CausalNav: A Long-term Embodied Navigation System for Autonomous Mobile Robots in Dynamic Outdoor Scenarios

Author Names Omitted for Anonymous Review.

Abstract— **Autonomous language-guided navigation in large-scale outdoor environments remains a key challenge in mobile robotics, due to difficulties in semantic reasoning, dynamic conditions, and long-term stability. We propose CausalNav, the first scene graph-based semantic navigation framework tailored for dynamic outdoor environments. We construct a multi-level semantic scene graph using LLMs, referred to as the *Embodied Graph*, that hierarchically integrates coarse-grained map data with fine-grained object entities. The constructed graph serves as a retrievable knowledge base for Retrieval-Augmented Generation (RAG), enabling semantic navigation and long-range planning under open-vocabulary queries. By fusing real-time perception with offline map data, the *Embodied Graph* supports robust navigation across varying spatial granularities in dynamic outdoor environments. Dynamic objects are explicitly handled in both the scene graph construction and hierarchical planning modules. The *Embodied Graph* is continuously updated within a temporal window to reflect environmental changes and support real-time semantic navigation. Extensive experiments in both simulation and real-world settings demonstrate superior robustness and efficiency. The source code will be released to the community.**
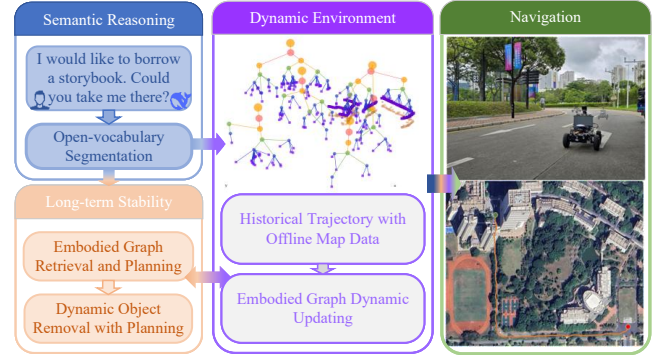
Fig. 1. The overall workflow of CausalNav. CausalNav introduces a novel embodied navigation framework that integrates open-vocabulary semantic reasoning, dynamic environment adaptation, and Embodied Graph-based planning. By dynamically updating the navigation graph with both historical and real-time data, the system enables robust, long-horizon, and language-directed navigation in complex outdoor environments.

I. INTRODUCTION

Recent advances in mobile robotics have shifted the focus beyond traditional control, perception, and navigation. Lifelong navigation demands not only accuracy, but also semantic understanding, incremental learning in dynamic environments, long-term robustness, and decision-making capabilities [1]. This calls for deeper integration between robotics and AI to enable scalable and adaptive autonomy.

Autonomous navigation in large-scale outdoor environments remains challenging due to their dynamic and unpredictable nature [2], [3]. Robots must perform both local and global planning while reasoning about semantics and high-level task intent. However, most visual-language navigation (VLN) benchmarks are confined to static indoor settings with step-by-step instructions, diverging from real-world scenarios where humans provide abstract goals and expect robots to understand semantics, infer spatial relations, and navigate from arbitrary start points [4].

Traditional outdoor navigation methods are primarily point-to-point and rely heavily on high-precision map construction [5]. However, most existing studies in this area offer limited or shallow semantic understanding and reasoning in open-world environments [6], [7], making sustained human–robot interaction difficult. Although learning-based navigation methods have advanced rapidly, their robustness in long-range, real-world scenarios remains insufficiently validated [8], [9]. They also struggle with dynamic environments and often require large-scale training data. Most

visual-language navigation (VLN) research is still conducted in indoor settings [10]–[13], with limited ability to handle environmental changes. In contrast, outdoor environments are large-scale, highly dynamic, and often degraded, posing greater challenges for embodied navigation and demanding higher levels of algorithmic robustness.

To address the above challenges, we propose CausalNav, an open-world semantic navigation system enhanced with offline map data. CausalNav leverages LLM-constructed scene graphs enhanced by RAG retrieval, coupled with hierarchical global-local planning, to achieve open-vocabulary long-range navigation in dynamic outdoor environments. Our framework extends recent efforts in applying RAG to robotics [14]–[16], introducing key improvements. By leveraging urban map data and real-time perception, we construct a multi-level Embodied Graph that integrates coarse-grained buildings and fine-grained object entities in urban campus environments. The graph is updated dynamically and supports multi-scale memory representations of navigation targets, while also incorporating dynamic objects within a temporal window. In addition, hierarchical summarization using LLMs improves the spatial abstraction capabilities of the Embodied Graph by organizing scene topology into semantically structured layers.

As illustrated in Fig. 1, our navigation framework integrates open-vocabulary semantic reasoning, dynamic environment adaptation, and Embodied Graph-based planning, which together enable language-directed, long-horizon, and robust navigation across dynamic outdoor scenes. Notably,

we demonstrate that deploying LLMs at the edge enables robust performance in urban campus environments without relying on commercial APIs (e.g., GPT-4o). Open-source models running locally on autonomous platforms offer a practical and effective alternative. In summary, our key contributions are as follows:

1) We construct a multi-level *Embodied Graph* with LLMs, integrating hierarchical semantics from coarse-grained map data to fine-grained object entities, serving as a RAG-retrievable knowledge base for semantic navigation.

2) We address dynamic object handling by continuously updating *Embodied Graph* and voxel maps at both the scene graph and planning levels. The graph is dynamically updated within a temporal window to support robust retrieval across different spatial scales.

3) We propose *CausalNav*, the first scene graph-based semantic navigation framework tailored for dynamic outdoor environments. We validate the system in both simulation and real-world campus scenarios. Compared with state-of-the-art methods, CausalNav demonstrates superior performance in terms of navigation success rate, trajectory efficiency, and robustness.

II. RELATED WORKS

A. Traditional Outdoor Navigation Methods

Achieving reliable and efficient navigation for ground robots in complex outdoor environments remains a long-standing challenge [2]. Traditional pipelines decompose navigation into independent modules—perception [4], localization [17], path planning, and control—offering interpretability and robustness. However, their reliance on high-definition (HD) maps and manual calibration limits scalability in large and dynamic environments. While suitable for structured scenarios without large-scale data or end-to-end training [5], these methods lack generalization to open-ended tasks and cannot handle natural language interaction. Their dependence on static maps and handcrafted rules further constrains adaptability to real-world dynamic scenes.

B. Learning-based Navigation Methods

Recent advances in learning-based navigation, particularly reinforcement learning (RL), have enabled direct policy learning from visual inputs [18]. Methods such as RE-CON [6] and ViKiNG [7] incorporate latent goal models, topological memory, and traversability estimation to support exploration and local planning. Meanwhile, diffusion models have emerged for local trajectory generation, with approaches like NoMaD [8] and ViNT [19] applying diffusion-based action or goal sampling for efficient planning and obstacle avoidance [20]. Despite their innovation, these methods are typically constrained to short-range navigation and lack explicit global path reasoning. Moreover, they often require large scale environment-specific training data and show limited generalization due to the sim-to-real gap. Their robustness in dynamic or long-horizon real-world scenarios remains insufficiently validated.

C. Scene Graph-based Navigation Methods

Scene graph-based visual-language navigation has rapidly progressed, especially in indoor settings [10]–[13]. Scene graphs provide high-level spatial abstractions that decouple perception from sensors and support hierarchical reasoning. Hydra [10] constructs hierarchical 3D scene graphs from metric to semantic levels, while VLMaps [12] fuses pre-trained vision-language features with 3D maps. LExis [13] aligns visual and language representations for semantic SLAM and topological planning. However, most of these methods remain constrained to indoor environments, where outdoor scale and dynamics pose major challenges.

With the rise of foundation models, recent works integrate LLMs for language-conditioned planning. SayPlan [11] combines LLMs with 3D scene graphs for scalable task reasoning, and LM-Nav [21] enables instruction-following for novel object goals. Yet, these systems often treat LLMs or VLMs as isolated evaluators rather than components of an integrated mapping framework. Moreover, hallucination and forgetting limit their reliability in dynamic, real-time outdoor scenarios.

III. METHODOLOGY

A. System Overview

As shown in Fig. 2, CausalNav adopts a hierarchical architecture comprising perception, graph construction, and planning components, tailored for long-horizon navigation in dynamic outdoor environments. Inspired by OpenGraph [22], the system constructs a multi-level Embodied Graph G composed of object nodes ν_i^{obj}, ego-vehicle nodes ν_i^l, building nodes ν_i^{build}, and hierarchical clustering nodes $\nu_i^{cluster}$. This graph integrates coarse-grained map information with fine-grained semantic entities. Its detailed structure is depicted in Fig. 4. In Sec. III-B semantic alignment is achieved by leveraging an open-vocabulary perception model [23] and LiDAR-based localization [24], enabling the extraction of object and ego-vehicle node information.

In Sec. III-C, the system distinguishes dynamic, static, and quasi-static objects through multi-object tracking. To mitigate the impact of environmental dynamics on the construction of the Embodied Graph, a spatio-temporal filtering mechanism is applied before updating the semantic graph. By introducing building nodes and performing hierarchical clustering with the assistance of large language models (LLMs), the constructed Embodied Graph supports retrieval-augmented generation, enabling the system to respond effectively to human language instructions.

In Sec. III-D, the system continuously integrates offline maps with live perception data to update the graph in real time. Global navigation relies on either pre-stored routes or dynamically generated scene graphs, whereas local path planning is adaptively performed using real-time LiDAR mapping and on-the-fly removal of dynamic obstacles. This design enables flexible, language-driven navigation with long-term stability and semantic awareness.
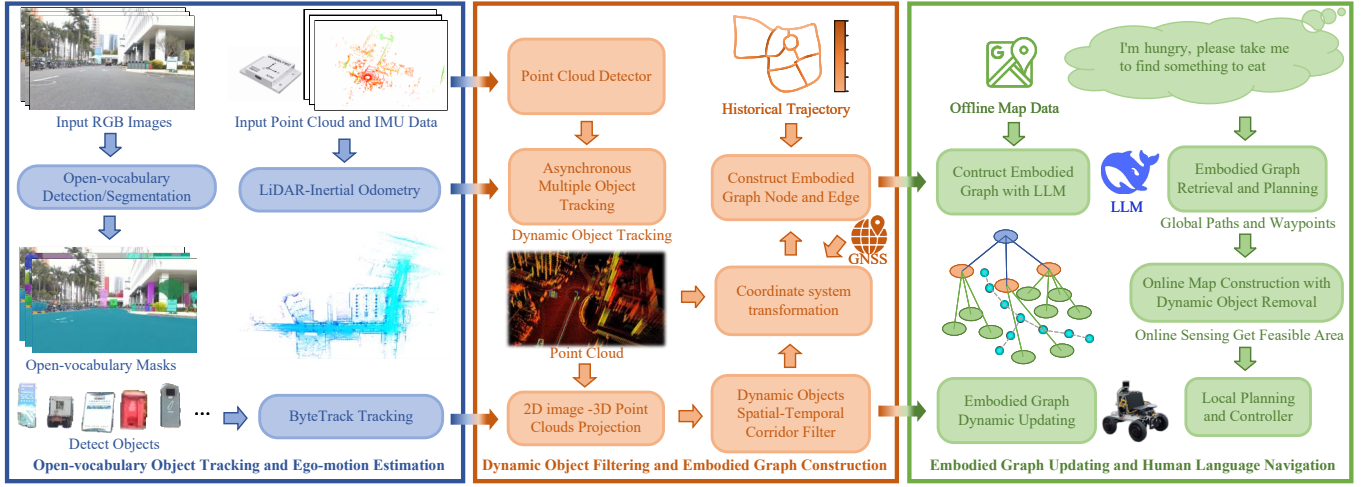
Fig. 2. The CausalNav framework comprises three sequential modules: (1) Open-vocabulary Object Tracking and Ego-motion Estimation(in Sec. III-B): Integrates RGB, LiDAR, and IMU inputs for open-vocabulary object detection and tracking, along with ego-motion estimation via 2D–3D spatial-temporal alignment. (2) Dynamic Object Filtering and Embodied Graph Construction(in Sec. III-C): Filters transient dynamic objects through asynchronous multi-object tracking and constructs a temporally stable embodied scene graph. (3) Embodied Graph Updating and Human Language Navigation(in Sec. III-D): Builds and updates a semantic graph using LLMs to interpret language commands and perform hierarchical planning, with real-time dynamic object removal for robust navigation.

B. Open-vocabulary Object Tracking and Ego-motion Estimation

1) Open-vocabulary Object Tracking: To obtain rich information about the description, shape, size, and position of object nodes, we denote the object state as $^w\mathbf{T}_{\text{obj}} = \{^w\mathbf{R}_{\text{obj}}, ^w\mathbf{p}_{\text{obj}}\}$, where the object's pose $^w\mathbf{T}_{obj}$ represented in the $SE(3)$ space. In outdoor environments, we use YOLO-World [23], a lightweight open-vocabulary detector, to extract 2D bounding boxes and segmentation masks from each RGB frame I_t. These detections are then temporally associated by the multi-object tracking function $\mathcal{C}(\cdot)$, implemented by ByteTrack [25], yielding stable tracking results:

$$\mathcal{S}_t = \mathcal{C}(\text{YOLO-World}(I_t)). \tag{1}$$

Here, $\mathcal{S}_t$ denotes the set of detected objects in the current frame, defined as

$$\mathcal{S}_t = \{S_i = (c_i, 2\text{DBBox}_i, \mathcal{B}_i) \mid i = 1, 2, \ldots, n\}, \tag{2}$$

where $\mathcal{B}_i$ is the segmentation mask, 2DBBox_i the 2D bounding box, and c_i the description for the i-th tracked object.

Due to the limited performance of depth cameras in real outdoor environments, we fuse camera and LiDAR data for object localization. The raw LiDAR point cloud is denoted as $\mathcal{P} = \{\mathbf{P}_i = (x_i, y_i, z_i)\}_{i=1}^N$. For the current time step t, each LiDAR point $\mathbf{P}_i \in \mathcal{P}_t$ is projected onto the image plane of frame I_t via the calibrated camera projection model, yielding image-space coordinates $^c\mathbf{p}_i$, computed as: $^c\mathbf{p}_i = \mathbf{K} \cdot \mathbf{H} \cdot \mathbf{P}_i$, where $\mathbf{K}$ is the camera intrinsic matrix and $\mathbf{H}$ is the extrinsic transformation matrix obtained through joint LiDAR-camera calibration [26].

We then define the object-specific 3D point cloud in the camera frame as:

$$^l\mathcal{P}_{\text{obj}} = \{\mathbf{P}_i \in \mathcal{P}_t \mid ^c\mathbf{p}_i \in \mathcal{B}_i\}. \tag{3}$$

Based on the filtered set $^l\mathcal{P}_{\text{obj}}$, a minimum-volume 3D bounding box is constructed. The centroid of this bounding box is taken as the estimated 3D position of the object, denoted as $^l\mathbf{p}_{\text{obj}}$.

We obtain the current ego-pose $^w\mathbf{T}_l$ from a LiDAR-inertial odometry system [24]. The relative pose of the detected object in the LiDAR frame is denoted as $^l\mathbf{T}_{\text{obj}}$. The corresponding world-frame object pose is calculated by $^w\mathbf{T}_{\text{obj}} = ^w\mathbf{T}_l \cdot ^l\mathbf{T}_{\text{obj}}$. For incremental topological graph updates, each tracked object contributes a node defined as:

$$\nu_i^{obj} = \{c_i, 3\text{DBBox}_i, ^w\mathbf{p}_{\text{obj}}\}, \tag{4}$$

which is inserted or updated into the Embodied Graph G.

For each tracked object ν_i^{obj}, if it is not in the graph G, a new node is created; otherwise, its position is updated:

$$G \leftarrow \begin{cases} G \cup \{\nu_i^{obj}\}, & \text{if } \nu_i^{obj} \notin G \\ G \setminus \{^{old}\nu_i^{obj}\} \cup \{\nu_i^{obj}\}, & \text{if } ^{old}\nu_i^{obj} \in G. \end{cases} \tag{5}$$

2) Ego-motion Estimation: For the ego-vehicle, its state is represented as $^w\mathcal{E}_l = [^w\mathbf{R}_l, \ ^w\mathbf{p}_l, \ ^w\boldsymbol{\xi}_l]$, where $^w\mathbf{R}_l$ denotes orientation, $^w\mathbf{p}_l$ the position, and $^w\boldsymbol{\xi}_l$ the velocity in the world frame. When the movement step exceeds a threshold d, the position and velocity are stored as an ego-vehicle node $\nu_i^l = \{^w\mathbf{p}_l, \ ^w\boldsymbol{\xi}_l\}$, and a new edge $e_i^l = (\nu_i^l, \ \nu_{i-1}^l)$ is added to G. As shown in Fig. 4, the ego-vehicle nodes ν_i^l and edges e_i^l, representing the historical trajectory, are stored in the Embodied Graph and later utilized for the global planning task described in Sec. III-D.2.

C. Dynamic Object Filtering and Embodied Graph Construction

1) Dynamic Objects Spatial-Temporal Corridor Filter: To robustly handle dynamic objects in outdoor environments, we employ a multi-object tracking pipeline that integrates
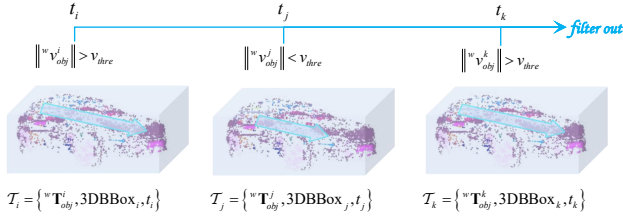
Fig. 3. Illustration of three observed trajectory points and their corresponding 3D bounding boxes within the spatial-temporal corridor. The point cloud depicts the same vehicle captured at different timestamps.

BEV-based detection with motion estimation. Specifically, CenterPoint [27] performs real-time point cloud detection, accelerated via TensorRT, while LIOsegmot [28] estimates object velocities by fusing detections with LiDAR-inertial odometry. This enables reliable classification of objects as dynamic, static, or quasi-static.

To mitigate false positives in the Embodied Graph caused by traditional velocity-based filters [29], we encode each object's trajectory as a spatial-temporal corridor using historical bounding boxes, as illustrated in Fig. 3. This representation enhances the filtering of dynamic entities and preserves the structural integrity of the graph.

$$\mathcal{T} = \left\{ {}^{w}\mathbf{T}_{\mathrm{obj}}^{i}, 3\mathrm{DBBox}_i, t_i \right\}_{i=1}^{n}. \tag{6}$$

When an object exceeds a displacement threshold of k steps, its spatial-temporal corridor $\mathcal{T}$ is excluded, and the corresponding dynamic nodes ${}^{d}\nu_i^{\mathrm{obj}}$ are removed from graph construction:

$$G \leftarrow G \setminus \{\mathcal{T} \mid \mathcal{T} \in D\}. \tag{7}$$

This corridor-based filtering effectively removes transient or mobile objects from the Embodied Graph, reducing motion-induced errors. It is particularly effective for entities with intermittent motion patterns, such as vehicles near intersections.

2) Hierarchical Construction of the Embodied Graph: Beyond the ego-vehicle nodes described in Sec. III-B.2, we construct a hierarchical spatial–semantic representation of the static environment after filtering transient dynamic entities via the spatial–temporal corridor. The resulting Embodied Graph integrates geometric and semantic information, serving as the foundation for downstream reasoning and navigation.

The static topological graph includes two node types: (1) building nodes $\nu_i^{\mathrm{build}} = \{c_i^{\mathrm{build}}, {}^{w}\mathbf{p}_{\mathrm{build}}\}$, extracted from offline maps and assigned the highest level L; and (2) object nodes ν_i^{obj}, representing fine-grained elements (e.g., fire hydrants, bus stops) at level $L-1$. Together, they form a multi-level spatial–semantic abstraction of the environment.

Following [15], hierarchical clustering is performed with a spatial–semantic similarity measure:

$$\kappa_{ij} = (1 - \alpha)\kappa_{ij}^{\mathrm{spatial}} + \alpha\kappa_{ij}^{\mathrm{semantic}}, \tag{8}$$

where $\kappa_{ij}^{\mathrm{spatial}} = \exp(-d_{\mathrm{haversine}}(i,j)/\theta)$, and $\kappa_{ij}^{\mathrm{semantic}}$ is the cosine similarity between embeddings $\mathbf{e}_i$ and $\mathbf{e}_j$.

Embedding-based similarity enhances robustness to LLM label variations (e.g., "trash can" vs. "garbage bin").

Level-$L-1$ object nodes are grouped bottom-up to form level-L clusters $\nu_i^{\mathrm{cluster}} = \{\mathrm{LLM}(r), \mathrm{centroid}(r)\}$. Each cluster $r \in \mathcal{R}$ contains several nodes, with its semantics summarized by an LLM and position given by the mean of its members. Clustering proceeds recursively, jointly considering cluster and building nodes, until convergence.

3) Semantic Retrieval over the Embodied Graph: Given a query q, semantic retrieval is performed hierarchically using LLM-based selection. At level l, the selection probability of node $n_l \in \mathcal{L}_l$ is:

$$\pi(n_l \mid q) = \frac{\exp[\gamma \cdot \mathrm{LLM}(q, C(n_l))]}{\sum_{n' \in \mathcal{L}_l} \exp[\gamma \cdot \mathrm{LLM}(q, C(n'))]}, \tag{9}$$

where $C(n_l)$ denotes the node description and $\gamma > 0$ controls sharpness.

The hierarchical path score is:

$$\Lambda(\zeta) = \prod_{l=1}^{D} [\pi(n_l \mid q) \cdot \phi(n_l, n_{l-1})], \tag{10}$$

where $\phi(n_l, n_{l-1}) = \mathbf{1}_{\{n_{l-1} \in \mathrm{Children}(n_l)\}}$ ensures valid parent–child links.

If the agent's location $\mathbb{L}$ is known, candidates are re-ranked by a hybrid score:

$$\eta(n) = \beta\kappa^{\mathrm{spatial}}(n, \mathbb{L}) + (1 - \beta)\Lambda(\zeta), \tag{11}$$

where $\kappa^{\mathrm{spatial}}(n, \mathbb{L}) = \exp[-d_{\mathrm{haversine}}(n, \mathbb{L})/\theta]$ measures spatial proximity, and $\beta \in [0, 1]$ balances spatial and semantic relevance.

Evaluating $\eta(n)$ across candidate paths allows the system to select nodes most aligned with the query both semantically and spatially for downstream planning.

D. Embodied Graph Updating and Human Language Navigation

1) Online Embodied Graph Updating: Essentially, the *Embodied Graph* serves as both a Scene Operator and a Memory Tank, abstracting elements such as time, scenes, objects, and events into structured memory representations. Its core value lies in enabling robots to understand complex environments—that is, to perform a form of abstract reasoning—which facilitates more intelligent and diverse decision-making in navigation tasks. During the online construction and updating of the Embodied Graph, we focus on the following aspects:

- Multi-granularity topological representation of scenes;
- Spatial relationships among static objects within a single scene;
- Modeling and management of dynamic objects over temporal windows;
- Implicit encoding of inter-scene relationships.

The online update process of the Embodied Graph is formally described in Algorithm 1.

Algorithm 1 Online Embodied Graph Updating

Require: $\mathcal{C}$, k
1: $G \leftarrow \emptyset$, $t \leftarrow 0$
2: **while** System is running **do**
3:      $t \leftarrow t + 1$, $\mathcal{S}_t \leftarrow \mathcal{C}(I_t, \mathcal{P}_t, \text{IMU})$
4:      **for all** $S_i \in \mathcal{S}_t$ **do**
5:          **Compute** $^w\mathbf{T}_{\text{obj}}$, $\mathcal{B}_i$, 3DBBox$_i$
6:          $\nu_i^{obj} \leftarrow \{c_i, \text{3DBBox}_i, {}^w\mathbf{p}_{\text{obj}}\}$
7:          **if** $\nu_i^{obj} \notin G$ **then** $G \leftarrow G \cup \{\nu_i^{obj}\}$
8:          **else** $G.\textbf{update}(\nu_i^{obj})$
9:          **end if**
10:      **end for**
11:      **for all** $^d\nu_j^{obj} \in \{\mathcal{T} \mid \mathcal{T} \in D\}$ **do**
12:          **if** $^d\nu_i^{obj}.\textbf{steps} \geq k$ **then**
13:            $G \leftarrow G \setminus \{\mathcal{T}\}$
14:          **end if**
15:      **end for**
16:      **Update** ν_i^l, E_ν for all $\nu \in G$
17:      $\mathcal{R} \leftarrow \textbf{HCluster}(G)$
18:      **for all** $r \in \mathcal{R}$ **do**
19:          $E_r \leftarrow \{(\nu_i^{cluster}, \nu_i^{obj}) \mid \nu_i^{obj} \in r\}$
20:          $G \leftarrow G \cup \{\nu_i^{cluster}\} \cup E_r$
21:      **end for**
22: **end while**
23: **return** G

---

2) Human Language Navigation:

CausalNav performs navigation in response to human language instructions. Based on the semantic retrieval method described in Sec. III-C.3, the system first infers a global target location from the human language input by leveraging the real-time constructed and updated *Embodied Graph*. If the target is not reachable from the robot's historical trajectory, a coarse global route is generated using either offline road map data or external APIs such as Google Maps or Amap. If the target is connected via past trajectories, a Dijkstra-based shortest path is computed. The resulting global path is represented as a waypoint sequence $\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n\}$.

In real-world outdoor environments, robots must operate in highly dynamic scenes. Residual motion trails from dynamic objects often appear as static obstacles on the map, degrading localization and navigation performance. To mitigate this, we adopt RH-Map [30], a 3D region-wise hash map framework for real-time local mapping and dynamic object removal.

The feasible region for planning that the mobile robot can obtain through RH-Map is denoted as $\mathcal{F} \subseteq \mathbb{R}^3$. Within $\mathcal{F}$, an initial trajectory is generated using the informed-RRT* [31] algorithm:

$$\mathcal{Z}_{\text{init}} = \{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_N\}, \quad \mathbf{z}_i = [x_i, y_i, z_i]^T. \quad (12)$$

The path $\mathcal{Z}_{\text{init}}$ is then smoothed via B-spline interpolation. Each interpolated point is assigned an orientation θ_i computed from the direction of adjacent waypoints. This yields

an oriented reference trajectory:

$$\mathbf{X}_g = \{\mathbf{x}_g^0, \mathbf{x}_g^1, \ldots, \mathbf{x}_g^M\}, \quad \mathbf{x}_g^i = [x_i, y_i, z_i, \theta_i]^T \in \mathbb{R}^4. \quad (13)$$

Given the current robot state $\mathbf{x}_t = [x_t, y_t, z_t, \theta_t]^T$ and control input $\mathbf{u}_t = [v_t, \omega_t]^T$, we formulate a Nonlinear Model Predictive Control with Control Barrier Function [32] (NMPC-CBF) problem as a constrained optimization:

$$\min_{\{\mathbf{x}_k, \mathbf{u}_k\}} \sum_{k=0}^{N-1} (\|\mathbf{x}_k - \mathbf{x}_g^k\|_Q^2 + \|\mathbf{u}_k\|_R^2) \quad (14a)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \quad (14b)$$

$$\mathbf{x}_0 = \mathbf{x}_{\text{init}} \quad (14c)$$

$$\mathbf{x}_k \in \mathcal{X}, \quad \mathbf{u}_k \in \mathcal{U}, \quad (14d)$$

$$\Delta h_{\text{ob}}^i(\mathbf{x}_k, \mathbf{u}_k) + \lambda_k h_{\text{ob}}^i(\mathbf{x}_k) \geq 0. \quad (14e)$$

Here, $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$ denote the feasible sets of states and control inputs, respectively, incorporating physical and safety constraints. The last constraint incorporates a control barrier function that guarantees the forward invariance of the safe set defined by dynamic obstacles such as pedestrians or vehicles. Each barrier function $h_{\text{ob}}^i(\mathbf{x}) > 0$ defines the safety margin to obstacle i, evolving with its predicted trajectory, typically formulated as $h_i(\mathbf{x}) = (x - x_i^p)^2 + (y - y_i^p)^2 - d_{\text{safe}}^2$, where $[x_i^p, y_i^p]^T$ denotes the predicted position of obstacle i, and d_{safe} represents the safety radius.

IV. EXPERIMENTS AND RESULTS

A. Experiments Setup

Simulation Environment. As shown in Fig. 4, we simulate a Gazebo-based ground robot equipped with a RealSense D435i and 3D LiDAR in an urban environment. The Embodied Graph is dynamically constructed and visualized during navigation. A Gazebo-based state estimator logs task completion and trajectories. Due to the lack of suitable open-source alternatives, we compare our method with four representative learning-based systems: NoMaD, ViNT, GNM, and City-Walker, all of which perform goal-driven motion prediction with obstacle avoidance. For fair comparison, we collected image data to build topological maps for NoMaD, ViNT, and GNM, and adopted our generated waypoints for CityWalker. Performance is evaluated on 25 randomly sampled tasks (10 trials each) using four metrics: Success Rate (SR), Success weighted by Path Length (SPL), Collision Count (CC), and Trajectory Length (TL). A task succeeds if the robot reaches within 10 meters of the target. The system runs on an Intel i9-14900K CPU and a single RTX 3090 GPU, with key modules operating in real time: Open-vocabulary object tracking and ego-motion at 30 Hz, spatio-temporal corridor filtering at 20 Hz, local dynamic mapping and planning at 10 Hz, and hierarchical clustering or Embodied Graph updates at 1 Hz.

Real Environment. As shown in Fig. 5, we deploy our system on a wheeled mobile robot equipped with an Intel Core i9-13900H CPU and an NVIDIA GeForce RTX 4070 GPU. A RealSense D435i camera captures RGB images, while the RSHelios lidar and an RTK GNSS/INS module
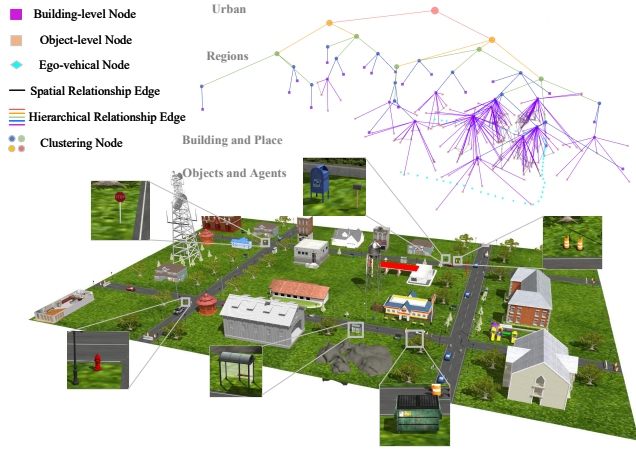
Fig. 4. The simulation environment and the constructed Embodied Graph. The environment includes coarse-grained objects (e.g., buildings) and fine-grained ones (e.g., fire hydrants, mailboxes). The Embodied Graph fuses both levels and updates dynamically as the agent moves.

Fig. 5. The robot used in real-world navigation experiments.

provide positioning data at 10 Hz with 5 cm accuracy. We utilize FAST-LIO2 [24] in conjunction with RTK GNSS/INS for coordinate transformation and precise localization. The system was tested in a large-scale, campus-like outdoor environment to evaluate its effectiveness and robustness in dynamic real-world scenarios.

B. Simulation Environment Experiments

This experiment evaluates our approach in long-range navigation tasks, focusing on success rate, navigation efficiency, and adaptability to dynamic environments. In simulation, pedestrians and vehicles emulate real-world complexity. According to the spatial scale of the navigation target, tasks are categorized into short-range, medium-range, and long-range scenarios. The results are summarized in Table I. In terms of SR and SPL, both CityWalker and CausalNav exhibit strong performance. However, analysis of TL in long-range tasks reveals that topological methods such as ViNT, NoMaD, and GNM suffer from unidirectional connectivity, resulting in inefficient path planning and significantly longer trajectories even when the destination is reachable. While CityWalker and CausalNav achieve comparable results on SR, SPL, and TL, they differ significantly in CC. CityWalker, as a learning-based method, is easy to deploy without extensive parameter tuning. However, it shows limited generalization in dynamic environments, often failing to avoid moving obstacles. In contrast, CausalNav enhances dynamic responsiveness by constructing safe zones in real time and performing perception and re-planning within about 100 ms, leading to more robust navigation in dynamic settings. Overall, CausalNav outperforms existing baselines across multiple metrics, demonstrating superior performance in long-range success rate, trajectory efficiency, and dynamic adaptability.

Different LLM on Navigation. Large Language Models (LLMs) play a critical role in Embodied Graph construction and semantic reasoning for navigation. However, relying on online APIs is im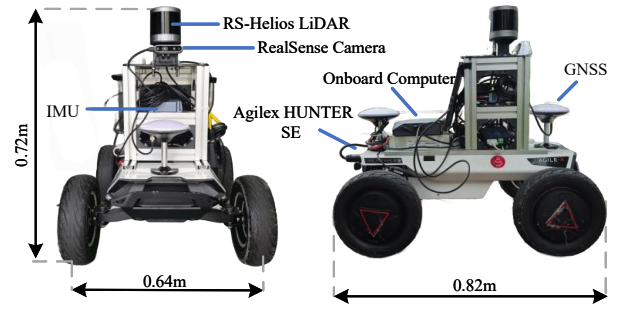practical in real-world deployments due to limited network access, high latency, and data privacy concerns. This highlights the need for locally deployed LLMs in autonomous navigation. We evaluate four recent open-source LLMs with comparable scales, along with the GPT-4o online API, in a simulated environment where the Embodied Graph is built online. Navigation performance is summarized in Table II. Although GPT-4o achieves the best performance, its margin over smaller open-source models is modest, owing to hierarchical semantic retrieval within the Embodied Graph, which improves query accuracy and mitigates hallucinations. Among open-source models, Deepseek-R1-Distill-14B shows the highest success rate and most stable performance.

Impact of Dynamic Updates to the Embodied Graph. In outdoor environments, localized changes, such as moving pedestrians and vehicles, can introduce temporary occlusions and traversability changes that disrupt the consistency between the Embodied Graph and real-time perception. Although the initial scene graph is constructed through full-environment exploration, it quickly becomes outdated without continuous updates. We evaluated the impact of such updates through a series of continuous navigation tasks with randomly generated destinations, simulating real-world scenarios. As shown in Table III, the dynamic update strategy yields significant improvements in SR, SPL, CC, and TL metrics compared to the static baseline. These results demonstrate that even without global semantic changes, maintaining up-to-date local information is critical for accurate semantic retrieval and reliable navigation performance.

Runtime Efficiency. To further evaluate computational efficiency, we compare the average per-cycle latency with representative baselines under identical hardware configurations. As shown in Table IV, CausalNav achieves real-time performance (10 Hz) with only a 11% overhead over NoMaD, demonstrating competitive runtime efficiency while providing richer semantic reasoning and dynamic graph maintenance.

Ablation on Semantic Retrieval Parameters. As shown in Fig. 6, accuracy and recall follow bell-shaped trends, peaking near $\alpha=\beta=0.5$ and $\gamma=1.5$, which indicates balanced spatial–semantic fusion and stable LLM reasoning. The trajectory length slightly decreases with higher spatial weighting, showing that moderate spatial bias improves efficiency without sacrificing semantic precision.

TABLE I

COMPARATIVE ANALYSIS OF NAVIGATION PERFORMANCE ACROSS VARIOUS SEMANTIC NAVIGATION METHODS IN SIMULATION ENVIRONMENT.

Method	Small				Medium				Large			
	SR(%)↑	SPL(%)↑	CC↓	TL(m)↓	SR(%)↑	SPL(%)↑	CC↓	TL(m)↓	SR(%)↑	SPL(%)↑	CC↓	TL(m)↓
ViNT [19]	84	68.4	0.6	47.16	64	47.8	0.9	92.74	48	32.2	1.6	160.03
NoMaD [8]	82	70.9	0.8	45.61	43	34.2	1.3	81.13	22	14.6	2.3	173.72
GNM [33]	84	72.3	0.5	42.02	26	15.2	1.5	88.32	0	0	-	-
CityWalker [9]	100	82.4	1.2	43.07	85	73.6	3.4	86.32	80	68.3	4.5	136.63
CausalNav	100	88.9	0.2	40.66	92	82.2	0.6	83.16	80	66.0	1.2	141.82

TABLE II

PERFORMANCE OF DIFFERENT LLMs ON NAVIGATION

Method	SR(%)↑	SPL(%)↑	CC↓	TL(m)↓
phi4-14B	83	69.5	1.1	106.27
Qwen3-14B	82	70.4	1.0	102.45
Gemma3-12B	84	67.6	1.2	110.32
DeepSeek-R1-Distill-14B	85	72.1	1.1	103.63
GPT-4o	88	75.3	1.0	103.24

TABLE III

THE IMPACT OF ONLINE EMBODIED GRAPH UPDATING ON NAVIGATION

Method	SR(%)↑	SPL(%)↑	CC↓	TL(m)↓
CausalNav †	78	54.7	1.8	120.35
CausalNav ‡	90	80.1	1.1	98.25

†Without Embodied Graph updates; ‡With Embodied Graph updates.

C. Real-world Environmental Experiments

We deployed CausalNav on robotic platforms and conducted real-world comparisons with ViNT, NoMaD, GNM, and CityWalker, as shown in Fig. 7. Experiments were performed under two conditions. The first scenario (Fig. 7(a)) involved short-range navigation (130 m) with object-level instructions, where only ViNT and CausalNav succeeded. The second scenario (Fig. 7(b)) tested long-range navigation (512 m) using building-level instructions; only CausalNav completed the task, while others failed due to collisions. Simulation results are further validated in real-world settings. Both ViNT and CausalNav succeed in completing navigation tasks over distances of approximately 100 meters; however, only CausalNav is able to successfully perform long-range navigation in highly dynamic outdoor environments exceeding 500 meters. In particular, CityWalker exhibits significantly lower performance in real-world experiments compared to simulation. We observed that CityWalker is highly sensitive to lighting conditions and environmental changes, leading to unstable trajectory predictions and inadequate handling of dynamic objects. Although it demonstrates a high success rate in long-range tasks under simulation, the real world introduces additional complexities not captured in simulation. In particular, minor collisions that are tolerable in simulation may lead to task failure in real-world deployments. These findings confirm the robustness of CausalNav for long-distance semantic navigation in complex

TABLE IV

MEAN PER-CYCLE LATENCY RUNTIME COMPARISON

Method	NoMaD [8]	ViNT [19]	GNM [33]	CityWalker [9]	CausalNav
Runtime (ms) ↓	95	150	110	180	105

Fig. 6. Ablation results on key parameters. From left to right: α, β, and γ. Left axis: Accuracy / Recall (%); Right axis: Trajectory Lengt (m).
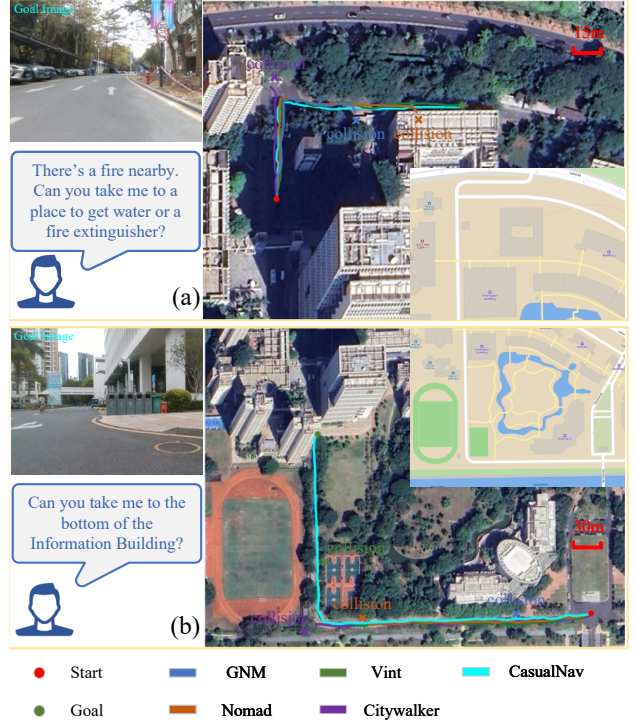

Fig. 7. Experiments under different distance scales in real-world scenarios.

and dynamic urban environments.

V. CONCLUSION AND FUTURE WORK

This work presents *CausalNav*, the first scene-graph-based semantic navigation framework for dynamic outdoor environments. The system integrates LLM-constructed multi-level *Embodied Graph* with retrieval-augmented reasoning

and hierarchical planning, enabling long-range and open-vocabulary navigation with real-time adaptability. Extensive experiments in simulation and real-world settings demonstrate the effectiveness of our approach in improving outdoor navigation robustness and efficiency.

Discussion and Limitations. While effective in dynamic scenes, CausalNav still faces limitations in scalability, robustness under extreme lighting or weather, and long-horizon consistency. Compared with previous graph-memory systems such as *EmbodiedRAG* [15] and *NavRAG* [16], CausalNav extends the RAG paradigm to dynamic outdoor navigation via bidirectional LLM–RAG reasoning and real-time multi-level graph updates, achieving open-vocabulary understanding and continuous spatial-semantic adaptation.

Future Work. We plan to enhance graph compression and memory recall mechanisms to improve scalability, explore multimodal fusion for better robustness under extreme conditions, and extend the system toward long-term autonomous exploration and lifelong learning.

REFERENCES

[1] P. Yin, J. Jiao, S. Zhao, L. Xu, G. Huang, H. Choset, S. Scherer, and J. Han, "General place recognition survey: Towards real-world autonomy," *IEEE Transactions on Robotics*, pp. 1–20, 2025.

[2] T. Guan, D. Kothandaraman, R. Chandra, A. J. Sathyamoorthy, K. Weerakoon, and D. Manocha, "Ga-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8138–8145, 2022.

[3] P. Yin, A. Abuduweili, S. Zhao, L. Xu, C. Liu, and S. Scherer, "Bioslam: A bioinspired lifelong memory system for general place recognition," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4855–4874, 2023.

[4] F. Zhu, X. Liang, Y. Zhu, Q. Yu, X. Chang, and X. Liang, "Soon: Scenario oriented object navigation with graph-based exploration," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 12 684–12 694.

[5] S. Niijima, R. Umeyama, Y. Sasaki, and H. Mizoguchi, "City-scale grid-topological hybrid maps for autonomous mobile robot navigation in urban area," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2065–2071.

[6] D. Shah, B. Eysenbach, N. Rhinehart, and S. Levine, "Rapid exploration for open-world navigation with latent goal models," in *Conference on Robot Learning*. PMLR, 2022, pp. 674–684.

[7] D. Shah and S. Levine, "ViKiNG: Vision-Based Kilometer-Scale Navigation with Geographic Hints," in *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022.

[8] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 63–70.

[9] X. Liu, J. Li, Y. Jiang, N. Sujay, Z. Yang, J. Zhang, J. Abanes, J. Zhang, and C. Feng, "Citywalker: Learning embodied urban navigation from web-scale videos," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 6875–6885.

[10] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3D scene graph construction and optimization," in *Robotics: Science and Systems*, 2022.

[11] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "Sayplan: Grounding large language models using 3D scene graphs for scalable robot task planning," in *Conference on Robot Learning*. PMLR, 2023, pp. 23–72.

[12] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 608–10 615.

[13] C. Kassab, M. Mattamala, L. Zhang, and M. Fallon, "Language-extended indoor slam (lexis): A versatile system for real-time visual scene understanding," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 15 988–15 994.

[14] Q. Xie, S. Y. Min, T. Zhang, K. Xu, A. Bajaj, R. Salakhutdinov, M. Johnson-Roberson, and Y. Bisk, "Embodied-rag: General non-parametric embodied memory for retrieval and generation," in *Language Gamification-NeurIPS 2024 Workshop*.

[15] M. Booker, G. Byrd, B. Kemp, A. Schmidt, and C. Rivera, "Embodiedrag: Dynamic 3D scene graph retrieval for efficient and scalable robot task planning," *arXiv preprint arXiv:2410.23968*, 2024.

[16] Z. Wang, Y. Zhu, G. H. Lee, and Y. Fan, "Navrag: Generating user demand instructions for embodied navigation through retrieval-augmented llm," *arXiv preprint arXiv:2502.11142*, 2025.

[17] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[18] J. Hao, T. Yang, H. Tang, C. Bai, J. Liu, Z. Meng, P. Liu, and Z. Wang, "Exploration in deep reinforcement learning: From single-agent to multiagent domain," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[19] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "Vint: A foundation model for visual navigation," in *Conference on Robot Learning*. PMLR, 2023, pp. 711–733.

[20] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, "Motion planning diffusion: Learning and planning of robot motions with diffusion models," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 1916–1923.

[21] D. Shah, B. Osiński, S. Levine *et al.*, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Conference on robot learning*. PMLR, 2023, pp. 492–504.

[22] Y. Deng, J. Wang, J. Zhao, X. Tian, G. Chen, Y. Yang, and Y. Yue, "Opengraph: Open-vocabulary hierarchical 3D graph representation in large-scale outdoor environments," *IEEE Robotics and Automation Letters*, 2024.

[23] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 901–16 911.

[24] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[25] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," 2022.

[26] D. Tsai, S. Worrall, M. Shan, A. Lohr, and E. Nebot, "Optimising the selection of samples for robust lidar camera calibration," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 2631–2638.

[27] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.

[28] Y.-K. Lin, W.-C. Lin, and C.-C. Wang, "Asynchronous state estimation of simultaneous ego-motion estimation and multiple object tracking for lidar-inertial odometry," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 616–10 622.

[29] Y. Jia, T. Wang, F. Cao, X. Chen, S. Shao, and L. Liu, "Trlo: An efficient lidar odometry with 3-d dynamic object tracking and removal," *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–10, 2025.

[30] Z. Yan, X. Wu, Z. Jian, B. Lan, and X. Wang, "Rh-map: Online map construction framework of dynamic object removal based on 3D region-wise hash map structure," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1492–1499, 2023.

[31] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2997–3004.

[32] J. Zeng, Z. Li, and K. Sreenath, "Enhancing feasibility and safety of nonlinear model predictive control with discrete-time control barrier functions," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6137–6144.

[33] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "Gnm: A general navigation model to drive any robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 7226–7233.