed planner consistently generated feasible paths that respected environmental structure while accounting for semantic constraints. By operating directly on viewpoint graphs, the planner ensured navigability and provided a clear and transparent representation of routing decisions. This explicit structure makes the planning process easier to analyze and understand compared to black-box learning-based approaches.

A SLAM-inspired simulation framework was used to visualize navigation behavior and analyze trajectory execution. Although the pose updates and map construction were simplified and deterministic, they offered useful insight into how abstract planning decisions translate into spatial movement. Together, the graph-based and spatial visualizations helped connect high-level planning logic with low-level navigation behavior.

The path planning module was also integrated into a larger vision-language navigation architecture that included semantic perception and language-driven decision making. During execution, higher-level components occasionally failed due to missing dependencies, invalid outputs, or timeout conditions. In these situations, the classical planner acted as a reliable fallback, allowing the system to continue navigating toward the goal. This behavior highlights the importance of robust planning components in hybrid navigation systems that combine learned models with symbolic methods.

Several limitations were identified in this study, including the reliance on pre-recorded datasets, simulated execution, and the lack of strict computational constraints during evaluation. These limitations reflect common challenges faced when transitioning research prototypes toward real-world robotic deployment. By identifying these gaps, this work helps clarify the steps needed for future system improvement.

Overall, this research demonstrates that semantic-aware path planning provides a strong and interpretable foundation for Vision-Language Navigation systems. The proposed framework offers a practical balance between simplicity and expressiveness, making it suitable for integration with more advanced perception, language understanding, and localization modules. Future work focused on real-time robot deployment, tighter integration with SLAM, and optimization for embedded hardware will further strengthen the system and enable evaluation under realistic operating conditions. The insights from this study contribute to a clearer understanding of how classical planning methods can effectively complement modern semantic navigation approaches in embodied AI.

References

[1] P. Anderson *et al.*, "Vision-and-Language Navigation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[2] D. Fried *et al.*, "Speaker-Follower Models for Vision-and-Language Navigation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[3] A. Gupta, R. Vedantam, J. Malik, and D. Parikh, "Cognitive Mapping and Planning for Visual Navigation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] A. Chaplot *et al.*, "Learning to Explore Using Active Neural SLAM," in *International Conference on Learning Representations (ICLR)*, 2020.

[5] M. Savva *et al.*, "Habitat: A Platform for Embodied AI Research," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2019.

[6] G. Zhou *et al.*, "Explicit Reasoning in Vision-and-Language Navigation with Large Language Models," *arXiv preprint*, 2023.

[7] J. Chen *et al.*, "MapGPT: Map-Guided Prompting with Adaptive Path Planning for Vision-and-Language Navigation," *arXiv preprint*, 2024.

[8] L. Yu *et al.*, "VLN-Game: Vision-Language Equilibrium Search for Zero-Shot Semantic Navigation," *arXiv preprint*, 2024.