

A Scene Graph Backed Approach to Open Set Semantic Mapping

Martin Günther^{*1}, Felix Igelbrink^{*1}, Oscar Lima^{*1}, Lennart Niecksch^{*1,2}, Marian Renz^{*1,2},
Martin Atzmueller^{2,1}

¹German Research Center for Artificial Intelligence (DFKI),

Cooperative and Autonomous Systems (CAS), Hamburger Straße 24, Osnabrück, Germany

²Osnabrück University, Semantic Information Systems Group, Wachsbleiche 27, Osnabrück, Germany

{martin.guenther, felix.igelbrink, oscar.lima, lennart.niecksch, marian.renz}@dfki.de, martin.atzmueller@uos.de

Abstract

While Open Set Semantic Mapping and 3D Semantic Scene Graphs (3DSSGs) are established paradigms in robotic perception, deploying them effectively to support high-level reasoning in large-scale, real-world environments remains a significant challenge. Most existing approaches decouple perception from representation, treating the scene graph as a derivative layer generated post hoc. This limits both consistency and scalability. In contrast, we propose a mapping architecture where the 3DSSG serves as the foundational backend, acting as the primary knowledge representation for the entire mapping process.

Our approach leverages prior work on incremental scene graph prediction to infer and update the graph structure in real-time as the environment is explored. This ensures that the map remains topologically consistent and computationally efficient, even during extended operations in large-scale settings. By maintaining an explicit, spatially grounded representation that supports both flat and hierarchical topologies, we bridge the gap between sub-symbolic raw sensor data and high-level symbolic reasoning. Consequently, this provides a stable, verifiable structure that knowledge-driven frameworks, ranging from knowledge graphs and ontologies to Large Language Models (LLMs), can directly exploit, enabling agents to operate with enhanced interpretability, trustworthiness, and alignment to human concepts.

Code and video —

<https://dfki-ni.github.io/SSG-MAKE-2026/>

1 Introduction

Autonomous mobile robots operating in large-scale, real-world environments require a semantic understanding that extends beyond pure geometric mapping to effectively perform complex tasks. While volumetric representations like Truncated Signed Distance Functions (TSDFs) are highly effective for integrating geometric information and producing high-fidelity surface reconstructions, they fundamentally lack an explicit model of the scene’s structure. By treating the environment as a monolithic volume or a collection of

geometric primitives, these approaches often decouple perception from high-level representation, making it difficult to model the hierarchical and semantic relationships necessary for advanced reasoning.

To bridge this gap between sub-symbolic sensor data and symbolic reasoning, 3D Semantic Scene Graphs (3DSSGs) have emerged as a powerful abstraction, representing environments as hierarchical graphs of entities and their relationships. Unlike implicit representations or unstructured point clouds, a 3DSSG acts as an explicit knowledge structure. It achieves this by discretizing the continuous sensor stream into distinct symbolic entities (nodes) and encoding their semantic interactions as edges. This structure aligns the robot’s internal representation with human-understandable concepts, providing a stable grounding for higher-level logic. By formally defining these relationships—whether spatial (e.g., *on top of*) or semantic (e.g., *part of*)—the 3DSSG transforms raw perception into a queryable knowledge base. This allows the system to not only store where geometry is located but to reason about what the environment is, enabling the integration of external knowledge priors and facilitating open-set reasoning.

However, many existing frameworks treat the scene graph as a derivative layer generated post-hoc, which limits the system’s ability to maintain consistency over extended operations and prevents online interactions. In this work, we instead propose a mapping architecture where the 3DSSG serves as the foundational backend for the entire mapping process. Rather than relying solely on dense reconstruction for all tasks, our presented approach uses the 3DSSG to maintain an explicit, spatially grounded representation that supports both flat and hierarchical topologies. This structure allows knowledge-driven frameworks, ranging from knowledge graphs to Large Language Models (LLMs), to directly exploit the structure of the scene, enabling agents to operate with enhanced interpretability and alignment to human concepts.

In this paper, we present the following contributions:

- We describe a modular mapping architecture that establishes the 3DSSG as the foundational backend and primary knowledge representation (Section 3). This design moves beyond rigid hierarchies to support dynamic,

^{*}These authors contributed equally.

modular layers that efficiently separate spatial from semantic relations, utilizing voxel-based representations for scalability.

- We introduce a method for inferring and updating the graph structure in real-time as the environment is explored (Sections 3 and 4). This ensures topological consistency by integrating bottom-up sensory data with top-down expectations derived from prior observations.
- We demonstrate the versatility of our framework through three key integrations (Section 5):
 - *Dynamic Open-Vocabulary Queries*: Incorporating Vision-Language Foundation Models (VLFMs) such as Contrastive Language-Image Pretraining (CLIP) to facilitate natural language interaction.
 - *Real-Time Relation Prediction*: Integrating a 3DSSG prediction network to infer structural relationships dynamically during operation.
 - *Real-World Deployment*: Validating the system on a mobile robot (TIAGo) to build maps from real-world sensor data in large-scale environments.

2 Related Work

Our work builds upon 3DSSGs to bridge the gap between structured environment representations and open-set perception, integrating the knowledge-level expectation strategies of ExPriS (Renz et al. 2026) with the language-driven exploration capabilities of LIEREx (Igelbrink et al. 2026). In the following, we review relevant approaches in 3DSSGs and VLFM integration for semantic mapping.

3D Semantic Scene Graphs

A 3DSSG serves as a structured abstraction of the environment. As identified in our recent survey on online knowledge integration (Igelbrink et al. 2024), this paradigm is uniquely positioned to bridge the gap between sub-symbolic sensor data and high-level symbolic reasoning (Armeni et al. 2019; Rosinol et al. 2020). In this representation, nodes correspond to physical entities (ranging from individual objects to rooms and buildings) while edges encode the spatial or logical relationships between them (Wald et al. 2020; Hughes, Chang, and Carlone 2022). By organizing these entities, often in a hierarchical manner, 3DSSGs provide a flexible backend for complex downstream tasks such as navigation, task planning, and manipulation (Kim et al. 2020; Agia et al. 2022). Furthermore, this structured format acts as a vital interface for external reasoning tools, enabling the integration of knowledge graphs and LLMs to perform open-set queries and common-sense reasoning directly on the mapped environment (Lv et al. 2024; Strader et al. 2024; Ren et al. 2024).

Construction-Based Methods Pioneering works, such as Armeni et al. (2019), introduced hierarchical scene graphs to organize entities into layers (e. g., buildings, rooms, objects). This concept was extended by Rosinol et al. (2020) into 3D Dynamic Scene Graphs (DSGs), which include dynamic agents and capture actionable spatial information. The *Hydra* framework (Hughes, Chang, and Carlone 2022) further

advanced this by enabling the real-time, incremental construction of DSGs from sensor data using spatial topology and loop closures. While these methods produce spatially consistent maps suitable for navigation, they typically rely on rigid heuristics and fixed ontologies, limiting their flexibility in handling semantic relationships beyond simple containment or adjacency.

Prediction-Based Methods Conversely, learning-based approaches focus on inferring semantic relationships directly from data. Wald et al. (2020) introduced the 3DSSG dataset as the first benchmark for the prediction of 3DSSGs from sensor data, proposing Graph Neural Networks (GNNs) as the main model architecture. To address the limitations of offline processing, Wu et al. (2021) proposed *SceneGraphFusion*, an incremental pipeline built on top of the 3DSSG dataset that fuses frame-wise graph predictions into a global graph representation. However, these purely data-driven methods often operate in a bottom-up fashion, failing to exploit the top-down knowledge priors (e. g., structural expectations) that are central to hybrid systems. Furthermore, they rely heavily on labeled datasets and are difficult to extend to real domains. Our proposed architecture acts as a hybrid solution, utilizing the incremental update capabilities of prediction networks while maintaining a foundational backend that supports hierarchical grounding.

Open-Set Semantic Mapping

The advent of VLFMs, such as CLIP (Radford et al. 2021), has shifted semantic mapping from closed-set vocabularies to open-set embedding spaces.

Early integrations of VLFMs into 3D mapping utilized implicit representations, such as Neural Radiance Fields (NeRFs). LERF (Kerr et al. 2023) and CLIP-Fields (Shafiqullah et al. 2023) optimize scene-specific fields to query geometry with natural language. While visually accurate, these methods are computationally intensive and difficult to update. Geometry-based approaches like OpenScene (Peng and Genova 2023) and ConceptFusion (Jatavallabhula et al. 2023) project dense pixel-aligned features onto point clouds or meshes. Although scalable, these representations often lack object-level abstraction, treating the environment as a collection of points rather than distinct entities, thus limiting scalability.

Recent works have sought to combine the structure of scene graphs with the flexibility of open-set embeddings. Werby et al. (2024) proposed Hierarchical Open-Vocabulary Scene Graphs (HOV-SG), which aggregate CLIP features within a hierarchical 3DSSG inspired by Hydra to support language-based navigation. Similarly, ConceptGraphs (Gu et al. 2024) and CLIO (Maggio et al. 2024) construct object-centric maps that enable complex querying and task-dependent clustering. Our approach aligns with this trend but emphasizes the 3DSSG as the *foundational backend* for the entire mapping process, integrating prediction, construction, and open-set features into a single, unified consistency layer rather than treating them as separate downstream tasks.

3 Mapping Approach

The overarching design objective of our architecture is to establish the 3DSSG as the central data structure and single source of truth for the entire mapping life cycle. Unlike conventional approaches that generate scene graphs as a post-processing step after the semantic mapping itself has finished, our framework aims to maintain the graph as the primary backend and enhance it continuously over time. This ensures that the graph encapsulates all relevant data regarding the current map state and remains topologically and semantically valid at each discrete time step. The system is designed for the incremental, online integration of high-bandwidth sensor data, leveraging a fully GPU-accelerated pipeline to guarantee computational efficiency suitable for real-world robotic applications. The pipeline runs on a NVIDIA RTX 5080 GPU and achieves an average processing rate of 7–8 Hz, suitable for online operation.

Geometric Mapping

Since our research focuses primarily on map representation, we deliberately decided to prioritize the development of our mapping architecture rather than introducing the complexity of a full SLAM framework. Consequently, for real-world applications, our system relies on robust external pose estimates, e. g., from an external tracking system or by localization in an existing map. Leveraging available high-resolution polyhedral maps of our test environment, we employ MICP-L (Mock et al. 2024) for robot pose tracking using an Ouster OS0-128 LiDAR. Additionally, we align an RGB-D camera extrinsically to the LiDAR via ICP registration. We then utilize MICP-L to update the robot’s pose estimation at the LiDAR timestamp nearest to each captured camera frame. To enhance pose accuracy and mitigate extrinsic calibration and synchronization errors, we perform a subsequent offline pose-graph optimization step (Choi, Zhou, and Koltun 2015) directly on the camera poses. Given that the initial MICP-L estimates exhibit minimal error and negligible drift, loop closures are detected efficiently via frustum checks.

Scene Graph Backend

Our approach adopts an instance-based, object-centric mapping paradigm, drawing inspiration from state-of-the-art dense semantic mapping methods such as Concept-Graphs (Gu et al. 2024) and BBQ (Linok et al. 2025), as well as the view-dependency logic of Kassab et al. (2024).

The backend data structure is implemented as a multi-layered 3DSSG, organized into three distinct abstraction layers:

1. A *Frames Layer* that tracks 6D pose information and raw 2D segmentation masks for all keyframes processed;
2. A *Segments Layer* serving as an intermediate representation, tracking all currently active 3D object fragments including their raw geometry and extracted feature vectors. While currently integrated closely with the objects layer to reduce overhead, this layer is designed to support lazy, on-demand merging strategies in future iterations. Edges between the *Segments* and *Frames* layers

naturally describe co-visibility relations and the stability of each segment;

3. An *Objects Layer* that tracks persistent, consolidated object instances. This layer maintains the spatio-semantic relations between instances as well as their accumulated geometric and semantic feature information.

Mapping Pipeline

The mapping pipeline is designed to be strictly incremental, ensuring the 3DSSG backend transitions from one valid state to the next after each integration step.

Segmentation and Feature Extraction For each incoming RGB-D frame, we first employ a pre-trained *Segment Anything Model* (SAM) to extract a set of potentially overlapping segmentation masks. To ensure high-quality input, a filter bank immediately discards masks with low confidence, extreme aspect ratios, or insufficient size. Mask boundaries are further refined using depth discontinuities to prevent “bleeding” of segments across physical boundaries. The pipeline is agnostic to the specific SAM variant; in our experiments, we utilize FastSAM (Zhao et al. 2023) due to its favorable inference speed, which is critical for online operation.

Simultaneously, we extract dense visual feature vectors from the RGB image using the DINOv2 model (Oquab et al. 2024). Inspired by the efficiency strategies in Linok et al. (2025) and MaskCLIP (Dong et al. 2023), we utilize the local per-patch features directly from the model’s intermediate layers rather than computing aggregated features for separate object crops. This design allows us to execute the heavy feature encoder only once per frame, significantly increasing throughput. Since these features are primarily used for short-term data association between frames, the lack of global context in patch features is acceptable at this stage. Per-segment feature descriptors are obtained by aggregating the patch features within the corresponding refined segmentation masks.

Local Graph Generation The depth data is projected into a 3D point cloud for each valid segment. To mitigate sensor noise and remove segmentation artifacts (e. g., flying pixels), we filter these points using a custom CUDA-based implementation of the DBSCAN algorithm. This implementation utilizes a parallel, batched union-find approach to maximize performance on the GPU. The resulting point clouds are voxelized to ensure uniform point density across the map.

These processed 3D segments are initially instantiated in a *local* 3DSSG. This local graph mirrors the structure of the global graph but describes only the information contained in the current frame. This intermediate representation provides a crucial hook for additional graph-based refinement and prediction modules (see Section 4) before the data is committed to the global state.

Data Association and Global Integration The integration of the *local* graph into the persistent *global* graph is performed via a robust two-stage matching process.

Stage 1: Greedy Association First, we employ a conservative greedy matching strategy. We compute an affinity matrix based on the Intersection over Union (IoU) of 3D bounding boxes and the cosine similarity of the DINOv2 feature vectors. Any *local* segment that overlaps with a *global* segment and exceeds a strict similarity threshold is merged immediately. Unmatched segments are instantiated as new nodes in the global graph. This stage is designed to be safe, merging only unambiguous matches to prevent corruption of the map. This simple matching does not perform any conflict resolution and does not throw away potentially valuable segments from the segmentation stage.

Stage 2: Active Refinement Greedy merging alone often leads to over-segmentation and the accumulation of spurious artifacts due to temporal inconsistencies in the SAM predictions. Previous works typically addressed this via periodic, computationally expensive offline passes over the entire map (Gu et al. 2024). To maintain the 3DSSG as a real-time “best estimate,” we introduce an active refinement stage that runs immediately after greedy merging. Crucially, this refinement considers only the *active subset* of the graph—nodes that were either modified or newly created in the current step. These nodes are matched against their spatial neighbors using voxel-grid overlap. For each node in this subset, the system evaluates potential merge candidates based on feature stability and overlap. This process naturally resolves conflicts and “daisy-chain” merge effects, effectively fusing fragmented raw segments into coherent object instances that approach the semantic granularity of the underlying foundation model. In future work, we aim to augment this geometric heuristic with a learned semantic edge prediction model to infer merges based on learned spatial compatibility.

Vision-Language Feature Integration

To facilitate open-vocabulary queries, we integrate additional CLIP features directly into the graph nodes. Similar to our DINOv2 integration above, we circumvent the computational bottleneck of running the CLIP encoder on individual mask crops. Instead, we adopt the MaskCLIP (Dong et al. 2023) paradigm: we extract per-patch features from the penultimate transformer layer of a standard CLIP model and post-process them using the model’s native attention and projection weights.

While efficient, this method yields features that are strongly biased towards local textures rather than global object semantics (e. g., encoding “fabric pattern” rather than “couch”). To compensate, we implement a *gated integration* mechanism. We compute an additional global CLIP embedding for the entire frame and modulate the local patch features based on their cosine similarity to this global context. This enhances the semantic richness of segments that align with the scene context while allowing distinct outliers to retain their specific characteristics.

Finally, we compute a *view quality score* for each segment integration, derived from geometric factors such as the segment’s relative size and distance from the image center (Kassab et al. 2024). This score acts as a confidence weight during feature aggregation: features extracted from optimal

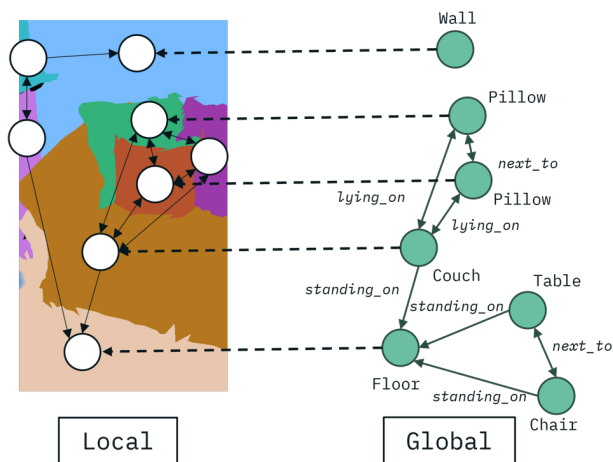


Figure 1: The heterogeneous graph model for Incremental Predicate Prediction (IPP) from Renz, Igelbrink, and Atzmüller (2025). The local graph corresponds to the segments layer or the frame-only local scene graph, the global graph corresponds to the objects layer or the fully integrated and refined graph from the mapping pipeline described in Section 3. Dashed edges are between matched objects between the local frame and global map to provide information flow during message passing. While object classes are displayed here for clarity, the actual global scene graph only contains open-set features.

viewpoints contribute significantly to the node’s persistent embedding, while oblique or occluded observations have minimal impact. This ensures that the open-set representation remains robust against noisy fleeting observations.

4 Incremental Predicate Prediction

The presented mapping framework, as well as existing approaches like ConceptGraphs (Gu et al. 2024) and CLIO (Maggio et al. 2024), show great flexibility towards environments with complex objects and structures. However, edges in these scene graphs usually have no deeper semantic meaning and are only included for hierarchy or spatial structure. Relationships such as *on top of* or *attached to* are useful information for reasoning algorithms such as task planners, or even LLMs used instead of a planner. For this reason, we are combining the mapping pipeline with methods from scene graph prediction to predict semantic labels for relationships between objects. By introducing semantic relationships between open-set objects, we maintain the flexibility of open-set scene graphs while enhancing the semantic part of the 3DSSG. Specifically, we use the RIO27 predicate label set, a subset of the 3DSSG dataset (Wald et al. 2020), because it supplies ground-truth annotations for 3DSSG prediction and is the most widely used dataset.

Using a closed-set prediction approach is less flexible than using an LLM, VLM, or VLFM. However, short inference times are a requirement for real-time mapping; therefore, we aim for a smaller architecture that could even be deployed on the edge.

Prediction Graph Model

To infer the semantic relationships between objects in the graph, we base the prediction on the heterogeneous incremental graph model introduced by Renz, Igelbrink, and Atzmueller (2025). The model consists of a local/global structure: a global layer, which contains the mapped scene graph from previous time steps, and a local layer, which contains the scene graph built only on the local sensor data frame. The overall goal for each time step is to predict the predicate labels between objects in the local layer while keeping the open-set features, in contrast to common scene graph prediction approaches like Wald et al. (2020) where the object classes are predicted as well. The layers in the graph prediction model correspond exactly to the local and global scene graphs introduced in Section 3. Predicate predictions from the local layer are incrementally integrated into the global scene graph. Since segments in the local frame have (for the most part) a corresponding object in the global graph, we can connect them in the graph model for prediction (see Figure 1). This way, information from the previously predicted semantic predicates can flow into the local prediction during the message-passing process, assisting predictions of predicates that have been observed before.

Node features: For each node in the local and the global scene graph, we sample 256 voxel points. Additionally, we construct a geometric descriptor of each node’s 3D segment in the map/frame, analogous to SceneGraphFusion (Wu et al. 2021) and the approach by Renz, Igelbrink, and Atzmueller (2025):

$$[c, \text{std}, l, w, h, L, V]$$

where c is the 3D segment’s oriented bounding box center, std the standard deviation of each point from the center, l , w , and h the bounding box dimensions, L the bounding box’s biggest side, and V the bounding box volume. Furthermore, we use the DINOv2 features the mapping framework provides for each segment.

Edges: A standard practice to determine the presence of an edge between two objects is by connecting objects within a 0.5 m radius to each other. However, we find that this leads to overly dense graphs and does not represent a scene’s structure sufficiently. Therefore, we decided to connect only touching objects and predict the labels of those edges. To find touching object pairs, we add 5 cm padding to each segment’s bounding box and check for intersections. To connect matching nodes between the local and global layers, we reuse the pairs found by the greedy association step in Section 3 and add unlabeled directed edges between matching nodes between the local and global layers.

Graph Neural Network

After constructing the prediction graph from the mapping pipeline’s scene graph, a heterogeneous graph neural network is used to classify the edge labels.¹ Descriptor, DINO

¹The implementation of the Incremental Predicate Prediction network described in this subsection is ongoing work as well as the tight integration with the real-time mapping pipeline. The following describes the architecture of this component.

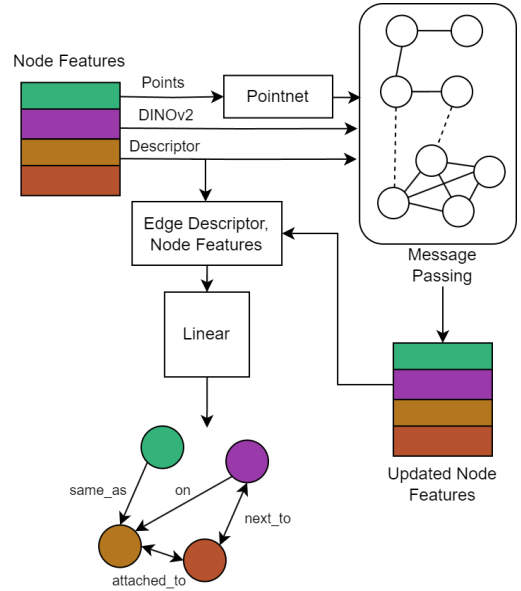


Figure 2: Graph neural network architecture for Incremental Predicate Prediction.

features, and points are used as node features for the local and the global layers, although the points are embedded into a latent vector using PointNet (Qi et al. 2017) first. Message passing is done using a heterogeneous GraphSage (Hamilton, Ying, and Leskovec 2017), where edges in the local and global layers respectively, as well as the intra-layer edges, are modeled using separate linear layers.

While both node types provide the same kind of information, the heterogeneity comes from the already predicted, integrated edges in the global scene graph, which serves as a semantic prior and is included in message passing. For this, we extend the GraphSage message passing for the global layer to include a text embedding such as GloVe (Pennington, Socher, and Manning 2014) and a respective linear layer:

$$x'_i = \gamma \left(x_i, \Theta \{ x_j + \phi(e_{ij}) \forall j \in \mathcal{N}(i) \} \right) \quad (1)$$

where γ is a feed-forward layer with non-linear activation, Θ is an aggregation function like mean or sum, ϕ is a linear layer, and x_i , x_j , and e_{ij} are target node-, neighbor node-, and edge features respectively.

After two layers of message passing, the final edge features are created by concatenating the two updated node features of the nodes connected by the respective edge, n_i and n_j , as well as a combined edge descriptor:

$$[c_j - c_i, \text{std}_j - \text{std}_i, \log(\frac{l_j}{l_i}, \frac{w_j}{w_i}, \frac{h_j}{h_i}), \log(\frac{L_j}{L_i}), \log(\frac{V_j}{V_i})]$$

Concatenated features and descriptors are then passed through a feed-forward network to obtain the final classification logits (see Figure 2).

The predicate prediction is modeled as a multilabel problem, since this way edges can have no labels and therefore would not be integrated into the global graph.

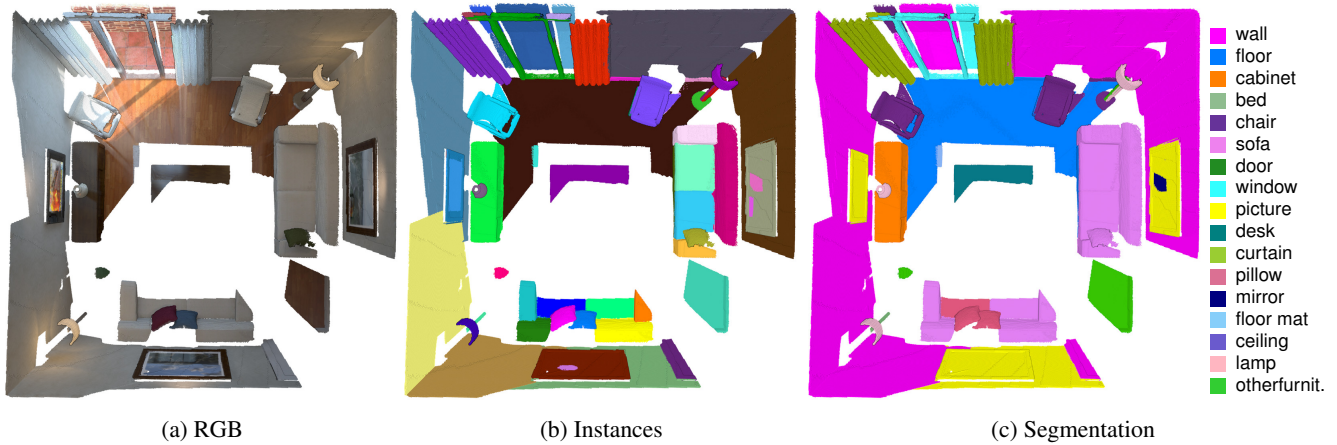


Figure 3: Preliminary results of the mapping pipeline on the small-scale ICL dataset. The segmentation was obtained by comparing each object’s CLIP feature with the embeddings of the NYU-40 label set.

Training Data Preparation

We construct the training data for the Incremental Predicate Prediction using 3D scenes from the 3RScan dataset (Wald et al. 2019) as well as the predicate labels from the 3DSSG dataset (Wald et al. 2020).

We incrementally build a 3D map using the mapping pipeline while storing both the local scene graph and the global scene graph at that timestep as described in Section 4, with sampled points, DINO features, and descriptors, as well as edges between touching objects, classified global edges, and matched nodes connected by intra-layer edges.

5 Preliminary Results

In this section, we evaluate the proposed mapping architecture through qualitative experiments on both synthetic and real-world datasets. We first demonstrate the mapping capabilities on the ICL RGB-D dataset, followed by an analysis of the IPP integration, and finally present a real-world deployment on a TIAGo mobile robot.

Mapping Results

To validate the fundamental capabilities of our proposed architecture, we conducted preliminary experiments on the ICL RGB-D dataset (Handa et al. 2014).

Figure 3 presents the qualitative results on the $k\tau 1$ trajectory, displaying the reconstructed color mesh, the instance segmentation, and the resulting semantic segmentation for a small scale environment. The semantic labels were generated by querying the open-set CLIP features of each node against the text embeddings of the NYU-40 label set.

As observed in the instance segmentation layer, the system successfully isolates major structural elements. However, certain areas exhibit over-segmentation artifacts. These occur primarily where the active refinement stage (see Section 3) has not yet accumulated sufficient evidence, such as consistent spatial overlap or feature similarity, to confidently merge adjacent segments. This behavior is a consequence of



Figure 4: Visualization of local scene graph (yellow) and global scene graph (blue) with inter- and intra-layer edges with a mapped scene from the 3RScan dataset. Predicted edges are integrated into the global graph using ground truth annotations from Wald et al. (2020).

our conservative merging strategy, designed to prevent the irreversible fusion of distinct objects.

Regarding semantic accuracy, the integration of CLIP features proves effective for distinct, well-separated objects within uncluttered scenes. However, the performance degrades in highly cluttered areas or with many very small objects. This limitation stems from the current inherent resolution constraints of the used MaskCLIP approach, where the patch size of the underlying vision transformer limits the granularity of the extracted features. Addressing this resolution mismatch by improving the feature aggregation remains a key objective for our future work.

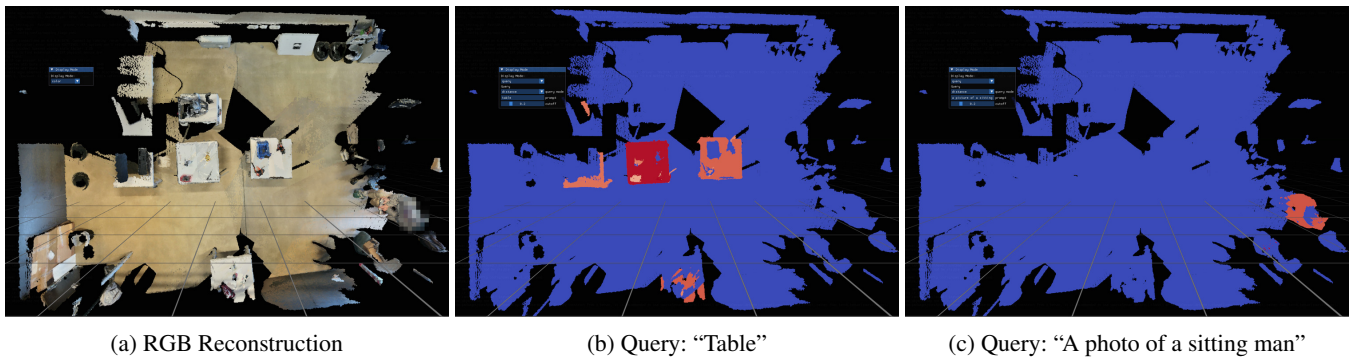


Figure 5: Mapping results and sample queries on real-world data. (a) Successful map integration given the estimated poses. (b) & (c) Visualization of cosine similarity between instances and the respective text queries.

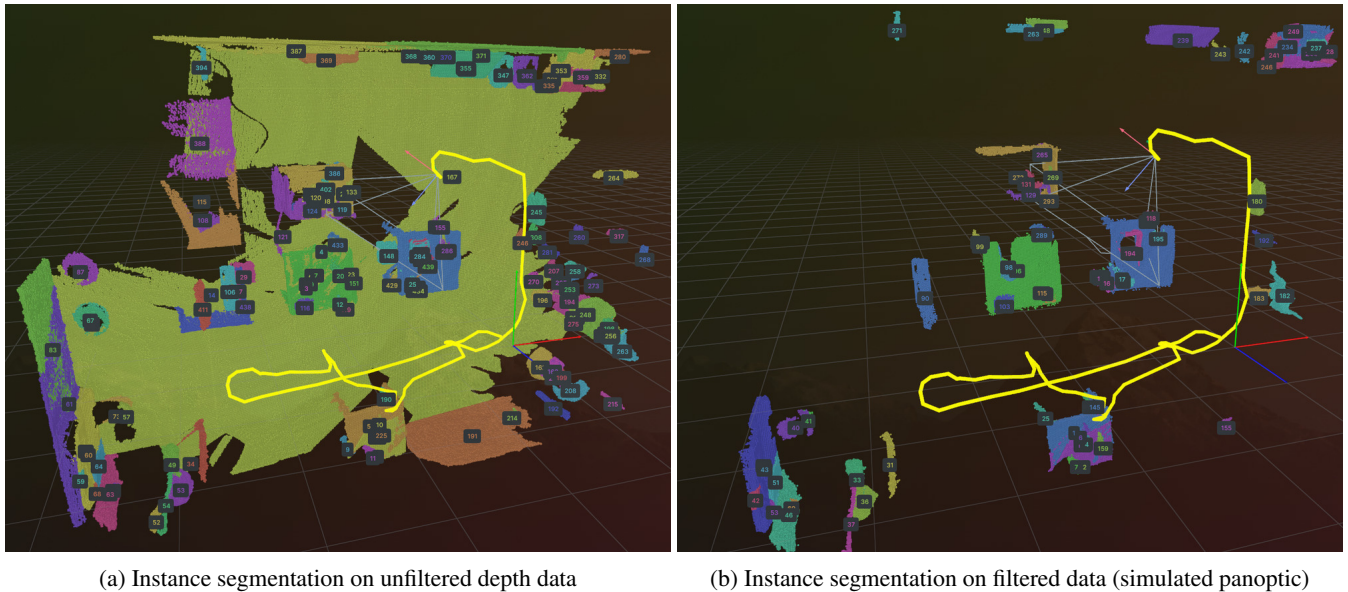


Figure 6: Comparison of mapping results with and without geometric filtering. (a) Mapping performed on raw unfiltered depth data. (b) Points corresponding to the static building map (used for localization) are filtered out, effectively mimicking panoptic segmentation to remove structural *stuff* classes.

Incremental Predicate Prediction Integration

To showcase the Incremental Predicate Prediction (IPP), we display the integration of the predicted predicates into the 3DSSG on example scenes from the 3RScan dataset (see Figure 4). RGB-D frames are rendered from the 3D scene reconstructions. For each frame, the predicate predictions are merged into the global scene graph and used in the next time step. Edges without predictions are not integrated.

While the IPP on 3D data using graph neural networks has been tested successfully in the past (Renz, Igelbrink, and Atzmueller 2025) on the 3DSSG dataset (Wald et al. 2020), integrating the IPP with the mapping framework and adapting it to different datasets is part of future work.

Real-World Application

To demonstrate real-world applicability of our proposed framework, we evaluated our system using the TIAGo mo-

bile robotic platform. Initial pose estimates are provided by MICP-L (Mock et al. 2024) using an Ouster OS0-128 LiDAR against a high-resolution triangle mesh of the environment², which includes only the static building structure. RGB-D sequences were recorded at 1280×720 resolution via a Femto Bolt ToF RGB-D camera. We processed a subset of 440 frames from the resulting frame-pose pairs in order to generate the map. The resulting map and the selected sample queries are presented in Figures 5 and 6. While instances are not as cleanly segmented as in the synthetic ICL dataset (c. f., Figures 3 and 6), the system successfully integrates the majority of instances into a coherent map. Most errors can be attributed to sensor noise, minor pose inaccuracies, and FastSAM over-segmentation, which is particularly

²The model can be found at: https://github.com/DFKI-NI/pbr_tools/tree/jazzy/pbr_gazebo/models/cic/meshes

noticeable on objects that were only partially observed.

Furthermore, we conducted an experiment using Rmagine (Mock, Wiemann, and Hertzberg 2023) to filter the input data. By matching depth data against the building model used for localization, we mimic a panoptic segmentation step to filter out *stuff* classes. Figure 6 illustrates the impact of this filter: floors and walls are effectively removed while most *things* remain, significantly reducing the number of spurious instances in the final map.

6 Discussion and Future Outlook

In this paper, we presented a hybrid mapping architecture that establishes the 3DSSG not as a derivative output, but as the foundational backend for robot perception. By moving away from monolithic geometric maps and towards an explicit semantic knowledge structure, we have demonstrated a powerful and flexible framework that bridges the gap between sub-symbolic sensory data and symbolic reasoning – a core objective of hybrid intelligence, which seeks to synergize the robustness of statistical learning with the interpretability of symbolic reasoning.

A key strength of this architecture is its representation-agnostic nature. While our current implementation utilizes a voxel-based representation to efficiently handle the scalability requirements of large-scale environments, the high-level symbolic layer is designed to be independent of the specific underlying geometric format. Consequently, the backend could readily support TSDFs, meshes, or neural fields as the geometric source, provided they can be grounded into the graph’s node structure. This modularity not only offers significant flexibility for different robotic platforms but also ensures that the system maintains a stable, verifiable structure that aligns with human concepts. As argued in Section 1, this explicit structure is what enables the seamless integration of top-down expectations and open-vocabulary queries, capabilities that are difficult to achieve with purely implicit or geometric representations.

Despite these advantages, the current implementation of the IPP module faces a limitation common to many neuro-symbolic approaches: a reliance on supervised learning. The graph neural networks employed for relation inference currently require extensive ground-truth labeling, such as that provided by the 3DSSG dataset (Wald et al. 2020). Hence, such a dependence on annotated datasets currently restricts the system’s ability to generalize to novel environments where such dense relational labels are unavailable.

To overcome the bottleneck of supervised relational learning, our future work will focus on inferring spatial relations in an unsupervised manner. We aim to adopt strategies similar to the SEMAP framework (Deeken et al. 2015), which extracts geometric spatial relations (e.g., *on*, *close to*) directly from the map topology without requiring training data. We are currently developing a unified approach for object detection, 6D pose estimation, and spatial relation inference that leverages these geometric heuristics (Niecksch et al. 2024). By replacing the supervised prediction head with these unsupervised geometric extractors, we aim to enable the autonomous construction of rich semantic scene graphs in entirely open and unlabelled worlds.

Acknowledgements

This work is supported by the ExPrIS and LIEREx projects through grants from the German Federal Ministry of Research, Technology and Space (BMFTR) with grant numbers 16IW23001 and 16IW24004. The DFKI Niedersachsen (DFKI NI) is sponsored by the Ministry of Science and Culture of Lower Saxony and the VolkswagenStiftung.

References

- Agia, C.; Jatavallabhula, K. M.; Khodeir, M.; Miksik, O.; Vineet, V.; Mukadam, M.; Paull, L.; and Shkurti, F. 2022. Taskography: Evaluating robot task planning over large 3D scene graphs. In *CoRL*, 46–58.
- Armeni, I.; He, Z.-Y.; Gwak, J.; Zamir, A. R.; Fischer, M.; Malik, J.; and Savarese, S. 2019. 3D Scene Graph: A Structure for Unified Semantics, 3D Space, and Camera. In *ICCV*, 5664–5673.
- Choi, S.; Zhou, Q.-Y.; and Koltun, V. 2015. Robust reconstruction of indoor scenes. In *CVPR*, 5556–5565.
- Deeken, H.; Wiemann, T.; Lingemann, K.; and Hertzberg, J. 2015. SEMAP – A Semantic Environment Mapping Framework. In *ECMR*, 1–6.
- Dong, X.; Bao, J.; Zheng, Y.; Zhang, T.; Chen, D.; Yang, H.; Zeng, M.; Zhang, W.; Yuan, L.; Chen, D.; Wen, F.; and Yu, N. 2023. MaskCLIP: Masked Self-Distillation Advances Contrastive Language-Image Pretraining. In *CVPR*, 10995–11005.
- Gu, Q.; Kuwajerwala, A.; Morin, S.; Jatavallabhula, K. M.; Sen, B.; Agarwal, A.; Rivera, C.; Paul, W.; Ellis, K.; Chellappa, R.; Gan, C.; de Melo, C. M.; Tenenbaum, J. B.; Torralba, A.; Shkurti, F.; and Paull, L. 2024. ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning. In *ICRA*, 5021–5028.
- Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. *Adv. Neural Inf. Process. Syst. (NIPS)*, 1025–1035.
- Handa, A.; Whelan, T.; McDonald, J.; and Davison, A. 2014. A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM. In *ICRA*.
- Hughes, N.; Chang, Y.; and Carlone, L. 2022. Hydra: A Real-time Spatial Perception System for 3D Scene Graph Construction and Optimization. In *Robotics: Science and Systems (RSS)*, Article 50.
- Igelbrink, F.; Niecksch, L.; Renz, M.; Günther, M.; and Atzmueller, M. 2026. LIEREx: Language-Image Embeddings for Robotic Exploration. *Künstl. Intell.*
- Igelbrink, F.; Renz, M.; Günther, M.; Powell, P.; Niecksch, L.; Lima, O.; Atzmueller, M.; and Hertzberg, J. 2024. Online Knowledge Integration for 3D Semantic Mapping: A Survey. *arXiv preprint arXiv:2411.18147*.
- Jatavallabhula, K. M.; Kuwajerwala, A.; Gu, Q.; Omama, M.; Iyer, G.; Saryazdi, S.; Chen, T.; Maalouf, A.; Li, S.; Keetha, N. V.; Tewari, A.; Tenenbaum, J. B.; de Melo, C. M.; Krishna, K. M.; Paull, L.; Shkurti, F.; and Torralba, A. 2023. ConceptFusion: Open-set multimodal 3D mapping. In *Robotics: Science and Systems (RSS)*.

- Kassab, C.; Mattamala, M.; Morin, S.; Büchner, M.; Valada, A.; Paull, L.; and Fallon, M. F. 2024. The Bare Necessities: Designing Simple, Effective Open-Vocabulary Scene Graphs. *arXiv preprint arXiv:2412.01539*.
- Kerr, J.; Kim, C. M.; Goldberg, K.; Kanazawa, A.; and Tancik, M. 2023. LERF: Language Embedded Radiance Fields. In *ICCV*, 19672–19682.
- Kim, U.-H.; Park, J.-M.; Song, T.-J.; and Kim, J.-H. 2020. 3-D scene graph: A sparse and semantic representation of physical environments for intelligent agents. *IEEE Trans. Cybern.*, 50(12): 4921–4933.
- Linok, S.; Zemskova, T.; Ladanova, S.; Titkov, R.; Yudin, D.; Monastyrny, M.; and Valenkov, A. 2025. Beyond Bare Queries: Open-Vocabulary Object Grounding with 3D Scene Graph. In *ICRA*, 13582–13589.
- Lv, C.; Qi, M.; Li, X.; Yang, Z.; and Ma, H. 2024. SGFormer: Semantic Graph Transformer for point cloud-based 3D scene graph generation. *Proc. Conf. AAAI Artif. Intell.*, 38(5): 4035–4043.
- Maggio, D.; Chang, Y.; Hughes, N.; Trang, M.; Griffith, D.; Dougherty, C.; Cristofalo, E.; Schmid, L.; and Carlone, L. 2024. Clio: Real-Time Task-Driven Open-Set 3D Scene Graphs. *IEEE Robot. Autom. Lett.*, 9(10): 8921–8928.
- Mock, A.; Wiemann, T.; and Hertzberg, J. 2023. Rmagine: 3D Range Sensor Simulation in Polygonal Maps via Ray Tracing for Embedded Hardware on Mobile Robots. In *ICRA*.
- Mock, A.; Wiemann, T.; Pütz, S.; and Hertzberg, J. 2024. MICP-L: Mesh-Based ICP for Robot Localization Using Hardware-Accelerated Ray Casting. In *IROS*, 10664–10671.
- Niecksch, L.; Mock, A.; Igelbrink, F.; Wiemann, T.; and Hertzberg, J. 2024. Mesh-based Object Tracking for Dynamic Semantic 3D Scene Graphs via Ray Tracing. In *RSS'24 Workshop on Semantics for Robotics: From Environment Understanding and Reasoning to Safe Interaction*.
- Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H. V.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; Assran, M.; Ballas, N.; Galuba, W.; Howes, R.; Huang, P.; Li, S.; Misra, I.; Rabbat, M.; Sharma, V.; Synnaeve, G.; Xu, H.; Jégou, H.; Mairal, J.; Labatut, P.; Joulin, A.; and Bojanowski, P. 2024. DINOv2: Learning Robust Visual Features without Supervision. *Trans. Mach. Learn. Res.*
- Peng, S.; and Genova, K. 2023. OpenScene: 3D Scene Understanding With Open Vocabularies. In *CVPR*, 815–824.
- Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*, 1532–1543.
- Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*, 77–85.
- Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; et al. 2021. Learning Transferable Visual Models from Natural Language Supervision. In *ICML*, 8748–8763.
- Ren, A. Z.; Clark, J.; Dixit, A.; Itkina, M.; Majumdar, A.; and Sadigh, D. 2024. Explore until Confident: Efficient Exploration for Embodied Question Answering. In *Robotics: Science and Systems (RSS)*, Article 89.
- Renz, M.; Günther, M.; Igelbrink, F.; Lima, O.; and Atzmueller, M. 2026. ExPrIS: Knowledge-Level Expectations as Priors for Object Interpretation from Sensor Data. *Künstl. Intell.*
- Renz, M.; Igelbrink, F.; and Atzmueller, M. 2025. Integrating Prior Observations for Incremental 3D Scene Graph Prediction. In *ICMLA*.
- Rosinol, A.; Gupta, A.; Abate, M.; Shi, J.; and Carlone, L. 2020. 3D Dynamic Scene Graphs: Actionable Spatial Perception with Places, Objects, and Humans. *Robotics: Science and Systems (RSS)*.
- Shafiullah, N. M. M.; Paxton, C.; Pinto, L.; Chintala, S.; and Szlam, A. 2023. CLIP-Fields: Weakly Supervised Semantic Fields for Robotic Memory. In *Robotics: Science and Systems (RSS)*.
- Strader, J.; Hughes, N.; Chen, W.; Speranzon, A.; and Carlone, L. 2024. Indoor and outdoor 3D scene graph generation via language-enabled spatial ontologies. *IEEE Robot. Autom. Lett.*, 9(6): 4886–4893.
- Wald, J.; Avetisyan, A.; Navab, N.; Tombari, F.; and Niessner, M. 2019. RIO: 3D object instance re-localization in changing indoor environments. In *ICCV*.
- Wald, J.; Dharmo, H.; Navab, N.; and Tombari, F. 2020. Learning 3D semantic scene graphs from 3D indoor reconstructions. In *CVPR*, 3961–3970.
- Werby, A.; Huang, C.; Büchner, M.; Valada, A.; and Burgard, W. 2024. 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*.
- Wu, S.; Wald, J.; Tateno, K.; Navab, N.; and Tombari, F. 2021. SceneGraphFusion: Incremental 3D Scene Graph Prediction From RGB-D Sequences. In *CVPR*, 7515–7525.
- Zhao, X.; Ding, W.; An, Y.; Du, Y.; Yu, T.; Li, M.; Tang, M.; and Wang, J. 2023. Fast Segment Anything. *arXiv preprint arXiv:2306.12156*.