LOST-3DSG: Lightweight Open-Vocabulary 3D Scene Graphs with Semantic Tracking in Dynamic Environments

Sara Micol Ferraina[*,1] Michele Brienza[*,1] Francesco Argenziano[1] Emanuele Musumeci[1]
Vincenzo Suriani[1] Domenico D. Bloisi[2] Daniele Nardi[1]

[1]Sapienza University of Rome, Rome, Italy
[2]International University of Rome UNINT, Rome, Italy
[*]Authors contributed equally

ferraina.1857726@studenti.uniroma1.it
{brienza, argenziano, musumeci, suriani, nardi}@diag.uniroma1.it
domenico.bloisi@unint.eu

Abstract

Tracking objects that move within dynamic environments is a core challenge in robotics. Recent research has advanced this topic significantly; however, many existing approaches remain inefficient due to their reliance on heavy foundation models. To address this limitation, we propose LOST-3DSG, a lightweight open-vocabulary 3D scene graph designed to track dynamic objects in real-world environments. Our method adopts a semantic approach to entity tracking based on word2vec and sentence embeddings, enabling an open-vocabulary representation while avoiding the necessity of storing dense CLIP visual features. As a result, LOST-3DSG achieves superior performance compared to approaches that rely on high-dimensional visual embeddings. We evaluate our method through qualitative and quantitative experiments conducted in a real 3D environment using a TIAGo robot. The results demonstrate the effectiveness and efficiency of LOST-3DSG in dynamic object tracking. Code and supplementary material will be publicly released upon acceptance, in order to comply with anonymity requirements.

1. Introduction

The ability to reconstruct and represent the surrounding environment is a fundamental requirement for autonomous robots. This task is inherently challenging due to the need to recognize objects across different scales, handle duplicate instances, and remain robust to varying lighting conditions. The problem becomes even more complex in dynamic settings, where objects may change position or appearance over time. Without an accurate and continuously updated model of the world, a robot's capacity to reason, plan, and act safely and effectively is severely constrained.

Humans naturally build internal representations of their environment and continuously refine them as the world evolves. This process involves tracking entities over time, understanding their motion, and maintaining semantic consistency despite changes. Replicating these capabilities in robots remains a significant challenge, particularly in real-world scenarios characterized by partial observability and frequent environmental changes.

Recent work on scene representation has increasingly adopted 3D Scene Graphs (3DSGs) [3, 29] as a flexible and expressive framework for modeling complex environments. By integrating geometric structure with semantic information, 3DSGs describe scenes as collections of object-centric nodes augmented with attributes and relational edges. This abstraction shifts scene understanding from low-level geometry to a structured, object-level representation that explicitly captures entities and their relationships.

Within this line of research, open-vocabulary 3DSGs [10, 15, 19, 32] have gained significant attention. These approaches leverage large pre-trained Foundation Models (FMs), including Vision-Language Models (VLMs) and CLIP [25], to encode object nodes with rich semantic representations that are not limited to a predefined taxonomy. As a result, robots can recognize and reason about previously unseen objects and concepts, substantially improving generalization in unstructured environments.

Despite these advantages, such expressiveness often comes at a significant computational and memory cost. Many existing methods rely on storing dense CLIP embeddings at the voxel or point level, producing large-scale

semantic maps that are expensive to construct, update, and maintain [10, 33]. These costs are intensified in dynamic environments, where frequent updates are necessary to reflect changes in object pose and state. The repeated extraction, storage, and management of high-dimensional semantic features ultimately limit scalability and real-time applicability, highlighting the need for more efficient representations that preserve open-vocabulary reasoning while reducing computational overhead.

We propose *LOST-3DSG*, a lightweight open-vocabulary 3D scene graph designed for dynamic environments. In contrast to existing approaches that store dense CLIP embeddings for objects in the scene, LOST-3DSG relies on low-cost *word2vec* [20] embeddings derived from semantic attributes extracted using a VLM. These compact semantic representations enable robust tracking of dynamic objects over time by reasoning at the attribute level. Rather than depending solely on geometric consistency, the system determines whether an observation corresponds to a previously seen object or a new instance by matching its semantic attributes. For example, if an object previously identified as a `"red and brown, wooden and metal hammer"` reappears at a different location, it is more likely the same object that has moved rather than a newly observed one. By benchmarking our method against CLIP-based approaches and conducting an extensive ablation study, we demonstrate that our system can accurately track objects in the scene while maintaining a low computational footprint. To further validate the proposed approach, we deploy it in a real-world environment using a TIAGo robot[1]. We summarize our contributions as follows:

- *LOST-3DSG*, a lightweight open-vocabulary 3D scene graph tailored for dynamic environments;
- an efficient semantics-based tracking algorithm that updates the 3DSG as previously observed objects move or temporarily disappear;
- an extensive experimental evaluation, including comparisons with CLIP-based methods, ablation studies, and real-world deployment on a TIAGo robot, demonstrating accurate object tracking with a limited computational footprint.

The remainder of this paper is organized as follows. Section 2 reviews related work on 3DSGs and scene representations for dynamic environments. Section 3 describes the proposed approach in detail. Sections 4 and 5 present the experimental setup and the obtained results, which are then discussed in Section 6. Finally, Section 7 draws the conclusions and outlines directions for future work.

[1] https://pal-robotics.com/robot/tiago/

2. Related Work

3D Scene Graphs In recent years, 3DSGs [3, 13, 29] have emerged as a prominent and widely adopted representation for modeling 3D environments. These representations describe a scene as a graph in which objects are encoded as nodes, while semantic, spatial, and functional relationships between them are captured through edges. This structured formulation provides a compact yet expressive way to model the entities present in an environment and to reason about their interactions. A key strength of 3DSGs lies in their flexibility. By tailoring node attributes and relational edges to the requirements of a given application, 3DSGs can support a broad range of downstream robotic tasks, including navigation [34], task planning [18, 22, 26], manipulation [12, 14, 27], and human-robot interaction [4, 17]. Moreover, their graph-based structure makes them easily serializable in formats such as JSON, facilitating their integration into LLM-augmented applications [6, 9, 26].

Open-Vocabulary Scene Understanding Recent advances in computer vision, together with the development of Large Language Models (LLMs) and other FMs, have enabled open-vocabulary scene understanding and parsing [37]. These approaches allow scenes to be analyzed and objects to be detected without being constrained by a fixed, predefined taxonomy of labels, thereby improving generalization to novel and previously unseen concepts. A common strategy to achieve open-vocabulary understanding in 3D environments is to augment existing scene representations with semantic features extracted from pre-trained models such as CLIP [25]. By associating CLIP embeddings with elements of a scene representation, whether voxels, points, primitives, or object nodes, it becomes possible to endow a wide range of mapping paradigms with rich, open-vocabulary semantics, including NeRF [8], Gaussian splatting [11], point clouds [24], and 3DSGs [15]. In doing so, however, open-vocabulary 3DSGs such as ConceptGraphs [10] and DovSG [33] require storing high-dimensional semantic features for each voxel in the scene, resulting in a queryable representation at the expense of a substantial memory footprint. In contrast, our method enables an open-vocabulary representation while requiring only a fraction of the computational resources used by CLIP-based approaches.

Tracking in Dynamic Environments Object tracking in dynamic environments has long been a fundamental challenge in robotics. Over the years, a wide range of approaches have been proposed, including filtering techniques such as Kalman filters [31], probabilistic formulations [5], learning-based methods [23], and generative models [2]. More recently, a growing body of work has focused on semantic tracking, where objects are tracked not only based on their motion but also according to their semantic properties and behaviors. In this direction, Li et al. [16] propose

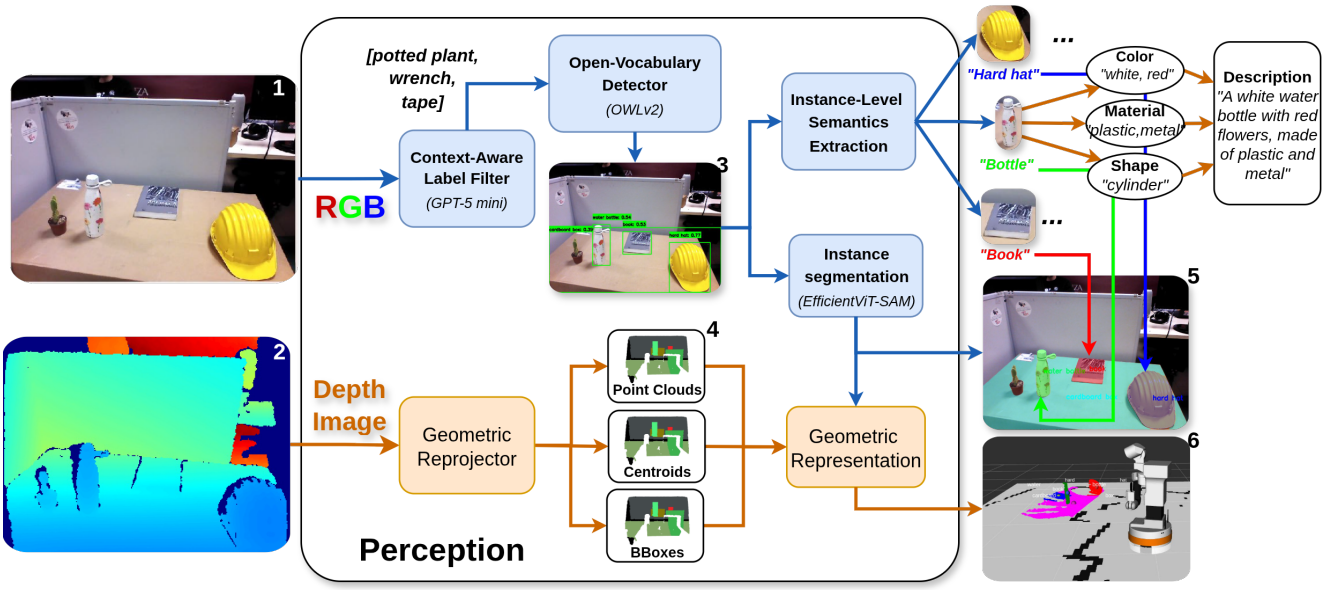

Figure 1. **Perception Module.** The current RGB frame (1) and the corresponding depth image (2) are processed to build the 3DSG of the scene. From the RGB image, open-vocabulary object labels are extracted using a VLM and then grounded in the image to detect the corresponding object bounding boxes on the camera plane (3). At the same time, the VLM is used to extract object-level semantic attributes, including label, color, material, and a fine-grained description. For each object instance, pixel-level segmentation masks are obtained using an object segmentation model and subsequently reprojected into 3D using depth information (5). In parallel, geometric primitives such as centroids and 2D bounding boxes are computed through geometric reprojection and used to estimate 3D bounding boxes (6).

tracking entities in video by jointly reasoning about both their trajectories, answering the question of where an object is, and the underlying semantic events, addressing what is happening. Similarly, Zhang et al. [35] leverage deep convolutional features extracted from a pre-trained VGG [28] network to continuously track dynamic objects over time. SemTrack [30] further advances this line of research by introducing the first large-scale dataset designed to train and evaluate models for semantic tracking in unconstrained environments. Inspired by these approaches, we design a semantic tracking algorithm that maintains object identity by reasoning over semantic attributes such as color and material, while updating object positions within the scene graph as the environment evolves. To the best of our knowledge, this is the first method to perform semantic object tracking directly within a 3DSG representation.

3. Methodology

To construct a lightweight 3DSG from sensory observations, LOST-3DSG relies on two main components: the Perception Module and the Scene Update Module. The Perception Module processes raw sensor data to generate an initial 3DSG of the environment and, at the same time, extracts the semantic attributes associated with each object node. These attributes are later exploited for semantic tracking across observations. The Scene Update Module integrates newly acquired information into the previously constructed 3DSG, which serves as a persistent world anchor. Leveraging the proposed semantic tracking algorithm, this module detects when previously observed object instances reappear, move, or disappear as new observations are processed, and updates the 3DSG accordingly.

This section is organized as follows. Section 3.1 describes the Perception Module and its main components. Section 3.2 details a similarity score based on objects semantic attributes. Finally, Section 3.3 explains how the Scene Update Module uses this function to perform semantic tracking and to update the 3DSG over time.

3.1. Perception Module

The core objective of the Perception Module (PM) (Fig. 1) is to generate a 3DSG representation of the scene. Formally, we define a 3DSG as a hierarchical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v \in \mathcal{V}$ represents an object in the environment. In terms of hierarchical structure, we consider three layers: the `room` layer, the `supporting objects` layer, and the `objects` layer. Edges are defined only between adjacent layers and encode *belonging* relationships. This design choice reflects our focus on object-centric modeling, as the extraction of more complex semantic relations between nodes lies outside the scope of this work. Each node is enriched with a set of attributes that are essential for

the proposed semantic tracking algorithm. In addition to an object *label* ℓ, we assign to each node its 3D *bounding box* $b_{3D} \in \mathbb{R}^6$, *color* c, *material* m, and a short *description* d capturing fine-grained visual characteristics. This description supports instance-level discrimination between objects. In real-world environments, multiple objects may share the same label, material, and color while still differing in subtle visual details, such as surface patterns, wear, or scratches.

The PM builds upon the EMPOWER framework [1], with several modifications tailored to continuous perception in dynamic environments. In particular, we extend EMPOWER with a streaming processing pipeline that incrementally handles sequential observations as the robot navigates through the environment. A first modification concerns open-set object labeling. In the original EMPOWER pipeline, a multi-role planner identifies task-relevant objects, which are then passed to *YOLO-World* [7] for detection. Since our objective is not task planning but scene understanding, we replace this component with a direct VLM query. Given an RGB observation I_t at time t, the VLM produces a set of object labels $\mathcal{L}_t = \{\ell_1, \ell_2, \ldots, \ell_{N_t}\}$, corresponding to the objects present in the scene. In our implementation, we use *GPT-5-mini* as the VLM. The extracted labels are then passed to an open-vocabulary object detector to spatially ground the symbols in the image. Formally, the detector maps the image and label set to a collection of 2D bounding boxes $\mathcal{B}_t = \{b_i \mid b_i = f_{\text{det}}(I_t, \ell_i)\}$, where each bounding box b_i localizes the corresponding object ℓ_i. Unlike EMPOWER, which relies on *YOLO-World*, we adopt *OWLv2* [21] due to its superior open-vocabulary detection performance. Each detected bounding box is then processed by *EfficientViT-SAM* [36] to obtain a pixel-level segmentation mask $m_i = f_{\text{seg}}(I_t, b_i)$, which precisely delineates the spatial extent of the object in the image. In parallel, the cropped image region corresponding to each bounding box is analyzed by the VLM to extract a set of semantic attributes $a_i = \{\ell, c, m, d\}$, which are later used by the semantic tracking module. Using camera intrinsics $\mathbf{K}$, extrinsics $\mathbf{T}$, and the depth map D_t, each segmentation mask is reprojected into 3D space to label the point cloud:

$$\mathcal{P}_i = \pi^{-1}(m_i, D_t, \mathbf{K}, \mathbf{T}), \tag{1}$$

where π^{-1} denotes the back-projection operation. This step associates each 3D point with its corresponding object instance. Finally, the downstream components of EMPOWER are employed to extract object nodes from the segmented point cloud and to infer hierarchical relations between them, resulting in the final 3DSG representation of the environment produced by the PM.

3.2. Lost Similarity Function

To associate object observations over time, we introduce the *Lost Similarity Function (LSF)*, a composite metric designed to quantify the likelihood that two observations, namely a current detection and an existing node in the 3DSG, correspond to the same physical instance. The function integrates multiple complementary cues into a single similarity score, capturing both semantic and appearance-based consistency. The LSF is defined as the weighted combination of four terms:

- *Semantic similarity* (s_ℓ): computed using pre-trained *word2vec* embeddings to measure the cosine similarity between object labels. This term captures semantic relatedness between categories, allowing the tracker to associate objects even when their labels slightly differ but remain conceptually related.
- *Chromatic similarity* (s_c): computed by converting object colors into RGB space and measuring their normalized Euclidean distance. The similarity is defined as

$$s_c = 1 - \frac{\|\mathbf{rgb}_1 - \mathbf{rgb}_2\|_2}{\sqrt{3}}, \tag{2}$$

  where normalization by $\sqrt{3}$ ensures that $s_c \in [0, 1]$.
- *Material similarity* (s_m): analogous to semantic similarity, this term uses *word2vec* embeddings to compare the material attributes of objects, enabling robust matching across observations despite minor variations in appearance.
- *Description similarity* (s_d): computed using a Sentence Transformer (specifically, OpenAI's *text-embedding-3-small* model) to embed fine-grained textual descriptions of objects. This term captures detailed visual characteristics and supports instance-level discrimination between objects that share the same label, color, and material.

The final score is obtained as a weighted linear combination of these components:

$$LSF(o_1, o_2) = \alpha s_\ell + \beta s_c + \gamma s_m + \delta s_d, \tag{3}$$

where α, β, γ, and δ control the relative contribution of each similarity term, and $\alpha+\beta+\gamma+\delta = 1$. These weights can be adjusted to balance semantic and appearance cues depending on the characteristics of the environment and the desired tracking behavior. In our experiment we set $\alpha = 0.15$, $\beta = 0.30$, $\gamma = 0.15$ and $\delta = 0.40$ as we empirically noticed the best results were achieved with this combination.

3.3. Scene Update Module

The Scene Update Module (SUM) (Fig. 2) is formalized in the `scene_update` procedure described in Algorithm 1. To associate a current detection with an object already present in the 3DSG, the system computes the LSF between the detection and all persistent objects. The object with the highest similarity score is selected as the best candidate match, provided that the score exceeds a minimum threshold τ. If no such candidate exists, the

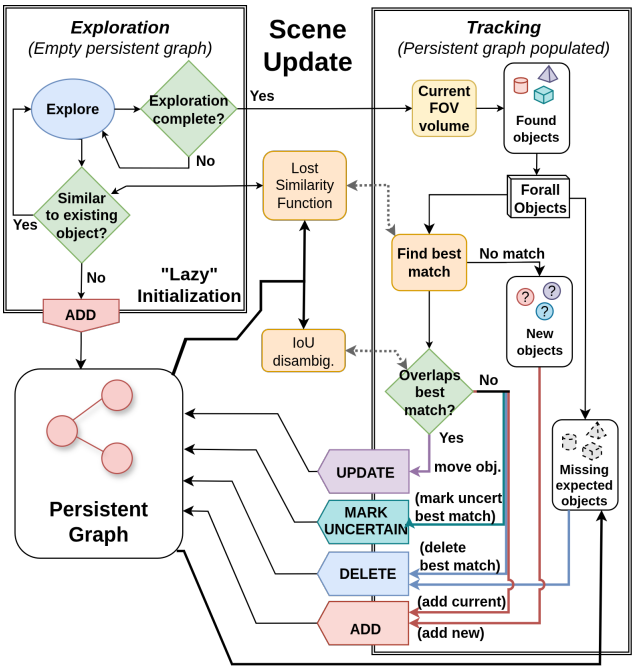

Figure 2. **Scene Update Module**. During the *exploration* phase, the Persistent 3DSG is incrementally populated using LSF-based disambiguation. Once exploration ends, the system switches to the *tracking* phase, where objects in the current FOV are matched by semantic similarity and spatial consistency: moved objects are updated, new objects are added, and missing objects are removed or marked as uncertain.

detection is treated as a previously unseen object. The SUM operates in two modes, controlled by the exploration flag e: *exploration* and *tracking*, and the behavior of the system differs significantly between these modes.

Exploration mode When the exploration flag e is active, the system focuses on populating the 3DSG by incrementally discovering objects as the robot navigates the environment. For each detection, the system attempts to associate it with an existing persistent object. If no valid association is found, a new object node is spawned and added to the set of persistent objects. If a match exists, the corresponding bounding box is updated, but no object is removed or marked as uncertain. This design choice reflects the assumption that, during exploration, all observations contribute to building a complete catalog of the environment. As a result, object removal and identity conflict resolution are intentionally disabled to avoid prematurely discarding valid objects.

Tracking mode Once exploration is complete, the system switches to tracking mode by setting $e = \texttt{false}$. In this phase, the scene manager actively maintains consistency between the 3DSG and the current observations. For each detection $d \in \mathcal{D}$, the algorithm proceeds as follows. First, the best matching persistent object p is retrieved using the LOST similarity function. If no match is found, a new object is spawned and marked as observed in the current frame. Otherwise, the validity of the association is evaluated based on spatial consistency. If the association is deemed valid, the bounding box of the persistent object is updated and the object is marked as seen in the current frame. If the association is invalid, indicating a semantic match but inconsistent spatial evidence, the system resolves the ambiguity by removing the object from its previous location and marking it as uncertain. A new object instance is then spawned at the newly observed position. The removed object is stored in the `uncertain_objects` set, which preserves nodes affected by identity conflicts and allows potential recovery if future observations resolve the ambiguity.

Graph maintenance and cleanup At the end of each update cycle in tracking mode, the system performs a cleanup step based on the robot's point-of-view (POV). The visible volume V is computed from the current camera pose, and persistent objects that were not observed in the current frame but lie within the POV volume are pruned. Similarly, uncertain objects that remain unobserved despite being within the visible region are also removed. This mechanism ensures that objects are only removed when they should have been visible but were not detected, thereby reducing false deletions due to occlusions or limited sensor coverage.

4. Experimental Setup

This section describes the experimental setup used to evaluate the proposed approach in a controlled laboratory environment. Experiments were conducted using a TIAGo robot equipped with ROS 2 Humble and operating in a real-world indoor setting. The robot is provided with a prior map of the environment and, during the exploration phase, continuously localizes itself while detecting and estimating the spatial positions of nearby objects. All observations are subsequently projected into the global map reference frame.

To systematically assess the behavior of the system under increasing levels of complexity, a set of experimental scenarios was designed, featuring multiple tables and objects distributed throughout the environment. In total, three scenes were created, ranging from simple static configurations to more challenging dynamic setups. The evaluated scenarios are summarized as follows, with increasing level of complexity:

1. **Level ($\star$):** the robot observes a scene containing three initially static objects that, over time, move, change position, and eventually disappear from the environment.
2. **Level ($\star$ $\star$):** the environment contains a substantially larger number of objects. While the robot explores the scene from multiple viewpoints, several objects change position and some are removed entirely. This level is more challenging due to the increased object density

Algorithm 1: Scene Update Algorithm

Input : detections $\mathcal{D}$, persistent objects $\mathcal{P}$, current scene $\mathcal{S}$, exploration flag e
Output: updated persistent objects $\mathcal{P}$ and uncertain objects $\mathcal{U}$

Function `scene_update`**:**

- $\mathcal{S}_{\text{seen}} \leftarrow \emptyset$
- **foreach** $d \in \mathcal{D}$ **do**
  - $b \leftarrow$ `FindBBox`(d)
  - **if** $b = \emptyset$ **then continue**
  - $p \leftarrow$ `FindBestMatch`$(d, \mathcal{P})$ `// LSF`
  - **if** $p = \emptyset$ **then**
    - `SpawnObject` $(d, b, \mathcal{S}_{\text{seen}})$
  - **else**
    - **if** e **then**
      - `UpdateBBox` (p, b)
    - **else if** `IsValidAssociation` (p, b) **then**
      - `UpdateBBox` (p, b)
      - `MarkSeen` $(p, \mathcal{S}_{\text{seen}})$
    - **else**
      - `MarkUncertainAndRemove` (p)
      - `SpawnObject` $(d, b, \mathcal{S}_{\text{seen}})$
  - **if** $\neg e$ **then**
    - $V \leftarrow$ `ComputePOVVolume`$(\mathcal{S})$
    - `PrunePersistentObjects` $(V, \mathcal{S}_{\text{seen}})$
    - `PruneUncertainObjects` (V)

Table 1. SUM performance at different Level of Complexity (LoC).

LoC	Objects	Detections	Deletions	Updates
⋆	3	3/3	3/3	3/3
⋆⋆	21	20/21	2/3	1/1
⋆⋆⋆	9	2/3	2/3	13/14

combined with partial observations, object updates, and disappearances that occur outside the robot's field of view.

3. **Level (⋆⋆⋆):** the robot operates in a smaller but highly dynamic environment. Objects undergo numerous position updates, with frequent changes occurring while they are unobserved. This level stress-tests the system's semantic tracking capabilities, requiring it to consistently associate object identities and infer updated locations across time and viewpoints.

An example of execution is given in Fig. 3.

5. Results

In evaluating our system, we aim to answer the following research questions: **Q1.** How effective is the SUM at tracking objects using semantic information alone? **Q2.** What is the relative contribution of each component of the LSF to accurate object identification and association? **Q3.** To what extent does the proposed approach reduce memory consumption compared to CLIP-based methods that store dense semantic features?

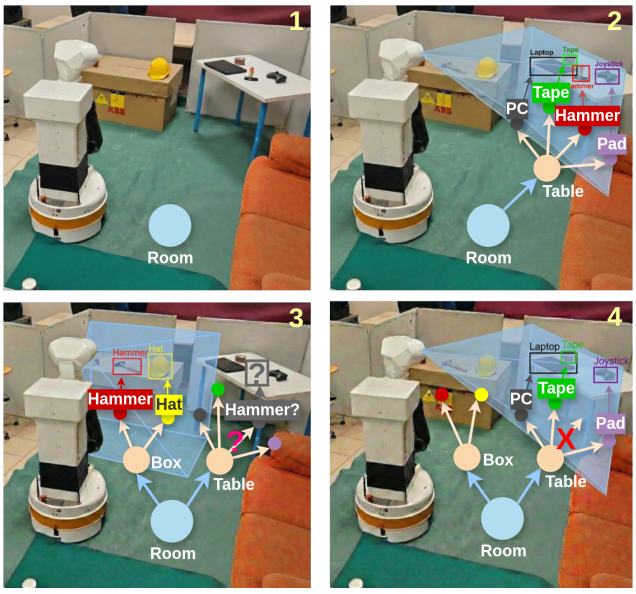

Figure 3. **Execution instance.** The agent observes a household environment (1). During the *exploration* phase, a laptop, a tape roll, a hammer, and a gamepad are added to the 3DSG (2). The hammer is then moved to the brown surface on the left. When the hammer is observed at its new location, the previous instance is marked as uncertain (3), and once the original location is revisited and the object is confirmed absent, the instance is removed from the 3DSG (4).

Answers to Q1 are shown in Table 1, that highlights both the system performance at different level of complexity, as discussed in Sec. 5. The system demonstrates strong robustness in handling object updates across different difficulty levels. In the medium scenario, the main source of complexity arises from the substantially larger number of objects in the environment. Several objects change position and some are deleted while outside the robot's field of view. As a result, the primary challenge lies in correctly retrieving the appropriate object instance through similarity matching, where ambiguities in semantic associations can lead to errors. The hard scenario is particularly demanding due to the high number of object updates. In this setting, many objects frequently change position, often without being directly observed. This configuration effectively stress-tests the system's semantic tracking capabilities, requiring persistent object identity maintenance and accurate graph updates over time. Despite these challenges, the system is able to consistently associate updated observations with previously seen objects and correctly reflect their new positions

in the 3DSG.

We perform an ablation study to answer Q2, and the results are shown in Table 2, averaged across the three levels of complexity. Detections are not reported since the LSF does not affect that component. The full LSF achieves the highest update stability and balanced deletion behavior. When only description similarity is retained, update performance remains relatively strong, confirming that fine-grained textual descriptions are effective at maintaining instance identity over time. However, deletions degrade noticeably in this setting, indicating that descriptions alone are insufficient to robustly decide when an object should be removed. Using only semantic labels leads to poor performance across both metrics, particularly for deletions, due to the susceptibility of label-based matching to hallucinations and category ambiguity. This confirms that labels alone are not reliable for long-term object persistence. Removing material and chromatic similarity does not affect deletions, but significantly degrades update accuracy, supporting the interpretation that low-level appearance cues are primarily used to disambiguate visually similar instances during temporal updates rather than for object lifecycle decisions. Overall, the results highlight the complementary roles of description, semantic, and appearance cues within the LSF.

Finally, we answer Q3 by reporting a comparison of the memory footprint of our system against CLIP-based approaches that rely on storing dense semantic features. We consider the lowest-dimensional CLIP configuration, namely *ViT-B/32*, which produces embeddings of 512 floating-point elements. Assuming 16-bit precision, each embedding requires $512 \times 2\,\mathrm{B} = 1024\,\mathrm{B} \approx 1\,\mathrm{KB}$. In our most complex experimental scene, 21 objects are present and the environment is represented at a voxel resolution of 25 mm per side, resulting in a total number of voxels equal to $N_{\text{voxels}} = 626\,140$ for the scene. Storing a CLIP embedding for each voxel would therefore require a total memory of $M_{\text{CLIP}} = N_{\text{voxels}} \times 1024\,\mathrm{B} \approx 641\,\mathrm{MB}$. In contrast, our approach stores semantic information only at the object level. For each object, we require at most 157 B: 12 B for the two bounding box extents, assuming 16-bit floating-point coordinates, up to 100 B for the textual description, and up to 15 B each for the material, color, and label attributes, assuming UTF-8 encoding with one byte per character. For the entire scene, this results in a total memory usage of $M_{\text{LOST-3DSG}} = 21 \times 157\,\mathrm{B} = 3297\,\mathrm{B}$. This corresponds to slightly more than 3 KB to represent the scene, while preserving open-vocabulary expressiveness and enabling efficient tracking of dynamic objects.

6. Discussion

The results indicate that lightweight, attribute-level semantic representations can support object tracking in dynamic environments without relying on dense visual embeddings.

Table 2. Ablation study on the LSF components, averaged through the various LoCs.

LSF	Deletions	Updates
Full	**0.778**	**0.944**
s_d, s_m, s_c	0.556	0.889
s_ℓ, s_m, s_c	0.667	0.667
s_ℓ, s_d	0.667	0.778
s_d	0.444	0.833
s_l	0.333	0.556

The SUM maintains object identity across time and varying scene complexity, correctly handling object updates and disappearances under partial observability, although errors still occur in the presence of semantic ambiguities. The ablation study shows that fine-grained descriptions play a central role in instance discrimination, while chromatic and material cues contribute to stable temporal updates, and label-only similarity is insufficient for long-term persistence. Also, storing semantics exclusively at the object level yields substantial memory savings compared to CLIP-based voxel representations. However, this efficiency comes with a modest performance cost in challenging scenarios where richer visual features could improve disambiguation. In particular, variability in VLM-generated descriptions can lead to identity fragmentation when similarity scores fall below the matching threshold, and the absence of temporal aggregation makes the system sensitive to noisy or inconsistent semantic estimates. Conservative graph update policies may also introduce temporary object duplication when semantic and spatial evidence disagree. Despite these limitations, the results suggest that the proposed approach offers a promising trade-off between efficiency and tracking accuracy, and that incorporating more robust object profiling, explicit temporal aggregation, and adaptive similarity modeling could further narrow the performance gap while preserving scalability.

7. Conclusion

In this paper, we introduced LOST-3DSG, a lightweight open-vocabulary 3DSG designed to support semantic object tracking in dynamic environments without relying on dense Foundation Models embeddings. By operating on compact, attribute-level representations, the proposed approach significantly reduces memory and computational overhead while still enabling reliable instance association over time. Through real-world experiments, we demonstrated that simple semantic cues, when combined in the proposed manner, are often sufficient to maintain consistent object identities across motion, partial observability, and viewpoint changes. Although the system is intentionally minimal, the results indicate that heavyweight visual embeddings are not always

required for effective open-vocabulary scene understanding. This work is intended to establish foundational insights rather than provide a complete solution. Several directions remain open for future research, including more robust temporal aggregation of semantic descriptions, improved similarity modeling, and closer integration with downstream reasoning and planning modules. We hope this work encourages further exploration of lightweight and scalable alternatives to dense semantic mapping approaches, and serves as a starting point for future research on long-term, open-world 3DSG-based scene understanding.

Acknowledgements

This work has been carried out while Michele Brienza, Francesco Argenziano and Emanuele Musumeci were enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome. Michele Brienza is founded by the European Union - Next Generation EU, Mission I.4.1 Borse PNRR Pubblica Amministrazione (Missione 4) Component 1 CUP B53C23003540006.

References

[1] Francesco Argenziano, Michele Brienza, Vincenzo Suriani, Daniele Nardi, and Domenico D Bloisi. Empower: embodied multi-role open-vocabulary planning with online grounding and execution. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12040–12047. IEEE, 2024. 4

[2] Francesco Argenziano, Miguel Saavedra-Ruiz, Sacha Morin, Daniele Nardi, and Liam Paull. Dynamic objects relocalization in changing environments with flow matching. *arXiv preprint arXiv:2509.16398*, 2025. 2

[3] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R. Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Intl. Conf. on Computer Vision (ICCV)*, 2019. 1, 2

[4] Ermanno Bartoli, Dennis Rotondi, Buwei He, Patric Jensfelt, Kai O. Arras, and Iolanda Leite. Social 3d scene graphs: Modeling human actions and relations for interactive service robots, 2025. 2

[5] Samuel S Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18, 2004. 2

[6] Yun Chang, Luca Ballotta, and Luca Carlone. D-lite: Navigation-oriented compression of 3d scene graphs for multi-robot collaboration. *IEEE Robotics and Automation Letters*, 2023. 2

[7] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16901–16911, 2024. 4

[8] Francis Engelmann, Fabian Manhardt, Michael Niemeyer, Keisuke Tateno, Marc Pollefeys, and Federico Tombari. Opennerf: open set 3d neural scene segmentation with pixel-wise features and rendered novel views. *arXiv preprint arXiv:2404.03650*, 2024. 2

[9] Nicolas Gorlo, Lukas Schmid, and Luca Carlone. Long-term human trajectory prediction using 3d dynamic scene graphs. *IEEE Robotics and Automation Letters*, 2024. 2

[10] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2024. 1, 2

[11] Jun Guo, Xiaojian Ma, Yue Fan, Huaping Liu, and Qing Li. Semantic gaussians: Open-vocabulary scene understanding with 3d gaussian splatting. *arXiv preprint arXiv:2403.15624*, 2024. 2

[12] Daniel Honerkamp, Martin Büchner, Fabien Despinoy, Tim Welschehold, and Abhinav Valada. Language-grounded dynamic scene graphs for interactive object search with mobile manipulation. *IEEE Robotics and Automation Letters*, 2024. 2

[13] Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A real-time spatial perception system for 3d scene graph construction and optimization. *Robotics: Science and Systems (RSS)*, 2022. 2

[14] Hanxiao Jiang, Binghao Huang, Ruihai Wu, Zhuoran Li, Shubham Garg, Hooshang Nayyeri, Shenlong Wang, and Yunzhu Li. Roboexp: Action-conditioned scene graph via interactive exploration for robotic manipulation. In *Conf. on Robot Learning (CoRL)*, 2024. 2

[15] Sebastian Koch, Narunas Vaskevicius, Mirco Colosi, Pedro Hermosilla, and Timo Ropinski. Open3dsg: Open-vocabulary 3d scene graphs from point clouds with queryable objects and open-set relationships. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2

[16] Yunhao Li, Qin Li, Hao Wang, Xue Ma, Jiali Yao, Shaohua Dong, Heng Fan, and Libo Zhang. Beyond mot: Semantic multi-object tracking. In *European Conference on Computer Vision*, pages 276–293. Springer, 2024. 2

[17] Donggeun Lim, Jinseok Bae, Inwoo Hwang, Seungmin Lee, Hwanhee Lee, and Young Min Kim. Event-driven storytelling with multiple lifelike humans in a 3d scene. In *Intl. Conf. on Computer Vision (ICCV)*, 2025. 2

[18] Yuchen Liu, Luigi Palmieri, Sebastian Koch, Ilche Georgievski, and Marco Aiello. Delta: Decomposed efficient long-term robot task planning using large language models. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2025. 2

[19] 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. 1

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean.

Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2

[21] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023. 4

[22] Emanuele Musumeci, Michele Brienza, Francesco Argenziano, Abdel Hakim Drid, Vincenzo Suriani, Daniele Nardi, and Domenico D. Bloisi. Context matters! relaxing goals with llms for feasible 3d scene planning, 2025. 2

[23] Peter Ondruska and Ingmar Posner. Deep tracking: Seeing beyond seeing using recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016. 2

[24] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–824, 2023. 2

[25] 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. 1, 2

[26] 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. In *Conf. on Robot Learning (CoRL)*, 2023. 2

[27] Dennis Rotondi, Fabio Scaparro, Hermann Blum, and Kai O. Arras. Fungraph: Functionality aware 3d scene graphs for language-prompted scene interaction. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2025. 2

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[29] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[30] Pengfei Wang, Xiaofei Hui, Jing Wu, Zile Yang, Kian Eng Ong, Xinge Zhao, Beijia Lu, Dezhao Huang, Evan Ling, Weiling Chen, et al. Semtrack: A large-scale dataset for semantic tracking in the wild. In *European Conference on Computer Vision*, pages 486–504. Springer, 2024. 3

[31] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995. 2

[32] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. *Robotics: Science and Systems (RSS)*, 2024. 1

[33] Zhijie Yan, Shufei Li, Zuoxu Wang, Lixiu Wu, Han Wang, Jun Zhu, Lijiang Chen, and Jihong Liu. Dynamic open-vocabulary 3d scene graphs for long-term language-guided mobile manipulation. *IEEE Robotics and Automation Letters*, 2025. 2

[34] Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2

[35] Jianming Zhang, Xiaokang Jin, Juan Sun, Jin Wang, and Arun Kumar Sangaiah. Spatial and semantic convolutional features for robust visual object tracking. *Multimedia Tools and Applications*, 79(21):15095–15115, 2020. 3

[36] Zhuoyang Zhang, Han Cai, and Song Han. Efficientvit-sam: Accelerated segment anything model without performance loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7859–7863, 2024. 4

[37] Hang Zhao, Xavier Puig, Bolei Zhou, Sanja Fidler, and Antonio Torralba. Open vocabulary scene parsing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2