2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

MA3DSG: Multi-Agent 3D Scene Graph Generation for Large-Scale Indoor Environments

Yirum Kim[1], Jaewoo Kim[1], Ue-Hwan Kim[1†]

***Abstract*— Current 3D scene graph generation (3DSGG) approaches heavily rely on a single-agent assumption and small-scale environments, exhibiting limited scalability to real-world scenarios. In this work, we introduce Multi-Agent 3D Scene Graph Generation (MA3DSG) model, the first framework designed to tackle this scalability challenge using multiple agents. We develop a training-free graph alignment algorithm that efficiently merges partial query graphs from individual agents into a unified global scene graph. Leveraging extensive analysis and empirical insights, our approach enables conventional single-agent systems to operate collaboratively without requiring any learnable parameters. To rigorously evaluate 3DSGG performance, we propose MA3DSG-Bench—a benchmark that supports diverse agent configurations, domain sizes, and environmental conditions—providing a more general and extensible evaluation framework. This work lays a solid foundation for scalable, multi-agent 3DSGG research.**

I. Introduction

3D scene graph generation (3DSGG) serves as a cornerstone for comprehensive high-level 3D scene understanding. By detecting objects and describing their relationships via predicates, it provides valuable context for diverse tasks such as image captioning [1], [2], image generation [3], change detection [4], navigation [5], [6], and task planning [7], [8].

Since the introduction of the 3DSGG benchmark [9], extensive research has primarily focused on improving *performance*—through enhanced relational reasoning [10], the integration of structured prior knowledge [11]–[13], and the adoption of open-vocabulary settings [14], [15]—while relatively little focus has been given to the *scalability* of the methods. As machine agents are increasingly deployed across a diverse and expanding set of real-world domains [16], [17], the ability to not only achieve strong performance but also sustain it at scale [18], [19] has become a necessity. This raises a fundamental question: "***Are current 3DSGG methods scalable?***"

Our findings, as illustrated in Figure 1, demonstrate that contemporary 3DSGG methods encounter significant scalability challenges—with runtimes up to ***4× longer*** (vs. single-agent methods) and data traffic up to ***98× heavier*** (vs. naive multi-agent methods) compared to our proposed MA3DSG. We attribute these scalability issues to two key limitations: (1) the prevalent reliance on single-agent paradigms; and (2) benchmarks biased toward constrained environments.

Lastly, to enable rigorous evaluation of both performance and scalability, we introduce ***MA3DSG-Bench***—a flexible and extensible benchmark that supports diverse agent configurations (single- and multi-agent), varying scales (ranging from 1 to 47 rooms), and scene dynamics (static and long-term condition). In contrast to prior 3DSGG benchmarks [12], [14] that process each room in 3RScan [20] independently using a single agent, our benchmark treats all reference rooms as a unified navigable space, enabling parallel exploration by multiple agents. Furthermore, we incorporate rescan sequences [20] to reflect realistic long-term environmental changes. By facilitating joint perception and temporal context modeling in dynamic multi-agent environments, our MA3DSG-Bench sets a new standard in the field.

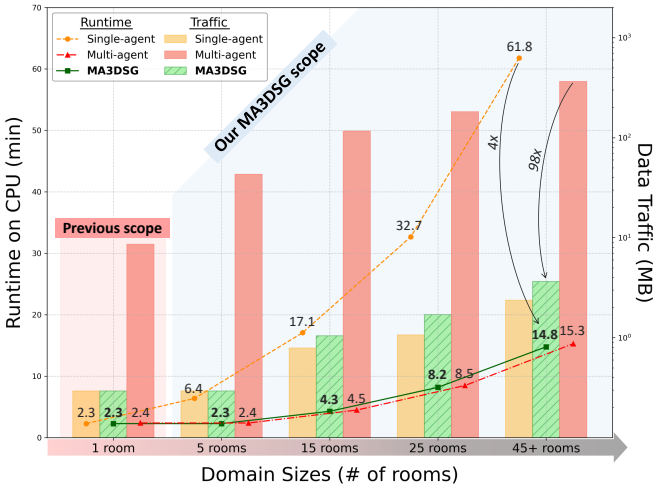

Fig. 1. Comparison of Runtime and Data Traffic. Our MA3DSG (*14.8 min, 3.7 MB*) runs 4× faster than single-agent system (SGFN, *61.8 min*), and uses 98× less data traffic than multi-agent system (SGFN + SG-PGM, *364.1 MB*) in extremely large-scale environments. Unlike the single-agent baselines and MA3DSG, which were only executed on CPUs, the multi-agent baselines utilized GPUs on the backend due to their model complexity.

To summarize, our contributions are as follows:

1) **Problem Formulation.** We extend the 3DSGG task to multi-agent, large-scale settings. To the best of our knowledge, this is the first holistic effort to address the scalability challenges in 3DSGG research.
2) **Model Design.** We propose MA3DSG built for the 3D semantic scene graph domain—featuring an efficient graph alignment algorithm. MA3DSG demonstrates strong scalability across diverse domain sizes while ensuring fast and robust performance.
3) **Benchmark Setup.** We introduce a comprehensive MA3DSG-Bench that expands previous single-agent, small-scale-only 3DSGG benchmarks. This contribution provides a solid foundation to guide and inspire future research in scalable 3DSGG.

[1] All authors are with the Department of AI Convergence, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, Republic of Korea. {kimyirum,kjw01124}@gm.gist.ac.kr
[†] Corresponding author: Ue-Hwan Kim uehwan@gist.ac.kr

II. Related Work

A. 3D Scene Graph Generation

Current research on 3DSGG can broadly fall into two areas based on their perspectives on 3D scene representations: hierarchical 3D scene graphs and 3D semantic scene graphs—the focus of our work.

1) Hierarchical 3D Scene Graphs: Hierarchical 3D scene graphs organize entities such as buildings, rooms, objects, and cameras into a unified structure [9]. Several works have enlarged the estimation of such hierarchical 3D scene graphs to large-scale environments. For example, 3D dynamic scene graphs handle scenes with moving agents [21], Kimera builds 3D dynamic scene graphs from visual-inertial data [22], and Hydra incrementally constructs the layers of a hierarchical scene graph [23]. While these approaches [9], [21]–[25] provide a compact and efficient representation of 3D scene environments, they primarily indicate the existence of entities for expressing relationships rather than capturing their detailed semantic nuances of how those objects are configured and interact. As a result, they lack the expressive power required for diverse downstream tasks such as task planning [7], scene change detection [4], and manipulation of 3D scenes [26].

2) 3D Semantic Scene Graphs: In contrast to hierarchical 3D scene graphs, 3D semantic scene graphs focus on inter-object semantics and contextual interactions [13], [27]. Following the development of the 3D scene graph dataset [28] built on top of 3RScan [20], 3D semantic scene graph generation from reconstructed point clouds has emerged: GNN-based analysis [29], performance enhancement through prior knowledge [11], instance embedding based generation [30], language-based contrastive pre-training [31], and visual-linguistic semantics assisted training [12]. Another stream of work has proposed to incrementally construct a 3D semantic scene graph from image sequences and depth data [32], [33].

By explicitly delineating spatial relationships between objects and their surroundings, these graphs enhance a variety of downstream applications, including 3D point registration [34], 3D scene reconstruction [26], change detection [4], and task planning [7]. Despite these advancements, conventional approaches hardly address the inherent scalability limitations of the 3D scene graph generation process. In this work, we concentrate on 3D semantic scene graph research and tackle the critical yet underexplored challenge—facilitating a *scalable* 3D semantic scene graph generation for large-scale environments.

B. Multi-Agent System

Multi-agent systems have been extensively studied for their potential to enhance the robustness and scalability of single-agent frameworks by harnessing the synergistic capabilities of a swarm. In the context of simultaneous localization and mapping (SLAM), recent works [35], [36] enable agents to explore collaboratively through exchanging sensor data and jointly optimizing 3D maps. Similarly, previous studies [25], [37] present multi-agent cooperation frameworks for fusing data from heterogeneous robots in hierarchical 3D scene graph generation. Furthermore, a diverse range of approaches, including multi-domain cooperation [38], probabilistic occupancy mapping [39], optimized cooperative exploration and communication [35], [40], and distilled collaboration graphs [41], have also demonstrated the versatility of multi-agent systems. Building on these notable advancements, we extend multi-agent methodologies to the realm of a 3D semantic scene graph framework. To the best of our knowledge, our work represents the first comprehensive effort to develop a robust multi-agent 3DSGG system, accompanied by a detailed and rigorous benchmark.

C. Graph Alignment

Graph alignment aims to maximize structural and attribute consistency across graphs, and has traditionally been formulated as a graph isomorphism or quadratic assignment problem [42]–[44]. However, classical approaches often suffer from high complexity as the graph size increases [42], [45], fail to capture edge structures [46], or require equal-sized graphs [47]. Recently, deep learning-based methods [48], [49] learn flexible matching functions and robust node representations. In the domain of 3D scene understanding, partial graph matching using geometric and semantic features improves alignment robustness and downstream performance [34], [49]. Nevertheless, current learning-based techniques are often limited by high training complexity, computational overhead, and slow inference speed, which restricts their applicability in large-scale or real-time scenarios.

III. Methodology

A. Problem Formulation

For each agent $k \in \{1,\dots,K\}$, we define a collection of RGB-D observation sequences as $\mathcal{S}^k = \{s_r^k\}_{r=1}^{R_k}$, where R_k denotes the number of rooms visited by agent k. Each sequence s_r^k, which corresponds to the scanned data of the r-th room explored by agent k, is a temporally ordered sequence of frames as $s_r^k = \left(I_{r,1}^k, I_{r,2}^k, \dots, I_{r,T_r^k}^k\right)$, where $I_{r,t}^k$ denotes the RGB-D frame captured at time step t and T_r^k is the total number of frames recorded in room r by agent k. Based on these observations, the objective of MA3DSG is to let each agent incrementally construct the local 3D scene graph $\mathcal{G}_r^k = (\mathcal{V}_r^k, \mathcal{E}_r^k)$, where $\mathcal{V}_r^k$ denotes the set of semantic entities (nodes) and $\mathcal{E}_r^k$ denotes the relationships (edges) between entities for room r. These partial graphs are integrated into a unified global scene graph $\mathcal{G}' = \bigcup_{r=1}^{R} \mathcal{G}_r^k$ in a decentralized and collaborative manner. We address the challenges posed by dynamic scenes and the integration of conflicting partial graphs, using a function $f : G_r^1 \times G_r^2 \times \cdots \times G_r^n \to \mathcal{G}'$ with G_r^k defined in its own frame $\mathcal{F}_r^k$. Notably, the scene graph is updated over time as $\mathcal{V}_r(t+1) = \mathcal{V}_r(t) + \Delta\mathcal{V}$ and $\mathcal{E}_r(t+1) = \mathcal{E}_r(t) + \Delta\mathcal{E}$, thereby ensuring the adjustment to long-term dynamic objects of the environment.

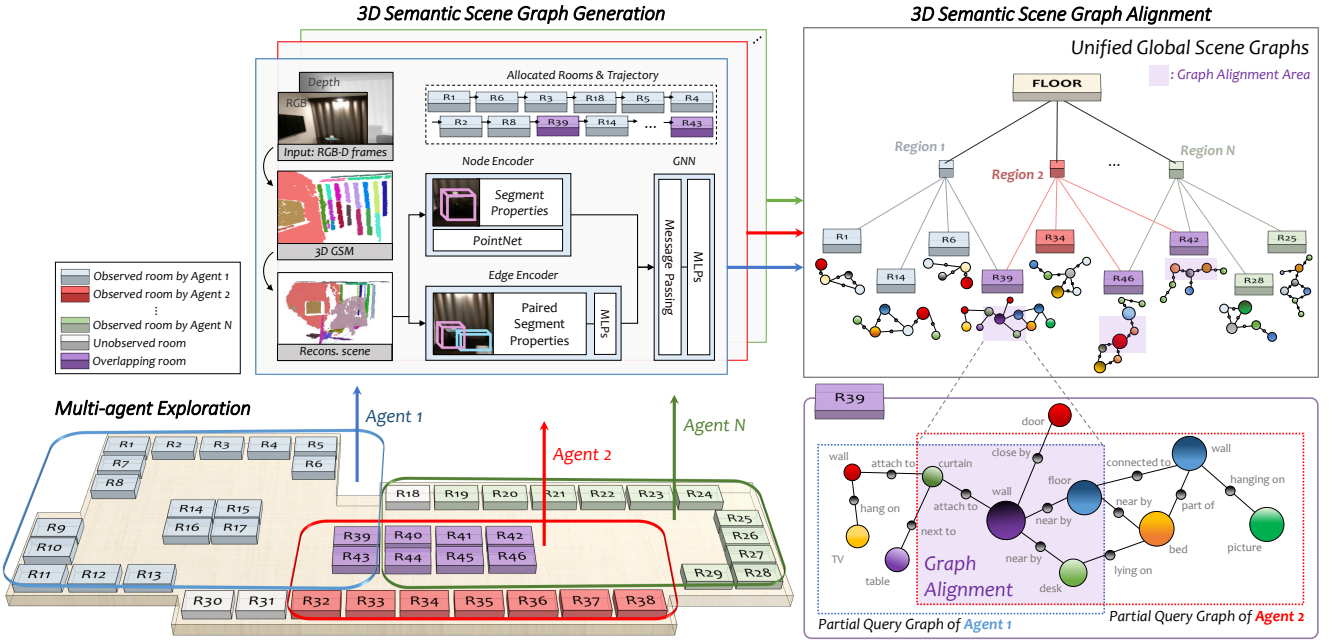

Fig. 2. The overall architecture of the proposed MA3DSG. Each agent incrementally generates 3D semantic scene graphs in a large-scale environment. The framework consists of multi-agent exploration, 3D semantic scene graph generation, and graph alignment, where agents collaboratively construct and integrate local scene graphs into a unified global representation.

B. Overview

We propose a decentralized multi-agent framework for 3D scene graph generation, addressing the need for scalable and efficient scene understanding in large-scale environments. Unlike single-agent approaches constrained to small domains, MA3DSG utilizes multiple agents to collaboratively explore diverse regions, as shown in Figure 2. It consists of three core components: (1) *Multi-Agent Exploration*, enabling distributed coverage of large spaces; (2) *3D Semantic Scene Graph Generation*, where agents incrementally build local 3D semantic scene graphs; and (3) *3D Semantic Scene Graph Alignment*, a lightweight algorithm that integrates local graphs into a unified global representation. By sharing information through overlapping exploration, MA3DSG enhances completeness and reduces overhead—increasing its utility in real-world scenarios.

C. 3D Semantic Scene Graph Generation

1) 3D Global Segmentation Map (3D GSM): Each agent performs incremental geometric segmentation [50] on the input RGB-D sequence to generate the 3D global segmentation map (GSM). 3D GSM consists of multiple segments $U = \{u_1, u_2, \ldots, u_n\}$, with each segment containing the point cloud $P_i = \{\mathrm{p}_i \mid \mathrm{p}_i \in \mathbb{R}^3\}$. With each new incoming frame, 3D GSM is updated by incorporating new segments or eliminating old ones. Each segment u_i is characterized by multiple properties: the centroid $\bar{\mathrm{p}}_i \in \mathbb{R}^3$, the standard deviation of the points σ_i, the size of the axis-aligned bounding box $\mathrm{b}_i = (b_x, b_y, b_z) \in \mathbb{R}^3$, the maximum length $l_i = max\{b_x, b_y, b_z\}$, and the volume $v_i = b_x \cdot b_y \cdot b_z$. Subsequently, segments for each instance are integrated into the reconstructed scene, with each instance treated as a node.

2) Feature Graph: Leveraging the seminal SGFN [32], the node encoder generates node features v_i by extracting a latent feature vector $E(\mathrm{P}_i)$ of the point cloud using PointNet [51]. To address scale insensitivity from normalization, spatial-invariant properties are concatenated to $E(\mathrm{P}_i)$. The edge encoder, consisting of three multi-layer perceptrons (MLPs), computes edge features e_{ij} for any two neighboring nodes i and j $(i \neq j)$ by processing their relative spatial properties. Formally, the node/edge features are defined as:

$$v_i = [E(\mathrm{P}_i), \sigma_i, ln(\mathrm{b}_i), ln(v_i), ln(l_i)], \tag{1}$$

$$e_{ij} = f_s([\Delta\bar{\mathrm{p}}_{ij}, \Delta\sigma_{ij}, \Delta\mathrm{b}_{ij}, \ln\left(\frac{v_i}{v_j}\right), \ln\left(\frac{l_i}{l_j}\right)]), \tag{2}$$

where $[\cdot]$ denotes a concatenation function and f_s represents MLPs. The feature-wise attention network (*FAN*) [32] is employed, where node and edge features are robustly updated in the message passing layer as follows:

$$v_i^{m+1} = f_v\left(v_i^m, \max_{j \in \mathcal{N}(i)} \left(FAN(v_i^m, e_{ij}^m, v_j^m)\right)\right), \tag{3}$$

$$e_{ij}^{m+1} = f_e\left(v_i^m, e_{ij}^m, v_j^m\right), \tag{4}$$

where f_v and f_e denote MLPs and $\mathcal{N}(i)$ denotes the set of neighbors of node i.

D. 3D Semantic Scene Graph Alignment

Merging a stored 3D scene graph with a newly generated graph is a significant challenge due to the complexity of the subgraph matching process, which arises from variations in graph size and structural differences. The task is inherently NP-hard, and its complexity is further exacerbated by the dynamic nature of objects in our test benchmark scenarios, where positions and states are subject to change. To address this, we propose a novel incremental graph alignment

algorithm that seamlessly integrates newly generated 3D scene graphs with existing data. The algorithm enhances robustness by aligning the new graphs with stored data, updating nodes and edges that have changed, and leveraging prior information to infer unscanned regions. The proposed graph merging consists of two key stages:

- **Graph Alignment**: The agent identifies the intersection subgraph between the newly generated query graph and the existing reference graph.
- **Graph Update**: The agent updates existing node/edge attributes or infers graphs for newly encountered regions as it continues along its trajectory.

*1) **Graph Alignment**:* A partial query graph and a reference graph containing label information are given. When the query graph G_q contains more than six nodes, an anchor node is randomly selected as the starting point for the search (line 12). The process begins by identifying nodes in the reference graph that share the same label as the anchor node. For each candidate node, the search expands iteratively by traversing neighboring nodes in the query graph (line 13) and attempting to find corresponding matches in the reference graph. This search recursively identifies triplet (node–edge–node) matches to maximize the alignment between the two graphs (line 1-9). The objective is to extract the intersection subgraph that best corresponds to the query graph. If the intersection subgraph exceeds the alignment threshold length θ_{len}, the corresponding nodes and edges are merged, and the 3D information of the reference graph is updated (line 15-17). Otherwise, if the match size falls below θ_{len}, the agent adds the query graph to the reference graph as new nodes and edges (line 18).

*2) **Graph Update**:* If the partial query graph and the reference graph are aligned, the remaining triplets in the reference graph G_r are updated in one of three ways, based on the newly recognized object O in G_q to improve the accuracy of the 3D information:

(i) Matching Node: If the centroid distance between object O and an existing node v is below threshold θ_{dis}, the Intersection over Union (IoU) between their bounding boxes exceeds θ_{bbox}, and their labels are identical, then O and v are treated as the same. The 3D attributes of v are refined by updating its bounding box to the union of bounding boxes of O and v, and the existing edge e is replaced with the corresponding edge from G_q.

(ii) Conflicting Label: If the centroid distance is less than θ_{dis} and the IoU exceeds θ_{bbox}, but the labels differ, then the existing node v is replaced with a new node corresponding to O. The new node inherits the class label and 3D spatial properties of O, including the bounding box coordinates, and the edge e connected to v is also updated accordingly.

(iii) New Node: If the centroid distance exceeds θ_{dis}, O is regarded as novel and inserted into the scene graph as a new node. The corresponding edge structure from G_q is also added to maintain relational consistency.

Algorithm 1 Graph Alignment Algorithm for Agent k

Input: Query Graph G_q, Reference Graph G_r
Output: Updated 3D Scene Graph G'_r

1: **procedure** GRAPHSEARCH($q, G_q, G_r, map, visit$)
2: $visit \leftarrow visit \cup \{q\}$
3: **for each** neighbor u of q in G_q **do**
4: **if** $u \notin visit$ **then**
5: **for each** v in neighbors of $map[q]$ in G_r **do**
6: **if** $G_q[u].label = G_r[v].label$ **then**
7: $map[u] \leftarrow v$
8: $map \leftarrow$ GraphSearch($u, G_q, G_r, map, visit$)
9: **return** map

10: **procedure** SCENEGRAPHUPDATE(G_q, G_r)
11: $map, visit \leftarrow Init()$
12: $anchors \leftarrow SelectRandomNodes(G_q)$
13: **for** node N_{anc} in $anchors$ **do**
14: $map_{max} \leftarrow$ GraphSearch($N_{anc}, G_q, G_r, map, visit$)
15: **if** $len(map_{max}) >$ threshold θ_{len} **then**
16: $G'_r \leftarrow$ GraphUpdate(G_q, G_r) ▷ update G_r
17: **return** G'_r
18: $G'_r \leftarrow G_r.Add(G_q)$ ▷ just add G_q as new graph
19: **return** G'_r

IV. MA3DSG-BENCH

A. Evaluation Scenarios

MA3DSG-Bench evaluates performance under two standard scenarios.

*1) **Static Collaborative Perception (SCP)**:* Multiple agents collaboratively operate within a large-scale indoor environment with no dynamic objects. Multiple agents simultaneously explore distinct regions and incrementally construct a unified 3D scene graph. This scenario evaluates the system's capability to accurately and consistently integrate spatial and semantic information from diverse viewpoints into a cohesive graph representation.

*2) **Long-term Dynamic Collaborative Perception (LDCP)**:* This scenario introduces temporal dynamics, where changes occur over an extended period. Initially, agents construct a scene graph through collaborative exploration, similar to SCP. However, upon revisiting the same location after a significant time lapse, agents encounter changes such as objects that have moved, appeared, or disappeared entirely. Since such scenarios frequently occur in real-world indoor environments, LDCP assesses the system's ability to detect and reconcile temporal inconsistencies by updating nodes and edges, thereby enhancing real-world applicability.

B. Evaluation Dataset

To evaluate the scalability of MA3DSG, we reformulate the 3RScan [20] and 3DSSG [28] datasets by assuming that all 47 room scenes in the test set can belong to *a single, unified domain*, rather than evaluating them independently as in prior works. This unified setting, designed to evaluate robustness under diverse layouts, serves as a valid basis for assessing instance-level perception accuracy within rooms.

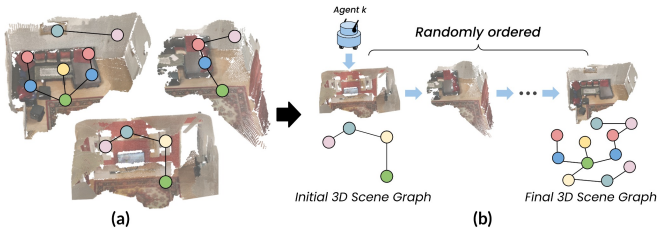

Fig. 3. Unified domain evaluation. (a) Prior works treat each explored scene separately. (b) A newly annotated final 3D scene graph reflects temporal changes from randomly ordered visits for the LDCP scenario.

TABLE I

THE NUMBER OF STATIC/DYNAMIC OBJECTS.

# of Instances	Static	Dynamic		
		moved	*removed*	*changed*
1,588	1,110	214	70	194

TABLE II

UNIFIED 3D SEMANTIC SCENE GRAPH SIZES IN THE EACH SETTING.

	SCP	LDCP
# of Nodes	1,588	1,518
# of Edges	5,546	5,054

As shown in Figure 3(a), although originating from the same physical environment, the scenes were previously treated as independent, with separate 3D scene graph generation and evaluation. Within the unified domain, as illustrated in Figure 3(b), agent k incrementally constructs a 3D scene graph by exploring randomly ordered $\mathcal{S}^k$ in our benchmark. To simulate realistic multi-agent deployment, each agent's trajectory is randomly generated while controlling the *overlap ratio*—defined as the fraction of one agent's trajectory intersecting another's—to maintain a balanced and realistic coverage distribution across agents. In particular, we utilize the rescans of each room, where object configurations are rearranged, to generate a newly annotated final 3D scene graph that captures both spatial and temporal changes—including 478 dynamic objects, as shown in Table I.

C. Metrics

1) Accuracy: Following prior works [32], [33], we evaluate our method on triplet, object, and predicate prediction tasks. While prior work primarily reports recall@k, we also include precision and F1, as high recall alone does not ensure meaningful graph construction—excessive relationship predictions can introduce noise, reducing interpretability. Moreover, whereas previous studies focus only on semantic labels, we also incorporate spatial accuracy by considering the object's center position and 3D bounding box IoU—ensuring alignment between the generated 3D scene graph and the environment's actual geometry.

2) Efficiency: We assess the efficiency of the 3D scene graph generation system by computing the total graph alignment time, the whole scenario completion time, and the per-agent data traffic, defined as the mean total volume of data transmitted per agent.

V. EXPERIMENTS

A. Baselines

We compare our **MA3DSG** with two types of baselines:

- **Single-Agent Baselines**: Conventional 3DSGG research includes 3DSSG [28] and SGFN [32]—the prevalent setting. In these methods, a single agent is responsible for generating the complete 3D scene graph. Unlike other baselines, the 3DSSG processes the entire point cloud at once, making total runtime measurement inappropriate for evaluating incremental 3D scene graph generation.
- **Multi-Agent Baseline**: Due to the challenges of constructing multi-agent systems, the literature lacks established baselines for direct comparison. To address this, we introduce strong baselines by combining SGFN with recent modules: SGAligner [34] and SG-PGM [52], denoted as *SGFN+SGAligner* and *SGFN+SG-PGM*.

B. Comparative Studies

In all experiments, we set five agents with an overlap ratio of 0.2 for the multi-agent setups. For MA3DSG, several thresholds were determined empirically through ablations, where we set θ_{dis} to 1.5 meters, θ_{len} to 3, and θ_{bbox} to 0.4.

*1) **Static Collaborative Perception (SCP)**:* Table III compares MA3DSG against single-agent and multi-agent baselines in the SCP setting, demonstrating that it achieves performance comparable to both. Across all domain sizes, MA3DSG maintains R@1, P@1, and F1@1 on par with those of the single-agent SGFN, with no substantial decline. Specifically, MA3DSG exhibits deviations in triplet F1@1 ranging from +1.5/-1.3%, in object F1@1 from -3.1/-4.4%, and in predicate F1@1 from +0.8/-4.4%. MA3DSG also operates ***2.8× to 4.2× faster*** than SGFN, with the advantage increasing as domain size scales. This stability stems from MA3DSG's incremental updates, which preserve prior information and leverage multi-agent observations to address unscanned regions—unlike full graph replacement in single-agent approaches. However, this mechanism introduces potential errors, such as conservative predictions from inconsistent updates, missed relationships, or noise accumulation over iterative updates, especially in large domains. In contrast, SGFN+SG-PGM exhibits a more pronounced performance gap relative to SGFN with deviations in Triplet F1@1 ranging from -1.5/-6.5%, in object F1@1 from +0.5/-1.7%, and in predicate F1@1 from -1.9/-10.6%.

*2) **Long-term Dynamic Collaborative Perception (LDCP)**:* Table IV shows quantitative results for the LDCP setting, which introduces temporal inconsistencies requiring adaptive refinement and continuous 3D scene graph updates. When comparing SGFN and MA3DSG, the deviations range up to +1.9/-0.4% in triplet F1@1, +9.6/+0.3% in object F1@1, and +2.4/-0.5% in predicate F1@1. MA3DSG operated ***3.4× to 4.1× faster*** in terms of processing efficiency, with greater gains as domain size increased. These results underscore MA3DSG's ability to adeptly manage dynamic scene changes by integrating

TABLE III

QUANTITATIVE EVALUATION OF ACCURACY AND EFFICIENCY UNDER THE SCP SETTING.

Method	Domain Size (# of rooms)	Triplet			Object			Predicate			Traffic. (MB)	Align. (sec)	Total. (min)
		R@1	P@1	F1@1	R@1	P@1	F1@1	R@1	P@1	F1@1			
Single-agent approach													
3DSSG	5	3.1	3.2	3.1	27.1	32.5	29.5	15.2	19.5	17.0	-	-	-
	15	16.8	15.5	16.1	31.5	37.7	34.4	26.3	39.2	31.5	-	-	-
	25	18.6	15.5	16.9	32.6	36.5	34.4	29.7	37.9	33.3	-	-	-
	47	14.8	9.0	11.2	34.0	33.4	33.7	25.9	26.8	26.3	-	-	-
SGFN	5	19.6	7.6	10.9	52.1	33.1	40.5	25.9	23.4	24.6	-	-	6.4
	15	20.4	9.5	13.0	51.7	30.7	38.5	25.7	26.3	26.0	-	-	17.1
	25	24.3	10.5	14.7	55.8	30.7	39.6	30.5	27.7	29.0	-	-	32.7
	47	26.4	9.6	14.1	56.2	29.4	38.6	32.0	25.8	28.6	-	-	61.8
Multi-agent approach													
SGFN + SGAligner	5	14.7	5.4	7.9	50.0	34.3	40.7	18.9	16.4	17.6	43.2	11.7	2.5
	15	10.8	4.7	6.6	46.1	30.5	36.7	14.6	15.8	15.2	116.2	35.1	4.9
	25	16.2	6.9	9.7	51.1	31.7	39.1	20.9	20.2	20.5	181.7	56.5	9.1
	47	22.5	8.8	12.7	50.0	31.3	38.5	27.5	24.6	25.9	364.2	107.1	16.6
SGFN + SG-PGM	5	13.8	5.2	7.5	52.1	33.8	41.0	19.6	17.3	18.4	43.1	3.50	2.4
	15	10.6	4.6	6.5	46.7	30.4	36.8	14.8	16.0	15.4	116.1	10.5	4.5
	25	16.2	6.9	9.7	51.3	31.0	38.6	21.2	20.4	20.8	181.6	16.9	8.5
	47	22.5	8.8	12.6	50.0	31.0	38.3	28.1	25.4	26.7	364.1	32.1	15.3
MA3DSG (Ours)	5	21.1	7.9	11.5	47.9	30.7	37.4	25.9	24.0	24.9	**0.3**	**0.0**	**2.3**
	15	17.3	8.9	11.7	51.4	25.5	34.1	21.1	22.1	21.6	**1.0**	**0.01**	**4.3**
	25	25.7	11.8	16.2	54.9	27.2	36.4	31.2	28.5	29.8	**1.7**	**0.01**	**8.2**
	47	24.2	9.5	13.7	55.8	25.6	35.1	27.7	22.2	24.6	**3.7**	**0.02**	**14.8**

* For each domain size, the top three F1 scores for each metric group are highlighted using three levels of color intensity, while the best efficiency result is shown in bold.

TABLE IV

QUANTITATIVE EVALUATION OF ACCURACY AND EFFICIENCY UNDER THE LDCP SETTING.

Method	Domain Size (# of rooms)	Triplet			Object			Predicate			Traffic. (MB)	Align. (sec)	Total. (min)
		R@1	P@1	F1@1	R@1	P@1	F1@1	R@1	P@1	F1@1			
Single-agent approach													
3DSSG	5	3.1	3.2	3.1	24.0	28.7	26.1	12.2	15.8	13.8	-	-	-
	15	11.2	9.4	10.2	27.7	32.8	30.1	18.3	26.6	21.7	-	-	-
	25	13.3	9.9	11.4	28.1	30.8	29.4	22.1	26.8	24.2	-	-	-
	47	10.9	6.0	7.7	29.5	28.6	29.1	20.6	20.3	20.5	-	-	-
SGFN	5	4.2	1.4	2.1	44.8	28.5	24.8	9.2	8.2	8.7	-	-	22.2
	15	9.2	3.6	5.2	44.6	26.2	33.0	14.2	13.8	14.0	-	-	69.7
	25	12.1	4.3	6.3	47.9	25.8	33.5	18.3	15.3	16.7	-	-	95.8
	47	14.2	4.3	6.6	47.6	24.6	32.4	19.5	14.6	16.7	-	-	166.7
Multi-agent approach													
SGFN + SGAligner	5	6.1	1.9	2.9	44.8	29.3	35.4	14.6	12.5	13.5	33.1	7.1	124.9
	15	8.9	3.9	5.4	36.9	27.3	31.4	17.5	20.7	18.9	408.1	124.7	18.5
	25	8.8	3.3	4.8	41.6	25.8	31.8	17.5	16.1	16.8	575.6	185.1	25.0
	47	11.9	4.3	6.4	39.6	25.4	31.0	21.2	18.8	19.9	1013.2	350.6	46.8
SGFN + SG-PGM	5	5.1	1.6	2.4	46.9	29.0	35.9	15.4	13.4	14.3	124.8	9.91	6.7
	15	8.6	3.8	5.3	37.6	27.3	31.6	17.7	21.0	19.2	408.0	37.3	17.0
	25	8.6	3.3	4.7	42.0	25.8	31.9	17.7	16.3	17.0	575.5	55.4	22.7
	47	11.1	4.1	6.0	39.4	25.3	30.8	20.1	17.9	19.0	1013.1	104.9	42.7
MA3DSG (Ours)	5	8.1	2.6	4.0	43.8	28.4	34.4	11.0	9.6	10.2	**1.0**	**0.03**	**6.5**
	15	10.6	4.2	6.0	44.9	26.4	33.3	16.7	16.2	16.4	**4.5**	**0.13**	**16.4**
	25	13.4	4.7	7.0	49.1	25.8	33.9	18.8	15.4	16.9	**6.1**	**0.19**	**21.8**
	47	13.1	4.0	6.2	48.3	25.0	33.0	18.9	14.1	16.2	**11.6**	**0.37**	**41.0**

* For each domain size, the top three F1 scores for each metric group are highlighted using three levels of color intensity, while the best efficiency result is shown in bold.

multi-agent observations and refining graphs incrementally. In contrast, the comparison between SGFN+SG-PGM and SGFN shows F1@1 deviations ranging from +0.3/-1.6% in triplet, +11.1/-1.6% in object, and from +5.6/+0.3% in predicate. While SGFN+SG-PGM leverages multi-agent collaboration to achieve faster execution than SGFN, its dependency on point cloud registration leads to higher processing time and computational cost than MA3DSG.

C. Efficiency Analysis

1) Alignment Time: MA3DSG substantially reduces computational overhead compared to multi-agent baselines. It achieves consistently low alignment latency—0.02 seconds in SCP and 0.37 seconds in LDCP—even in large-scale domains. In contrast, SGFN+SG-PGM incurs significantly higher alignment costs, highlighting the efficiency of our multi-agent communication and scene graph integration. Notably, MA3DSG performs inference entirely on CPU, emphasizing its lightweight and hardware-efficient design.

2) Data Traffic: To assess the communication efficiency, we compute traffic differently depending on the agent configuration. For the multi-agent baselines, the communication cost includes both the graph data and the point cloud data exchanged among agents. In contrast, MA3DSG transmits only lightweight graph representations rather than full point cloud, resulting in substantially lower traffic overhead. MA3DSG reduces communication cost by a factor of *98.4*× in the SCP and *87.3*× in the LDCP—highlighting its strong scalability in extremely large-scale 3D environments.

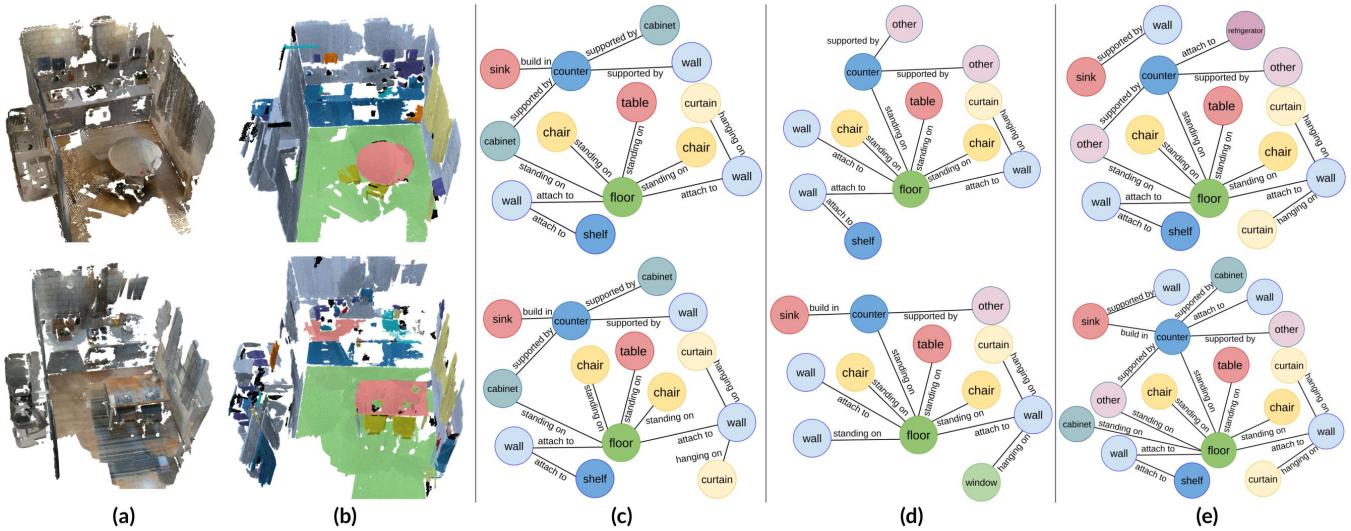

Fig. 4. Qualitative results of SGFN and MA3DSG. We visualize (a) incrementally scanned point clouds, (b) ground truth instance segmentation, (c) ground truth 3D Semantic Scene Graph, (d) SGFN-generated, and (e) MA3DSG-generated 3D Semantic Scene Graphs. For the same room, the upper row shows SCP results and the lower row shows LDCP results.

D. Qualitative Results

Figure 4 presents a qualitative comparison of 3D scene graphs under realistic, dynamic indoor environments, projected onto 2D for visualization purposes. The top row corresponds to the initial reference scan, while the bottom row shows the rescan data acquired after a time interval when an agent revisits the same room. As observed in (a) and (b), notable scene changes occur; the table changes from circular to rectangular, various furniture shifts, and lighting conditions vary due to open curtains. Notably, MA3DSG merges prior graphs when later agents encounter unscanned areas, preserving richer nodes such as shelves and cabinets, as well as edges that SGFN often overlooks.

VI. Conclusion

We introduced MA3DSG, a multi-agent framework for 3D scene graph generation that incrementally updates graphs, leveraging shared agent knowledge to achieve scalability and efficiency in large-scale settings. In the process, we developed MA3DSG-Bench, a novel benchmark tailored to evaluate 3DSGG scalability across diverse agent configurations, domain sizes, and dynamic conditions—surpassing prior single-agent, small-scale benchmarks. Together, these contributions establish a robust foundation for multi-agent 3DSGG research. Future work will refine the graph update mechanism, enhance robustness to scene variations, and optimize computational performance in dynamic environments. These efforts aim to advance scalable multi-agent 3DSGG systems, setting a new standard for the field.

ACKNOWLEDGMENT

This research was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-II220907); by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2022R1C1C1009989); and by the National Research Council of Science & Technology (NST) grant funded by the Korea government (MSIT) (No. GTL25041-000).

References

[1] J. Wang, W. Wang, L. Wang, Z. Wang, D. D. Feng, and T. Tan, "Learning visual relationship and context-aware attention for image captioning," *Pattern Recognition*, vol. 98, 2020.

[2] S. Chen, Q. Jin, P. Wang, and Q. Wu, "Say as you wish: Fine-grained control of image caption generation with abstract scene graphs," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9959–9968, 2020.

[3] L. Yang, Z. Huang, Y. Song, S. Hong, G. Li, W. Zhang, B. Cui, B. Ghanem, and M.-H. Yang, "Diffusion-based scene graph to image generation with masked contrastive pre-training," *ArXiv*, vol. abs/2211.11138, 2022.

[4] S. Looper, J. R. Puigvert, R. Y. Siegwart, C. Cadena, and L. M. Schmid, "3d vsg: Long-term semantic scene change prediction through 3d variable scene graphs," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8179–8186, 2022.

[5] Y. Li, Y. Ma, X. Huo, and X. Wu, "Remote object navigation for service robots using hierarchical knowledge graph in human-centered environments," *Intelligent Service Robotics*, vol. 15, pp. 459 – 473, 2022.

[6] S. Y. Gadre, K. Ehsani, S. Song, and R. Mottaghi, "Continuous scene representations for embodied ai," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14 829–14 839, 2022.

[7] C. Agia, K. M. Jatavallabhula, M. N. M. Khodeir, O. Mikvs'ik, V. Vineet, M. Mukadam, L. Paull, and F. Shkurti, "Taskography: Evaluating robot task planning over large 3d scene graphs," in *Conference on Robot Learning*, 2022.

[8] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. D. Reid, and N. Sünderhauf, "Sayplan: Grounding large language models using 3d scene graphs for scalable task planning," in *Conference on Robot Learning*, 2023.

[9] I. Armeni, Z.-Y. He, J. Gwak, A. Zamir, M. Fischer, J. Malik, and S. Savarese, "3d scene graph: A structure for unified semantics, 3d space, and camera," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5663–5672, 2019.

[10] C. Lv, M. Qi, X. Li, Z. Yang, and H. Ma, "Sgformer: Semantic graph transformer for point cloud-based 3d scene graph generation," in *AAAI Conference on Artificial Intelligence*, 2023.

[11] S. Zhang, S. Li, A. Hao, and H. Qin, "Knowledge-inspired 3d scene graph prediction in point cloud," in *Neural Information Processing Systems*, 2021.

[12] Z. Wang, B. Cheng, L. Zhao, D. Xu, Y. Tang, and L. Sheng, "Vl-sat: Visual-linguistic semantics assisted training for 3d semantic scene graph prediction in point cloud," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21 560–21 569, 2023.

[13] M. Feng, H. Hou, L. Zhang, Z. Wu, Y. Guo, and A. S. Mian, "3d spatial multimodal knowledge accumulation for scene graph prediction in point cloud," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9182–9191, 2023.

[14] S. Koch, N. Vaskevicius, M. Colosi, P. Hermosilla, and T. Ropinski, "Open3dsg: Open-vocabulary 3d scene graphs from point clouds with queryable objects and open-set relationships," *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14 183–14 193, 2024.

[15] L. Chen, X. Wang, J. Lu, S. Lin, C. Wang, and G. He, "Clip-driven open-vocabulary 3d scene graph generation via cross-modality contrastive learning," *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27 863–27 873, 2024.

[16] B. Vincent, P. M. Jacob, S. Pradeep, P. N. Fathima, R. Aswin, and M. Monachen, "An intelligent food serving robot prototype with android application for canteens," *2022 International Conference on Data Analytics for Business and Industry (ICDABI)*, pp. 565–569, 2022.

[17] Y. Okafuji, Y. Ozaki, J. Baba, J. Nakanishi, K. Ogawa, Y. Yoshikawa, and H. Ishiguro, "Behavioral assessment of a humanoid robot when attracting pedestrians in a mall," *International Journal of Social Robotics*, vol. 14, pp. 1731 – 1747, 2021.

[18] J.-P. Busch, L. Reiher, and L. Eckstein, "Enabling the deployment of any-scale robotic applications in microservice architectures through automated containerization*," *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 17 650–17 656, 2023.

[19] Y. Cao, R. Zhao, Y. Wang, B. Xiang, and G. Sartoretti, "Deep reinforcement learning-based large-scale robot exploration," *IEEE Robotics and Automation Letters*, vol. 9, pp. 4631–4638, 2024.

[20] J. Wald, A. Avetisyan, N. Navab, F. Tombari, and M. Nießner, "Rio: 3d object instance re-localization in changing indoor environments," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7657–7666, 2019.

[21] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans," *ArXiv*, vol. abs/2002.06289, 2020.

[22] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: From slam to spatial perception with 3d dynamic scene graphs," *The International Journal of Robotics Research*, vol. 40, pp. 1510 – 1546, 2021.

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

[24] U. Udugama, G. Vosselman, and F. Nex, "Mono-hydra: Real-time 3d scene graph construction from monocular camera input with imu," *ArXiv*, vol. abs/2308.05515, 2023.

[25] Y. Chang, N. Hughes, A. Ray, and L. Carlone, "Hydra-multi: Collaborative online construction of 3d scene graphs with multi-robot teams," *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10 995–11 002, 2023.

[26] H. Dhamo, F. Manhardt, N. Navab, and F. Tombari, "Graph-to-3d: End-to-end generation and manipulation of 3d scenes using scene graphs," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16 332–16 341, 2021.

[27] U.-H. Kim, J.-M. Park, T. jin Song, and J.-H. Kim, "3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents," *IEEE Transactions on Cybernetics*, vol. 50, pp. 4921–4933, 2019.

[28] J. Wald, H. Dhamo, N. Navab, and F. Tombari, "Learning 3d semantic scene graphs from 3d indoor reconstructions," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3960–3969, 2020.

[29] C. Zhang, J. Yu, Y. Song, and W. T. Cai, "Exploiting edge-oriented reasoning for 3d point-based scene graph analysis," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9700–9710, 2021.

[30] J. Wald, N. Navab, and F. Tombari, "Learning 3d semantic scene graphs with instance embeddings," *International Journal of Computer Vision*, vol. 130, pp. 630 – 651, 2022.

[31] S. Koch, P. Hermosilla, N. Vaskevicius, M. Colosi, and T. Ropinski, "Lang3dsg: Language-based contrastive pre-training for 3d scene graph prediction," *2024 International Conference on 3D Vision (3DV)*, pp. 1037–1047, 2023.

[32] S. cheng Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari, "Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7511–7521, 2021.

[33] S. cheng Wu, K. Tateno, N. Navab, and F. Tombari, "Incremental 3d semantic scene graph prediction from rgb sequences," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5064–5074, 2023.

[34] S. D. Sarkar, O. Miksik, M. Pollefeys, D. Barath, and I. Armeni, "Sgaligner: 3d scene alignment with scene graphs," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 21 927–21 937.

[35] P.-Y. Lajoie and G. A. Beltrame, "Swarm-slam: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems," *IEEE Robotics and Automation Letters*, vol. 9, pp. 475–482, 2023.

[36] A. Cramariuc, L. Bernreiter, F. Tschopp, M. Fehr, V. Reijgwart, J. I. Nieto, R. Y. Siegwart, and C. Cadena, "– a modular and multi-modal mapping framework," *IEEE Robotics and Automation Letters*, vol. 8, pp. 520–527, 2022.

[37] D. Zou, P. Tan, and W. Yu, "Collaborative visual slam for multiple agents: A brief survey," *Virtual Reality & Intelligent Hardware*, vol. 1, pp. 461–482, 2019.

[38] J. Yang, C.-K. Wen, X. Yang, J. Xu, T. tao Du, and S. Jin, "Multi-domain cooperative slam: The enabler for integrated sensing and communications," *IEEE Wireless Communications*, vol. 30, pp. 40–49, 2022.

[39] S. Sunil, S. Mozaffari, R. Singh, B. Shahrrava, and S. Alirezaee, "Feature-based occupancy map-merging for collaborative slam," *Sensors (Basel, Switzerland)*, vol. 23, 2023.

[40] P. Schmuck, "Multi-uav collaborative monocular slam," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3863–3870, 2017.

[41] Y. Li, S. Ren, P. Wu, S. Chen, C. Feng, and W. Zhang, "Learning distilled collaboration graph for multi-agent perception," *ArXiv*, vol. abs/2111.00643, 2021.

[42] Y. Aflalo, R. Kimmel, A. M. Bruckstein, E. Berton, E. Rivlin, and M. M. Bronstein, "On convex relaxation of graph isomorphism," *Proceedings of the National Academy of Sciences*, vol. 112, no. 10, pp. 2942–2947, 2015.

[43] J. Yan, X.-C. Yin, W. Lin, C. Deng, H. Zha, and X. Yang, "A short survey of recent advances in graph matching," *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, 2016.

[44] F. Emmert-Streib, M. Dehmer, and Y. Shi, "Fifty years of graph matching, network alignment and network comparison," *Information Sciences*, vol. 346, pp. 180–197, 2016.

[45] R. Raveaux, D. Conte, and A. Robles-Kelly, "Exact graph edit distance computation using a binary linear program," in *International Workshop on Graph-Based Representations in Pattern Recognition*. Springer, 2016, pp. 163–172.

[46] T. Yu, J. Yan, Y. Wang, W. Liu, and B. Li, "Generalizing graph matching beyond quadratic assignment model," in *Neural Information Processing Systems*, 2018.

[47] H. P. Maretic, M. E. Gheche, G. Chierchia, and P. Frossard, "Got: An optimal transport framework for graph comparison," in *Neural Information Processing Systems*, 2019.

[48] Z. Wang, Y. Liu, X. Wang, Z. Chen, Y. Wang, and Q. Huang, "Deep learning of partial graph matching via differentiable top-k," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1995–2005.

[49] W. Wang, H. Hao, H. Wang, Z. Zou, and W. Xing, *Graph Neural Network Methods and Applications in Scene Understanding*. Springer Nature, 2024.

[50] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6565–6574, 2017.

[51] C. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2016.

[52] Y. Xie, A. Pagani, and D. Stricker, "Sg-pgm: Partial graph matching network with semantic geometric fusion for 3d scene graph alignment and its downstream tasks," *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28 401–28 411, 2024.