

SuperMap: A Spatio-Temporal SLAM System for Visual-Language Navigation

Shibo Zhao[†], Guofei Chen[†], Honghao Zhu, Zhiheng Li, Changwei Yao,
Nader Zantout, Seungchan Kim, Wenshan Wang, Ji Zhang, and Sebastian Scherer

The Robotics Institute, Carnegie Mellon University

{guofei, shibo, basti}@andrew.cmu.edu

[†]Equal contribution

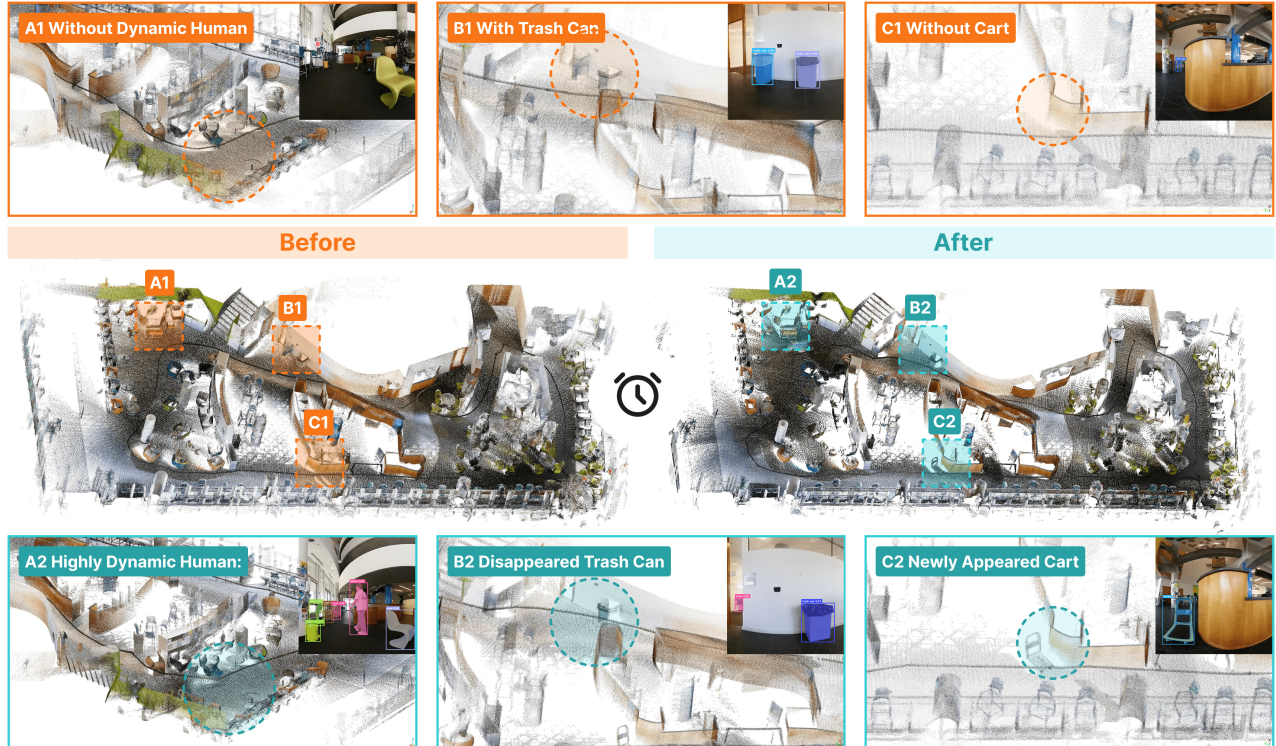


Fig. 1: We present SuperMap, a real-time, spatio-temporal SLAM system capable of instance-level object detection, tracking dynamic changes, and maintaining a scene graph for spatial reasoning. The figure shows how SuperMap detects short-term human movements (A1, A2) and long-term environmental changes (B1, B2, C1, C2), such as the disappearance of a trashcan and the appearance of a cart, demonstrating consistent spatio-temporal representation.

Abstract—Robotic navigation in human environments requires a spatio-temporal semantic representation that can reconcile open-vocabulary perception with long-term environmental changes. While foundation models provide strong zero-shot recognition, their predictions are intermittent and view-dependent, and naively integrating them into mapping pipelines leads to identity drift and stale semantics over time. We present SuperMap, a 4D spatio-temporal mapping framework for language-guided navigation that integrates high-frequency geometric SLAM with asynchronous open-vocabulary perception. Our core contribution is a consistency-driven mapping engine that combines 3D-aware instance association/re-activation with a principled existence-and-label confidence update to maintain stable object identities and prune outdated map content under occlusions and scene changes. SuperMap produces a queryable 4D scene-graph representation that interfaces naturally with Vision-Language Models by supporting compositional queries over object semantics, relations,

and history. We demonstrate SuperMap on benchmarks and real robots, including dynamic scenes with appearance/disappearance and relocation, and provide ablations and runtime analysis. We release the full system as open-source to provide the community with a deployable baseline for open-vocabulary spatio-temporal mapping. Project website: superodometry.com/supermap

I. INTRODUCTION

Robots deployed in human environments are increasingly expected to execute open-vocabulary navigation goals, e.g., “go to the monitor next to the whiteboard” or “return to the chair that was near the plant earlier.” Fulfilling such commands robustly requires more than accurate pose estimation and geometric reconstruction: the robot must maintain an object-centric spatio-temporal map representation that preserves in-

TABLE I: Capability Checklist for Semantic Mapping and Semantic SLAM Methods. ✓ indicates support, ✗ indicates limited or no support

Capability	Semantic Mapping						Semantic SLAM			
	HOV-SG[30]	ConceptGraphs[5]	RayFronts[1]	OpenScene[16]	OpenMask3D[28]	SeeGround[11]	Kimera[24]	OVO-SLAM[15]	Khronos[25]	SuperMap (Ours)
Real-Time	✗	✗	✓	✗	✗	✗	✓	✓	✗	✓
Open-Vocabulary	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓
Instance-Level	✓	✓	✗	✗	✗	✗	✗	✓	✗	✓
Short-Term	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
Long-Term	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
Scene Graph	✓	✓	✗	✗	✗	✗	✓	✗	✗	✓

stance identities, semantic attributes, and spatial relations over time, and that remains valid as the environment evolves. This requirement is difficult to meet in practice because real-world scenes are dynamic—objects may be occluded, relocated, removed, or newly introduced during operation. Consequently, systems that rely primarily on per-frame recognition or static semantic maps often degrade over long horizons: instance identities fragment, semantic content becomes stale, and the relational structure needed for language grounding is unavailable or inconsistent.

Recent progress in semantic SLAM and open-vocabulary perception has produced strong components—state estimation and dense reconstruction, 2D tracking, and open-vocabulary detection/segmentation [24, 10]. However, simply combining these modules rarely yields a map that is reliably usable for navigation. Many systems remain closed-vocabulary and class-level [24, 10], limiting instance identity tracking, while dynamic-SLAM methods often focus on short-term motion (e.g., moving people) and overlook long-term changes such as object relocation outside the robot’s view [18, 31, 27, 3]. In practice, missed detections and occlusions fragment identities, scene changes introduce stale map content, and most representations lack an efficient query interface for downstream tasks like language-guided navigation. As a result, there remains a gap between “semantic mapping outputs” and a robot-usable map representation that can be queried and acted upon online.

Recent scene graph approaches [5, 30, 14, 20] incorporate open-vocabulary detectors [23] to support natural language navigation, but they are time consuming (minutes to hours) to construct and not suitable for real-time mapping. Moreover, they struggle to capture dynamic relationships between objects, limiting their ability to model scene evolution.

In summary, existing approaches predominantly address either dynamic change detection (e.g., identifying moving humans or vehicles) or static object recognition (e.g., identifying objects and their relationships). However, *few methods tackle both challenges simultaneously: understanding which objects change and how they evolve spatially and temporally*. Addressing this dual challenge is crucial for robots to effectively perceive and adapt to dynamic real-world environments.

To meet this challenge, we introduce SuperMap: an open-source system that constructs a queryable open-vocabulary 4D scene memory online. We evaluate SuperMap on public benchmarks for open-vocabulary semantic and instance mapping, and on real-robot deployments in dynamic indoor environments that include object appearance/disappearance and long-horizon changes. We provide ablations to attribute mapping improvements to specific maintenance components

(association, updates, and graph construction), and runtime profiling to clarify which modules dominate compute. Finally, we demonstrate language-guided robot navigation driven by scene-graph queries, where structured 4D scene memory improves robust grounding compared to using raw detections or video-only inputs.

Our contributions are as follows:

- **Open-Vocabulary, Spatio-Temporal Semantic SLAM:** An online robotic system that builds a persistent, queryable open-vocabulary 4D scene memory suitable for downstream language-conditioned tasks.
- **Spatio-Temporal Object Tracking:** An online pipeline that integrates 2D–3D association, validation, and change-aware updates to maintain instance consistency under occlusions, partial observations, label variability, and scene change.
- **Instance-level Scene Graph:** Our 4D scene graph seamlessly incorporates spatial and temporal information for each object, equipping robots with instance-level spatio-temporal reasoning capabilities (e.g., locating moved objects, recalling past scenes).
- **Open-Source Framework:** We will release our change-detection benchmark, comprehensive ablations and runtime profiling, and the real-robot visual–language navigation pipeline to facilitate reproducible research in the robotics community.

II. RELATED WORKS

A. Semantic Mapping

Equipping mobile robots with the ability to perceive and understand their surroundings is crucial for real-world applications. To this end, researchers have developed semantic maps to capture environmental meaning and context [22]. However, many recent methods [16, 5] rely on offline processing, requiring pre-collected environmental data (images and depth) to build semantically labeled maps shown in Table I.

For example, OpenScene [16] uses OpenSeg [32] to extract CLIP embeddings [19] from images and then trains a scene-specific 3D model, but it operates offline, requiring a full scene scan. OpenMask3D [28] back-projects 2D object masks from multiple views to form 3D segments, while SeeGround [11] combines 2D-VLM with 3D geometric cues for instance segmentation. Despite improving 3D accuracy, these methods assume complete scene data, limiting online usability.

LERF [9] and LangSplat [17] employ radiance fields to build 3D maps with CLIP-based features, offering flexible semantic queries but lacking real-time capability due to computational demands. RayFronts [1] enables online semantic

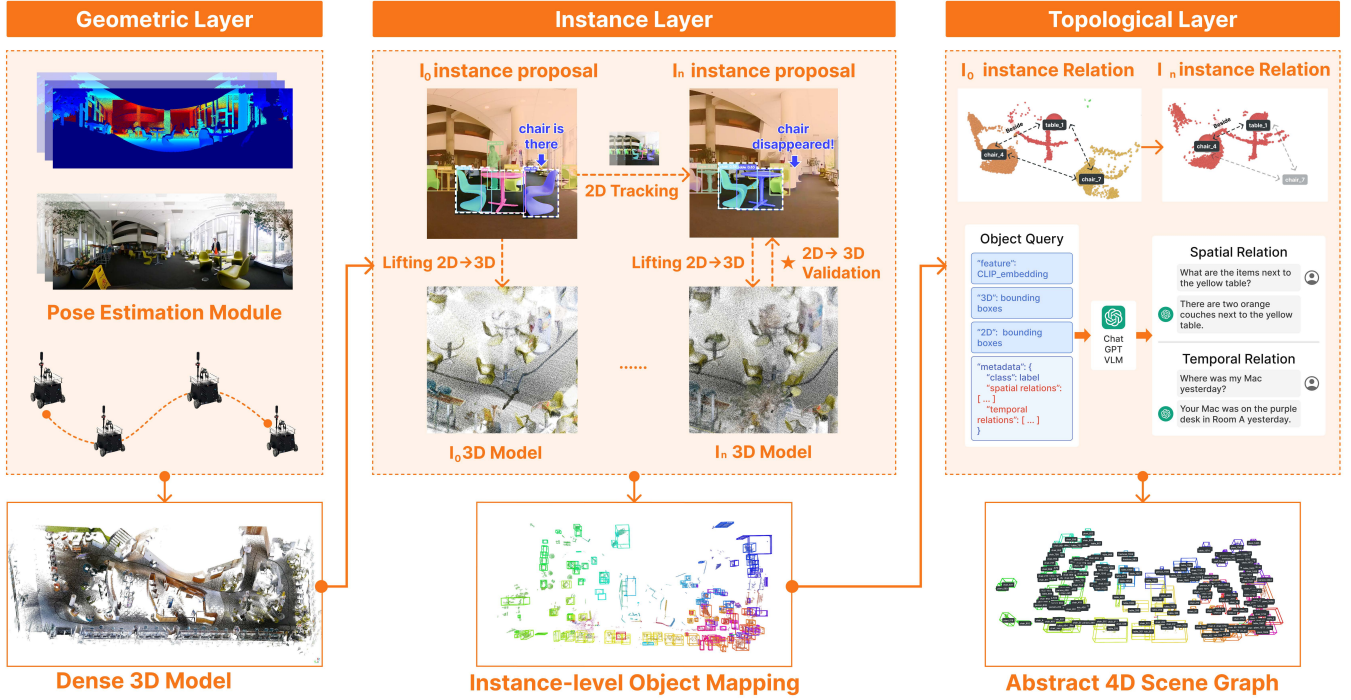


Fig. 2: **Overview.** The system delivers accurate state estimation together with a rich, multi-level map representation that spans from geometric details to abstract semantics. It consists of three main modules. First, an online 3D reconstruction module built on a LiDAR SLAM backbone performs state estimation and generates a dense 3D model of the environment. Next, the estimated state and reconstructed geometry are used to identify short-term and long-term object changes, enabling instance-level object mapping. Finally, the object-level representation is organized into an abstract 4D scene graph that captures both spatial and temporal relationships among objects.

mapping for outdoor environments but struggles with change detection and fails to achieve instance-level 3D semantics.

B. Semantic SLAM

To jointly estimate poses within the pipeline, we transform the mapping pipeline into a SLAM pipeline. In recent years, numerous SLAM systems have integrated semantic information during the mapping process. One notable example is Kimera [24], which offers a real-time semantic mapping solution by building a 3D scene graph. However, it relies on closed-set CNN segmentation, which limits its ability to generalize to novel classes. LiDAR-based SLAM approaches, such as SuMa++ [2] and LIOM [34], incorporate CNN-based per-point labels into LiDAR maps. Despite their effectiveness, these methods do not support open-vocabulary segmentation, restricting their adaptability to diverse environments. SlideSLAM [13] and OVO-SLAM [15] employ open-vocabulary detectors for instance segmentation, allowing for more flexible scene understanding. However, these methods face challenges when dealing with **long-term and short-term dynamic objects**.

C. Open-Vocabulary Scene Graph

HOV-SG [30], ConceptGraphs [5] and CLIO [14] segment and label objects from a full point cloud using SAM and CLIP, organizing them into a hierarchical graph with open-ended labels from CLIP. However, HOV-SG and ConceptGraphs require offline 3D reconstruction before applying SAM and

CLIP, making them unsuitable for online SLAM or real-time dynamic tasks. CLIO implements real-time open-vocabulary scene graph generation but lacks spatio-temporal capabilities.

D. Spatio-Temporal Semantic SLAM

Recently, Khronos [25] addresses both short-term and long-term dynamics in map representation. However, this method is limited to closed-set objects and lacks the capability for instance-level tracking. Additionally, Khronos struggles to operate in real time, making it unsuitable for robotic operations. To the best of our knowledge, SuperMap is the first real-time, spatio-temporal, open-vocabulary, and instance-level object mapping framework that effectively handles both short-term and long-term dynamic objects.

III. SPATIO-TEMPORAL SLAM PROBLEM

The spatio-temporal SLAM problem aims to provide an explicit scene representation that captures how object-level changes evolve across space and time through incremental data acquisition. We model the scene as a composition of objects, represented as a scene graph where nodes correspond to objects and edges denote spatio-temporal relationships.

In our setting, the system processes a continuous stream of either an RGB-D video sequence or a point-cloud video sequence, where RGB frames $C_{1:T} = \{C_1, C_2, \dots, C_T\}$ and depth(LiDAR) frames $D_{1:T} = \{D_1, D_2, \dots, D_T\}$ are given as input, and T is the sequence length and $t \in \{1, \dots, T\}$ is the time index. Camera poses $P_{1:T} = \{P_1, P_2, \dots, P_T\}$ must

be estimated. At each time t , the system outputs a consistent global map $\mathcal{M}_t$ consisting of semantic instance-level objects $\mathcal{O}_t = \{\mathcal{O}_t^j\}_{j=1}^{N_t}$, where j indexes objects and N_t is the number of objects at time t .

The spatio-temporal SLAM problem is defined as follows: given the current observation $Q_t = \{C_t, D_t\}$ and the existing map $\mathcal{M}_{t-1}$, estimate pose P_t , determine the instance IDs $\mathcal{I}_t$ and the updated map $\mathcal{M}_t$ by maximizing (or estimating)

$$P(\mathcal{I}_t, \mathcal{M}_t, P_t \mid \mathcal{M}_{t-1}, Q_t). \quad (1)$$

Applying the chain rule, we obtain

$$\begin{aligned} P(\mathcal{I}_t, \mathcal{M}_t, P_t \mid \mathcal{M}_{t-1}, Q_t) &= \underbrace{P(P_t \mid \mathcal{M}_{t-1}, Q_t)}_{\text{pose estimation (geometric)}} \\ &\times \underbrace{P(\mathcal{I}_t \mid \mathcal{M}_{t-1}, Q_t, P_t)}_{\text{spatio-temporal instance association}} \\ &\times \underbrace{P(\mathcal{M}_t \mid \mathcal{M}_{t-1}, Q_t, P_t, \mathcal{I}_t)}_{\text{online map update}}. \end{aligned} \quad (2)$$

This involves three tasks: pose estimation, spatio-temporal object instance association and the online map update [4].

IV. METHOD

The architecture of our spatio-temporal SLAM system, SuperMap, is shown in Figure 2. Given synchronized depth measurements, IMU data, and RGB images, SuperMap runs fully onboard to jointly estimate the robot state and maintain an instance-level spatio-temporal map representation as the environment evolves. We describe three main components: (1) *Online 3D Reconstruction* for dense geometric mapping, (2) *Spatio-Temporal Object Updates* for robust instance association, and (3) *Spatio-Temporal Scene Graph Construction* that exposes the map as a queryable interface for VLN.

A. Geometric Layer: Online 3D Reconstruction

We use SuperOdometry [35] to obtain the robot pose P_t and a colorized dense 3D reconstruction from image, depth/LiDAR, and IMU streams. The observation Q_t requires accurate camera poses P_t and depth observations D_t . For each time t , we estimate the robot's state in $SE(3)$, denoted as $\mathbf{T}_{WB}^{(t)}$, where W is the world frame and B is the body frame. Given a fixed extrinsic calibration $\mathbf{T}_{BC}$, the camera pose P_t is derived as:

$$P_t = \mathbf{T}_{WC}^{(t)} = \mathbf{T}_{WB}^{(t)} \cdot \mathbf{T}_{BC} \quad (3)$$

This estimated pose and the synchronized depth D_t form the observation Q_t . These geometric priors are critical for: (i) back-projecting 2D semantic observations into the 3D world, (ii) performing motion compensation for temporal tracking, and (iii) maintaining global map consistency. By anchoring all subsequent instance-level detections to this metric foundation, we ensure that the resulting spatio-temporal map is physically grounded and suitable for downstream navigation tasks.

B. Instance Layer: Spatio-Temporal Instance Association

For each image frame we first infer the 2D instance-level detection and segmentations using vision models [12] [21], then address the estimation of instance IDs $\mathcal{I}_t$ by solving $P(\mathcal{I}_t \mid \mathcal{M}_{t-1}, Q_t)$. The objective is to assign a unique instance ID to 2D detections by leveraging 3D spatial consistency.

1) *Hybrid Tracking State*: We define the tracklet state $\mathbf{S}_i(t) \in \mathbb{R}^6$ (or $\mathbb{R}^8$ if including scale velocity) as a decoupled vector:

$$\mathbf{S}_i(t) = [\mathbf{c}_i(t)^\top, \mathbf{s}_i(t)^\top, \dot{\mathbf{c}}_i(t)^\top]^\top \quad (4)$$

where $\mathbf{c}_i(t) = [x, y]^\top$ is the 2D image centroid, $\mathbf{s}_i(t) = [w, h]^\top$ represents the bounding box width and height, and $\dot{\mathbf{c}}_i(t)$ is the translational velocity in the image plane.

2) *3D-to-2D Motion-Compensated State*: Since standard 2D trackers (e.g., ByteTrack [33]) struggle with rapid ego-motion. We resolve this by defining a hybrid tracking state $\mathbf{S}_i(t)$ that is updated via 3D-to-2D Motion Compensation.

Instead of a linear motion model, we derive the predicted centroid $\hat{\mathbf{c}}_i(t)$ by projecting the 3D centroid $\mathbf{X}_i$ of the instance in $\mathcal{M}_{t-1}$ using the current pose P_t :

$$\hat{\mathbf{c}}_i(t) = \pi(\mathbf{K} \cdot P_t^{-1} \cdot \mathbf{X}_i) \quad (5)$$

The Kalman Filter state transition uses this projection as the prior:

$$\hat{\mathbf{S}}_i(t) = \mathbf{F}\mathbf{S}_i(t-1) + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}) \quad (6)$$

where $\mathbf{F}$ is the state transition matrix and $\mathbf{w}_t$ represents process noise. This projection-based prior allows the system to maintain identity through occlusions and aggressive robot maneuvers.

3) *Probabilistic Geometric Consistency Update*: The observation Q_t updates the map $\mathcal{M}_t$ by evaluating the posterior $P(\mathcal{M}_t \mid \mathcal{M}_{t-1}, Q_t, \mathcal{I}_t)$. We model the depth observation for each point $\mathbf{X}_k \in \mathcal{M}_{t-1}$ as a random variable.

Let $d_{proj} = \|\mathbf{T}_{CW}\mathbf{X}_k\|_z$ be the expected (projected) depth of the point in the current camera frame, and $D(\mathbf{u})$ be the raw sensor depth at the projected pixel $\mathbf{u} = \pi(\mathbf{X}_k)$. The depth residual is defined as:

$$\Delta d = d_{proj} - D(\mathbf{u}) \quad (7)$$

We model the measurement likelihood under the assumption of Gaussian sensor noise, $p(\Delta d) \sim \mathcal{N}(0, \sigma^2)$. The occupancy state o_k of the map point is updated via a log-odds formulation to ensure numerical stability and recursive update capability:

$$L(o_k \mid Q_{1:t}) = L(o_k \mid Q_{1:t-1}) + \text{logit}(P(o_k \mid Q_t)) \quad (8)$$

where $L(n) = \log \frac{n}{1-n}$. The inverse sensor model $P(o_k \mid Q_t)$ is determined by the thresholded geometric consistency of Δd :

$$s_k^{(t)} = \begin{cases} \text{Observable} & \text{if } |\Delta d| \leq \tau_\epsilon \\ \text{Unobservable} & \text{if } \Delta d > \tau_\epsilon \quad (\text{Point behind surface}) \\ \text{Disappeared} & \text{if } \Delta d < -\tau_\epsilon \quad (\text{Point in front of surface}) \end{cases} \quad (9)$$

Points classified as dynamic are penalized in the log-odds update, effectively pruning transient or moved objects from the global map to maintain a consistent representation of the environment. This enables the change detection in dynamic scenes, and also provides geometric cue for fusing the semantic inference across frames.

4) *Bayesian Semantic Fusion*: To handle the uncertainty in the output of vision models, each instance maintains a categorical distribution $P(L_j = c)$. Given observations $z_{1:t}$ from an open-vocabulary detector, the posterior belief is:

$$P(L_j = c \mid z_{1:t}) = \eta \cdot P(z_t \mid L_j = c) \cdot P(L_j = c \mid z_{1:t-1}) \quad (10)$$

where $P(z_t \mid L_j = c)$ is the detector’s confusion matrix.

For observable points in (9), we update the semantic label using (10), and consequently remove object points whose posterior belief is too small. This fusion suppresses transient misclassifications.

C. Topological Layer: Spatio-Temporal Scene Graph

The map $\mathcal{M}_t$ is abstracted into a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}_s, \mathcal{E}_t)$. Nodes $\mathcal{V}$ represent object instances.

- **Spatial Edges ($\mathcal{E}_s$)**: Established via class-dependent geometric predicates. For objects A, B : $\text{On}(A, B) \iff (z_{min}^A \approx z_{max}^B) \wedge (\text{IoU}_{xy}(\mathcal{B}_A, \mathcal{B}_B) > \gamma)$.
- **Temporal Edges ($\mathcal{E}_t$)**: Link nodes across time based on $\mathcal{I}_t$ results, tracing object trajectories.

For real-time operation, we first cluster objects based on their centroid distances and then add a spatial edge when an object pair satisfies class-dependent geometric predicates (e.g., on, beside, under). The temporal edges trace an object’s trajectory from our object association result, preserving its identity throughout the sequence. The resulting graph compactly captures both instantaneous layout and long-term dynamics, providing an abstract representation of the scene that is easy to reason on. Full construction details appear in the supplementary material.

D. Visual Language Navigation via 4D Scene Graphs

The 4D scene graph $\mathcal{G}$ provides a high-level interface between raw sensor data and the symbolic inputs required by Vision–Language Models (VLMs). By delegating geometric detail to the spatio-temporal map, the VLM can perform zero-shot reasoning over long-term scene dynamics without processing raw point clouds or video streams.

a) *Spatio-Temporal Serialization*: We serialize a local subgraph into a structured text representation for linguistic reasoning. Each node $v \in V$ is encoded by its instance ID, semantic label, and 3D centroid $\mathbf{X}_v \in \mathbb{R}^3$. Spatial and temporal edges (E_S, E_T) are represented as symbolic relations (e.g., *Instance_1 on Instance_2*). Reliability is improved by (1) prepending a schema that defines the graph structure and coordinate frame, and (2) injecting explicit spatio-temporal cues that encourage trajectory- and relation-aware reasoning.

b) Grounded Actuation and Waypoint Generation:

To convert VLM outputs into deterministic control signals, the model must return target instance IDs within `<answer></answer>` tags. A parser extracts these IDs and retrieves their 3D centroids $\mathbf{X}_j$ from the map $\mathcal{M}_t$, which are then used as navigation waypoints. This process translates high-level language instructions into precise, grounded robot actions.

V. EXPERIMENTS

A. Experiment Settings

a) **Hardware Configuration**: We evaluate SuperMap on a custom-built mecanum-wheeled mobile platform designed for agile indoor navigation. The sensor suite comprises a Livox Mid-360 LiDAR, which provides a $360^\circ \times 59^\circ$ field of view, and a panoramic camera providing 360° RGB coverage. We implement a software-based timestamp alignment to associate visual and LiDAR streams. All computations, including LVI-odometry, instance tracking, 4D graph maintenance, and VLM interfacing, run onboard in real-time on an Intel i9-14900H CPU and an NVIDIA RTX 4090 Laptop GPU (16GB VRAM). We aim to address the following research questions through our experiments:

- **Semantic Quality**: How does SuperMap’s incremental object-centric mapping compare to state-of-the-art offline and online 3D semantic mapping approaches on the ScanNet benchmark (Sec. V-B)?
- **Spatio-Temporal Consistency**: Can the 4D graph accurately detect scene changes (appearances/disappearances) while maintaining persistent identities for static objects over long-duration trajectories (Sec. V-C)?
- **Grounded Reasoning**: Does the structured scene graph representation improve the accuracy and token efficiency of VLM reasoning comparing to video input (Sec. V-F)?
- **Online Visual-language Navigation**: Can the integrated pipeline support end-to-end, zero-shot, language-guided navigation using solely onboard compute (Sec. V-G)?

B. Semantic Quality: Quantitative Results for Class-level and Instance-level Segmentation

To evaluate the semantic fidelity of SuperMap, we conduct a comparative analysis against state-of-the-art open-vocabulary mapping frameworks, focusing on both class-level precision and instance-level consistency.

a) *Class-level Segmentation*: Table II summarizes the performance on the ScanNet benchmark. SuperMap achieves a competitive accuracy of **55.48%**, outperforming established object-centric baselines such as *ConceptGraphs* [5] and *ConceptFusion* [7]. While our accuracy is slightly lower (1.32%) than the point-feature fusion method *RayFronts* [1], SuperMap maintains a significantly lower computational footprint, making it more suitable for high-rate robotic feedback loops. These results demonstrate its effectiveness in open-vocabulary segmentation among works on object-centric mapping.

TABLE II: **Class-level Segmentation Benchmarking on ScanNet.** Best results are highlighted as **first** and **second**.

Method	Mapping Approach	Without Background			With Background		
		mIoU (%)	f-mIoU (%)	Acc (%)	mIoU (%)	f-mIoU (%)	Acc (%)
ConceptFusion [7]	point-feature	21.76	26.71	34.10	18.57	23.06	28.77
NACLIP-3D [6]		22.32	24.32	33.46	22.32	24.32	33.46
Trident-3D [26]		29.97	37.62	51.06	24.80	27.77	38.43
RayFronts [1]		41.29	46.42	56.76	32.29	39.04	49.15
ConceptGraphs [5]	object-level	21.62	24.32	31.05	20.83	23.61	35.80
HOV-SG [30]		26.79	36.05	35.17	23.48	28.92	38.52
SuperMap (Ours)		27.42	43.50	55.48	22.61	29.10	33.00

TABLE III: **Instance-level Segmentation Benchmark on ScanNet.** Best results are highlighted as **first** and **second**.

Method	Chair		Window		Refrigerator		Sofa		Door	
	mAP ₅₀	mAP ₂₅	mAP ₅₀	mAP ₂₅	mAP ₅₀	mAP ₂₅	mAP ₅₀	mAP ₂₅	mAP ₅₀	mAP ₂₅
HOV-SG [30]	4.58	4.73	0.00	0.00	0.00	0.00	30.00	31.25	9.70	10.40
ConceptGraphs [5]	0.00	2.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SuperMap (Ours)	63.76	74.72	42.20	67.92	62.50	62.50	33.35	83.35	10.00	25.00

b) *Instance-level Segmentation*: Unlike offline frameworks that rely on global scene context, SuperMap targets online robotic deployment, where incremental data fusion and zero-shot generalization are essential. This setting requires establishing object correspondences from partial, streaming observations in real time. To enable zero-shot perception, we integrate GroundingDINO [12] for language-grounded detection and SAM2 [21] for robust instance segmentation. We report comparative results against *HOV-SG* [30] and *ConceptGraphs* [5] in Table III. SuperMap significantly outperforms these baselines, particularly for discrete objects contained within the camera’s instantaneous field of view. This gain demonstrates the efficacy of our 3D-aware tracking-by-detection approach over methods relying on point-feature clustering or over-segmented geometries.

C. Spatio-Temporal Consistency on Qualitative Evaluation

We evaluate SuperMap’s ability to maintain map integrity over extended durations while identifying environmental changes. In a 10-minute real-world experiment (30m × 20m indoor area), we introduced structural changes: three objects were removed (a plant, a trash can, and a chair) and three were added (a bucket, a cart, and a safety sign).

a) *Qualitative Performance*: Fig. 3 qualitatively demonstrates SuperMap’s long-horizon instance consistency under both disappearance and appearance events. The top row shows three objects that disappear from the scene. In all cases, the instance IDs remain consistent over time, indicating that the system preserves object identity even when objects are no longer observed.

For example, the top row shows three objects that newly appear in the scene. In the top-left example, we place a new bucket instance (**bucket 1964**) between **Chair 155** and **Chair 203**. In the bottom-left example, **Chair 7** disappears between **table 1** and **chair 4**. Nonetheless, the small ID label in the Rerun visualization remains unchanged, showing that the system maintains the same instance association before

and after the disappearance event. It demonstrates consistent object tracking and stable identity maintenance over long time horizons.

D. Spatial-Temporal Consistency on Quantitative Evaluation

We quantitatively assess SuperMap using six target objects (Fig. 3) against manually annotated 3D bounding boxes and semantic labels. We define two primary metrics:

- **Object Detection Recall**: Measured during the *appearance interval* (when the object is present). A True Positive (TP) requires a 3D IoU > 0.1, centroid distance < 0.3m, and the correct semantic label.
- **Change Detection Recall**: Evaluates the system’s ability to detect an object during its appearance and confirm its removal during the *disappearance interval*. A TP requires detection in the former and zero false-positive detections in the latter.

As shown in Table IV, DualMap [8] achieves near-zero recall for object detection due to unstable 2D segmentation and inconsistent 3D box estimation, which causes most instances to be filtered out by its mapping module. While it maintains a nominal ~ 0.5 recall for change detection, this is a mathematical artifact: the method “succeeds” during the disappearance interval only because it failed to detect the object initially. Meanwhile, Khronos fails to produce consistent 3D estimation results (details shown in Supplementary) due to latent semantic inference bottlenecks that cause significant frame drops and degraded semantic mask quality. In contrast, SuperMap maintains high recall across both metrics, demonstrating robust instance detection.

E. SuperMap System Ablation Study

We evaluated the contribution of our individual sub-modules, specifically the 2D tracking, geometric consistency update, and Bayesian semantic fusion. Using a scene with 41 annotated objects, we quantified the performance of the spatio-temporal object map using precision, recall, and F1 scores.

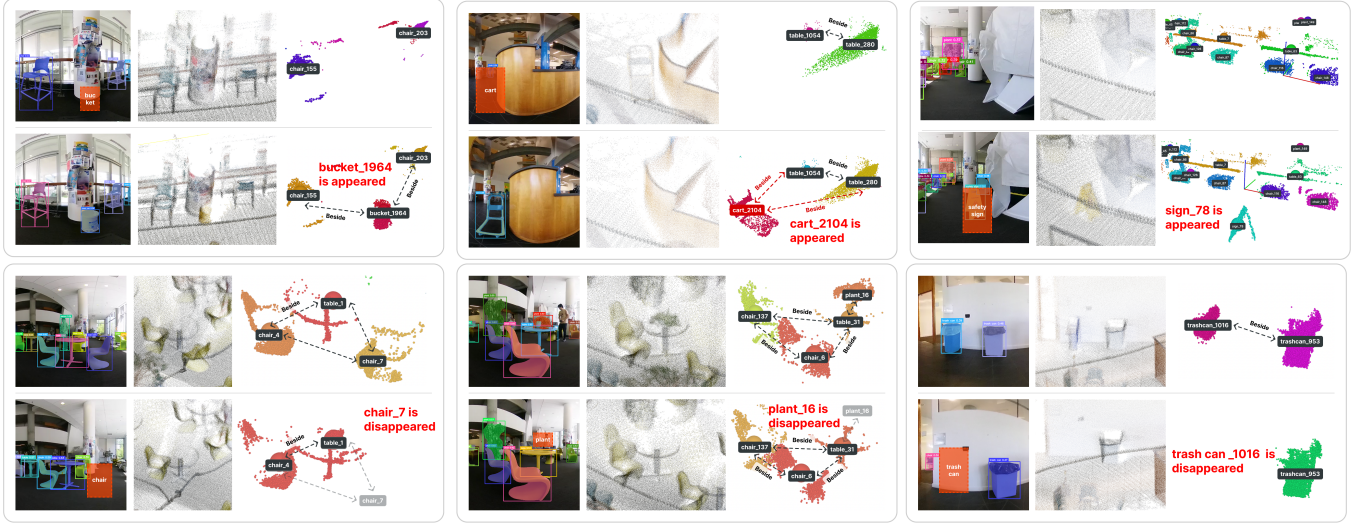


Fig. 3: **Spatio-temporal Consistency.** For each example (left to right), we show the 2D detections, the dense 3D reconstruction, and the object-centric map rendered in Rerun *before* and *after* the scene change. **Top:** three appearance events (bucket, cart, safety sign), where newly introduced objects receive new IDs while nearby objects retain their original IDs, demonstrating long-horizon re-identification and stable instance association in 3D. **Bottom:** three disappearance events (chair, plant, trash can), where the same instance ID is preserved across time despite the object leaving the field of view.

TABLE IV: **Spatial-Temporal Change Detection Performance.**

Method	Task	Appeared			Disappeared		
		Buc.	Cart	Sign	Plant	Trash	Chair
Khronos [25]	Detect.	—	—	—	—	—	—
DualMap [8]		0.000	0.000	0.000	0.000	0.310	0.000
SuperMap (ours)		1.000	0.262	0.583	0.755	0.434	1.000
Khronos [25]	Change	—	—	—	—	—	—
DualMap [8]		0.553	0.527	0.507	0.561	0.642	0.449
SuperMap (ours)		1.000	0.622	0.790	0.865	0.679	1.000

TABLE V: **Performance Comparison of Ablation Study**

Configuration	Precision	Recall	F1
W/o 2D Tracker	0.7787	0.4595	0.5780
W/o Semantic Fusion	0.7929	0.3870	0.5201
W/o Geometric Consistency Update	0.8189	0.4448	0.5764
All (proposed)	0.8677	0.4955	0.6308

As shown in Table V, the full system outperforms all baseline configurations across every metric, yielding the most accurate object map. This improvement stems from two primary factors. The 2D tracker maintains consistent identity association during significant viewpoint changes to prevent object fragmentation. Additionally, the integration of geometric and semantic fusion allows the system to remain resilient against momentary detection failures. By re-evaluating past inferences against new evidence over multiple frames, the map effectively self-corrects and filters out detection noise.

F. Grounding Reasoning

We evaluate the 4D scene graph as a grounding engine for Large Vision-Language Models (VLMs) using Gemini

2.0 Flash [29]. We compare reasoning performance between structured Scene Graph Input (serialized predicates) and raw Video Input (RGB sequences). As shown in Fig. 4, the 4D scene graph provides superior accuracy for complex queries:

- **Spatial Logic (Prompt 1):** Given a fire emergency scenario, the VLM utilizes the graph’s metric edges to disambiguate a specific “extinguisher” relative to a “cone” and “plant.” While raw video suffers from perspective distortion and object confusion, the scene graph enables precise distance calculation and instance retrieval.
- **Temporal Logic (Prompt 2):** When tracing a “bag’s” past trajectory to find a dropped item, the VLM traverses the graph’s temporal edges E_T . The structured history allows for filtered, timestamped retrieval of dynamic states. Conversely, video-based reasoning relies on unreliable “common-sense” heuristics and suffers from hallucinations as temporal depth increases.

G. Visual Language Navigation

We evaluate the downstream utility of SuperMap through zero-shot, language-guided navigation in previously unseen environments. Using the real-time 4D scene graph, the robot navigates reliably based solely on spatial relationships among 3D objects.

a) *Onboard Grounding and Execution:* During autonomous exploration, the robot incrementally builds an instance-level semantic map. When given a natural language command (e.g., “Go to the whiteboard next to the painting, then go to the painting next to the fridge”), the VLM interprets the spatial relations encoded in the 4D scene graph to resolve ambiguity and outputs precise 3D goal coordinates for the navigation stack.

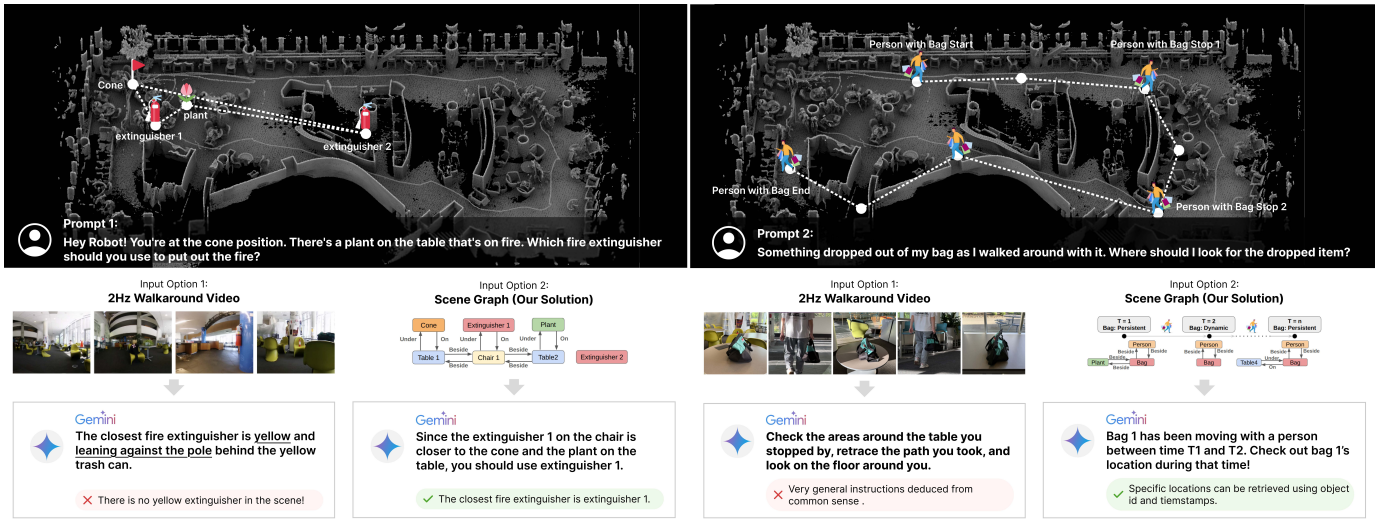


Fig. 4: **Comparison of Reasoning Results using Video Input and Scene-graph Input.** Compared to raw video, the scene-graph representation yields more reliable spatial reasoning, clearer temporal inference, and more accurate instance-level object retrieval.

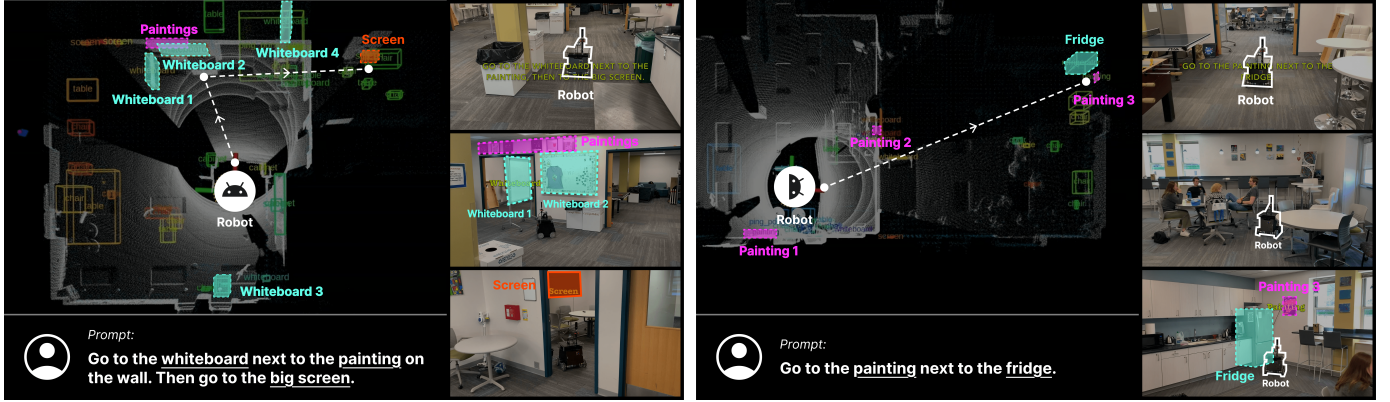


Fig. 5: **Online Visual-Language Navigation using Scene Graphs.** Our system enables robots to understand and reason about the spatial relationships in language instructions for VLN. See the supplementary video for more details.

b) Experimental Validation: As shown in Fig. 5, SuperMap supports robust navigation in visually ambiguous environments:

- **Spatial ordering :** The robot selects the correct target among four visually identical whiteboards by reasoning over their relative spatial configuration in the graph (left subfigure).
- **Relational retrieval:** The robot identifies the relationship between the fridge and the painting to reach the intended adjacent destination (right subfigure).

H. Runtime and Memory

To assess the computational efficiency of our system, we measure the throughput of each module. The pose estimation module maintains a consistent 10 Hz output, while 2D instance segmentation operates at 1 Hz due to the heavy inference requirements of semantic segmentation. The remainder of the architecture performs 3D mapping and 4D scene graph updates at 3 Hz and 5 Hz, respectively.

VI. CONCLUSION

We present SuperMap, a real-time, open-vocabulary SLAM system that constructs and maintains an instance-level 4D semantic map. By integrating geometric constraints from SLAM with semantic features from 2D foundation models, SuperMap performs joint 2D-3D object tracking, instance validation, and change detection online. This unified pipeline ensures spatio-temporal consistency despite occlusions, partial observations, or dynamic environmental changes. The resulting 4D scene graph provides an explicit, continuously updated world model, empowering robots with long-term memory, robust spatio-temporal reasoning, and language-guided navigation.

VII. LIMITATION

While SuperMap provides a robust framework for spatio-temporal mapping, its performance in tracking highly dynamic objects remains limited. Future iterations could address this by incorporating specialized tracking-by-detection modules or efficient segmentation-based tracking to improve temporal coherence. Furthermore, our current pipeline relies on a pre-defined list of object prompts for open-vocabulary 2D detec-

tion. Integrating an automated prompting mechanism based on open-world object discovery would significantly enhance the system’s adaptability, allowing it to function in truly novel environments without prior semantic knowledge.

REFERENCES

- [1] Omar Alama, Avigyan Bhattacharya, Haoyang He, Seungchan Kim, Yuheng Qiu, Wenshan Wang, Cherie Ho, Nikhil Keetha, and Sebastian Scherer. Rayfronts: Open-set semantic ray frontiers for online scene understanding and exploration. *arXiv preprint arXiv:2504.06994*, 2025.
- [2] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguere, Jens Behley, and Cyrill Stachniss. Suma++: Efficient lidar-based semantic slam. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4530–4537. IEEE, 2019.
- [3] Linyan Cui and Chaowei Ma. Sof-slam: A semantic visual slam for dynamic environments. *IEEE access*, 7: 166528–166539, 2019.
- [4] Yinan Deng, Bicheng Yao, Yihang Tang, Yi Yang, and Yufeng Yue. Openvox: Real-time instance-level open-vocabulary probabilistic voxel representation. *arXiv preprint arXiv:2502.16528*, 2025.
- [5] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE, 2024.
- [6] Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. Pay attention to your neighbours: Training-free open-vocabulary semantic segmentation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5061–5071. IEEE, 2025.
- [7] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023.
- [8] Jiajun Jiang, Yiming Zhu, Zirui Wu, and Jie Song. Dualmap: Online open-vocabulary semantic mapping for natural language navigation in dynamic changing scenes. *IEEE Robotics and Automation Letters*, 10(12), 2025.
- [9] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lurf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023.
- [10] Ghanta Sai Krishna, Kundrapu Supriya, and Sabur Baidya. 3ds-slam: A 3d object detection based semantic slam towards dynamic indoor environments. *arXiv preprint arXiv:2310.06385*, 2023.
- [11] Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- [12] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024.
- [13] Xu Liu, Jiuzhou Lei, Ankit Prabhu, Yuezhao Tao, Igor Spasojevic, Pratik Chaudhari, Nikolay Atanasov, and Vijay Kumar. Slideslam: Sparse, lightweight, decentralized metric-semantic slam for multi-robot navigation. *arXiv preprint arXiv:2406.17249*, 2024.
- [14] Dominic Maggio, Yun Chang, Nathan Hughes, Matthew Trang, Dan Griffith, Carlyn Dougherty, Eric Cristofalo, Lukas Schmid, and Luca Carlone. Clio: Real-time task-driven open-set 3d scene graphs. *IEEE Robotics and Automation Letters*, 2024.
- [15] Tomas Berriel Martins, Martin R Oswald, and Javier Civera. Ovo-slam: Open-vocabulary online simultaneous localization and mapping. *arXiv preprint arXiv:2411.15043*, 2024.
- [16] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 815–824, 2023.
- [17] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024.
- [18] Yuheng Qiu, Chen Wang, Wenshan Wang, Mina Henein, and Sebastian Scherer. Airdos: Dynamic slam benefits from articulated objects. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8047–8053. IEEE, 2022.
- [19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmlR, 2021.
- [20] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. *arXiv preprint arXiv:2307.06135*, 2023.
- [21] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.
- [22] Sonia Raychaudhuri and Angel X Chang. Semantic mapping in indoor embodied ai—a comprehensive survey

- and future directions. *arXiv preprint arXiv:2501.05750*, 2025.
- [23] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.
 - [24] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696. IEEE, 2020.
 - [25] Lukas Schmid, Marcus Abate, Yun Chang, and Luca Carlone. Khronos: A unified approach for spatio-temporal metric-semantic slam in dynamic environments. *arXiv preprint arXiv:2402.13817*, 2024.
 - [26] Yuheng Shi, Minjing Dong, and Chang Xu. Harnessing vision foundation models for high-performance, training-free open vocabulary segmentation. *arXiv preprint arXiv:2411.09219*, 2024.
 - [27] Seungwon Song, Hyungtae Lim, Alex Junho Lee, and Hyun Myung. Dynavins: A visual-inertial slam for dynamic environments. *IEEE Robotics and Automation Letters*, 7(4):11523–11530, 2022.
 - [28] Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. *arXiv preprint arXiv:2306.13631*, 2023.
 - [29] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
 - [30] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024.
 - [31] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Qiao Fei. Ds-slam: A semantic visual slam towards dynamic environments. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1168–1174. IEEE, 2018.
 - [32] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution vision transformer for dense predict. *Advances in neural information processing systems*, 34:7281–7293, 2021.
 - [33] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*, pages 1–21. Springer, 2022.
 - [34] Shibo Zhao, Zheng Fang, HaoLai Li, and Sebastian Scherer. A robust laser-inertial odometry and mapping method for large-scale highway environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1285–1292. IEEE, 2019.
 - [35] Shibo Zhao, Hengrui Zhang, Peng Wang, Lucas Nogueira, and Sebastian Scherer. Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8729–8736. IEEE, 2021.