MR-COGraphs: Communication-efficient Multi-Robot Open-vocabulary Mapping System via 3D Scene Graphs

Qiuyi Gu[1*], Zhaocheng Ye[1*], Jincheng Yu[1], Jiahao Tang[1], Tinghao Yi[23], Yuhan Dong[1], Jian Wang[1], Jinqiang Cui[4], Xinlei Chen[1], Yu Wang[1]

***Abstract*— Collaborative perception in unknown environments is crucial for multi-robot systems. With the emergence of foundation models, robots can now not only perceive geometric information but also achieve open-vocabulary scene understanding. However, existing map representations that support open-vocabulary queries often involve large data volumes, which becomes a bottleneck for multi-robot transmission in communication-limited environments. To address this challenge, we develop a method to construct a graph-structured 3D representation called COGraph, where nodes represent objects with semantic features and edges capture their spatial relationships. Before transmission, a data-driven feature encoder is applied to compress the feature dimensions of the COGraph. Upon receiving COGraphs from other robots, the semantic features of each node are recovered using a decoder. We also propose a feature-based approach for place recognition and translation estimation, enabling the merging of local COGraphs into a unified global map. We validate our framework using simulation environments built on Isaac Sim and real-world datasets. The results demonstrate that, compared to transmitting semantic point clouds and 512-dimensional COGraphs, our framework can reduce the data volume by two orders of magnitude, without compromising mapping and query performance. For more details, please visit our website at `https://github.com/efc-robot/MR-COGraphs`.**

I. Introduction

Multi-robot systems have emerged as powerful solutions for perception in large unknown environments [1]. The primary advantage of such systems lies in their ability to leverage distributed sensing and computing, enabling robots to share information and shorten task completion time. As the scale of these systems grows, the need to share environmental information to maintain system operations increases, necessitating data-efficient map representations for transmission.

Recent advances in visual foundation models (*e.g.*, SAM [2]) and vision-language models (*e.g.*, CLIP [3]) have enabled the development of open-vocabulary 3D map representations. Traditional semantic maps [4] [5] rely on predefined labels to describe the semantic information of the environment. They are closed-vocabulary since their labels are confined to the classes of objects annotated in the training datasets [6]. In contrast, open-vocabulary maps are not constrained by predefined classes and can understand new categories or words without retraining. This is achieved by extracting semantic feature vectors from images and projecting them into the 3D space [7]. Due to their strong semantic understanding ability and adaptability, open-vocabulary representations unlock new possibilities for various language-guided tasks such as object retrieval [8], language-based navigation [9] and manipulation [10].

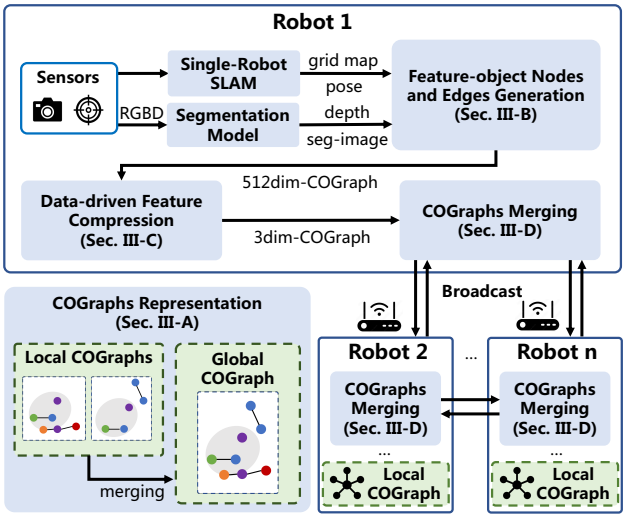


Fig. 1. Overview of the MR-COGraphs Framework.

However, current open-vocabulary 3D map representations demand significant data storage, which **becomes a communication bottleneck for multi-robot mapping systems.** Since each point in the map is associated with a high-dimensional feature vector, the map size grows rapidly as the robot continuously senses the environment [9]. For example, constructing only a small tabletop scene requires 1.3GB of data [11]. This data explosion makes it difficult for multiple robots to share and update maps in real time.

3D scene graphs (3DSGs) are favorable for semantic mapping in communication-constrained environments due to their compact and flexible representation of the scene [9]. They model the environment as graph structures, with nodes representing every object's attributes and edges encoding the relationships between these objects. Many studies have utilized 3DGSs for large-scale semantic mapping [12] [13], hierarchical 3D scene construction [14] [15], and robot task planning [9] [16]. However, these approaches are largely

[1] Tsinghua University, Beijing, China.
[2] Efort Intelligent Equipment, Wuhu, Anhui, China.
[3] University of Science and Technology of China, Hefei, Anhui, China.
[4] Pengcheng Laboratory, Shenzhen, Guangdong, China.
[*] Contributed equally to this work.

This research was supported by the National Natural Science Foundation of China (No.U19B2019, 62203257, M-0248), Tsinghua University Initiative Scientific Research Program, Tsinghua-Meituan Joint Institute for Digital Life, Beijing National Research Center for Information Science, Technology (BNRist), Beijing Innovation Center for Future Chips.

focused on single-robot systems. Although a few multi-robot mapping works [13] [17] have explored the collaborative construction of 3D scene graphs, **they do not consider open-vocabulary capabilities and have yet to address the critical challenge of reducing data size for efficient communication**.

To fulfill the requirements above, we propose a **C**ommunication-efficient **M**ulti-**R**obot **O**pen-vocabulary 3D Scene **Graphs**-based Mapping (MR-COGraphs) System with the following contributions:

- A data-efficient open-vocabulary 3D scene graph construction method, in which a data-driven feature encoder compresses the dimension of features in COGraphs without losing semantic information.
- A communication-efficient distributed multi-robot mapping system, leveraging the semantic features of local COGraphs shared among robots to achieve place recognition and translation estimation.
- We build and open source both simulated and real-world datasets to evaluate the performance of our system. Our framework can reduce 99.25%-99.98% of data volume during COGraph transmission.

As illustrated in Fig. 1, we propose a graph-structured open-vocabulary representation called COGraph (detailed in Section III-A). Firstly, each robot generates the nodes and edges of its local COGraph utilizing the output of the SLAM and the segmentation model (detailed in Section III-B). Then a data-driven lightweight feature encoder (detailed in Section III-C) is employed to resize the 512-dimensional semantic features of nodes into 3 dimensions. When receiving local COGraphs from other robots, the robot performs place recognition and translation estimation to merge local COGraphs into a global COGraph (detailed in Section III-D). Afterward, experimental results are presented in Section IV. Section V concludes this work and suggests future research directions.

II. Related work

A. 3D Scene Graphs

The concept of 3D scene graphs is first introduced in Armeni et al. [14], where the authors propose a semi-manual method to extract buildings, rooms, objects, and cameras from the environment, creating a multi-layer graph structure. This approach abstracts the environment into nodes and edges, where each node can encompass multiple attributes, and edges represent spatial relationships and hierarchies. Therefore, this representation is highly flexible, allowing for adjustments in its complexity and data volume.

There has been much progress in constructing 3DSGs with closed vocabulary [12] [15] [18]. Hydra [12] leverages HRNet [19] as a pre-trained model to obtain semantic labels. It adds a mesh layer to enable the online generation of room and building nodes. In StructNav [15], a robot employs visual SLAM along with MaskRCNN [6] to develop a structured representation, and semantics are then integrated into geometry-based frontiers to facilitate object-goal navigation.

For open-vocabulary 3D scene graphs, OVSG [20] presents an offline method to build nodes and edges based on OVIR-3D [11]. In this framework, three types of nodes are feature-encoded to support free-form text-based queries. Clio [21] utilizes the information bottleneck principle to evaluate task relevance and proposes an online framework to construct task-driven scene graphs with embedded open-set semantics. ConceptGraphs [9] are created by fusing the 2D outputs of foundation models into 3D space and various language-guided planning tasks are presented to demonstrate their utility. In this work, we adopt an approach similar to [21] and [9] for building scene graphs, with an added focus on further reducing the size of the scene graphs. **The aforementioned approaches are single-robot frameworks** and methods for multi-robot 3DSGs construction are introduced in Section II-B.

B. Multi-robot Mapping System

In communication-limited environments, it is crucial to minimize the data transmission between multiple robots while ensuring cooperative mapping and relative pose estimation. To achieve this, SMMR-Explore [22] only transmits submaps in the form of 2D occupancy grids and reconstructs place descriptors and point clouds locally for place recognition and map registration. Building on this, MR-TopoMap [23] and MR-GMMExplore [24] further reduce data volume by transmitting topological maps and Gaussian Mixture Model (GMM) maps. However, these approaches [22] [23] [24] only focus on constructing geometric maps of unknown environments and utilizing geometric features for map merging.

Existing multi-robot semantic mapping systems are closed vocabulary. Kimera-Multi [25] implements a distributed semantic-metric SLAM system, enabling robots to construct 3D mesh models of the environment in real-time collaboratively. In the work by Yue et al. [26], semantic-labeled point clouds are exchanged between robots during outdoor exploration, and an expectation-maximization method is introduced to merge local maps. Hydra-multi [13] extends Hydra [12] into a centralized multi-robot system. In this framework, each robot continuously publishes its entire local scene graph, including a 3D mesh layer, while a control station handles relative transform estimation. D-Lite [17] is the only work addressing communication constraints in multi-robot coordination. It employs graph theory to compress 3D scene graphs by greedily preserving the shortest paths between locations of interest.

C. Open-vocabulary Scene Understanding

With the emergence of foundation models, many works have begun to study language-guided object retrieval, which aims to locate objects based on text queries using open-vocabulary scene representations. Given that explicit representations can update maps incrementally, they are more suitable for multi-robot systems and therefore we focus on explicit representations. Explicit open-vocabulary representations are typically achieved by projecting 2D semantic features onto

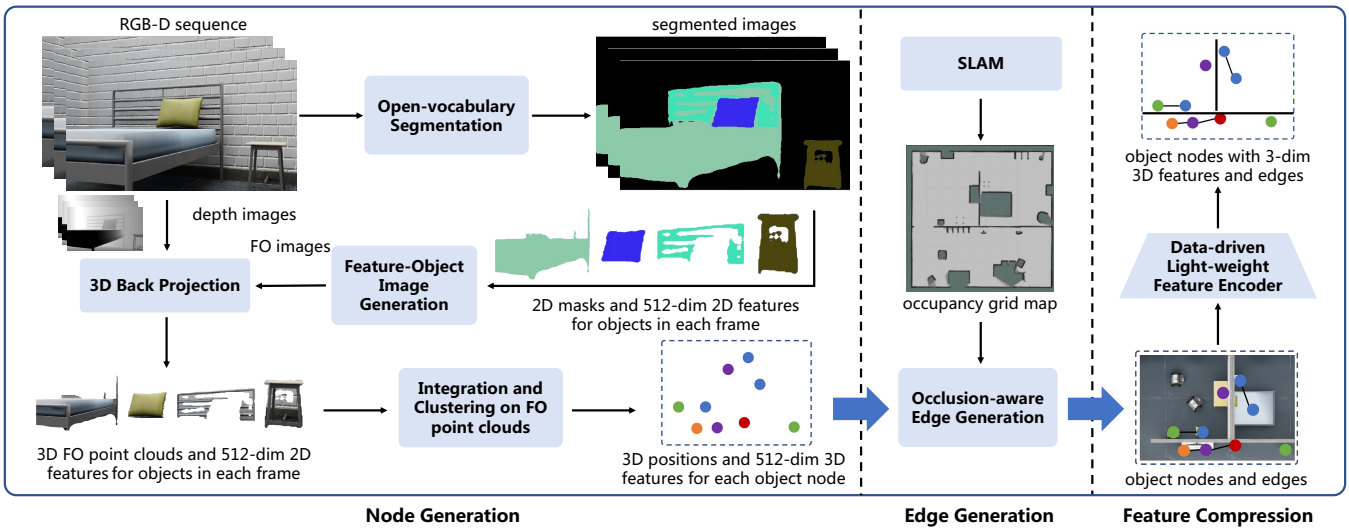


Fig. 2. The Generation Process of COGraphs.

3D points and then encoding features into points [7] [8], instances [11] [27], and Gaussians [28] [29].

In OVIR-3D [11] and OpenMask3D [27], the masks of instances are generated using foundation models and prior point clouds, and then the 3D features of each instance are computed offline. ConceptFusion [7] integrates pixel-aligned open-vocabulary features into 3D point cloud maps by combining traditional SLAM with multi-view images. Similarly, OpenScene [8] relies on prior point clouds and stores semantic features for each pixel, **resulting in a large data volume.**

3D Gaussian Splatting provides another way to realize open-vocabulary scene understanding using 3D Gaussian point cloud techniques. Recent studies [28] [29] integrate pre-trained 2D semantic features into 3D Gaussians to train a semantic 3DGS model, enabling object retrieval on rendered images. Notably, LangSplat [28] introduces an MLP encoder to reduce the dimensionality of CLIP features, thereby accelerating the training process. Our work proposes an encoder-decoder strategy to compress semantic features for efficient data transmission.

III. Method

A. COGraphs Representation

In our distributed multi-robot system, the robots construct and share local COGraphs of the explored area, aiming to perform semantic mapping cooperatively. Each robot utilizes an RGB-D camera to capture semantic and depth information and then the semantic features are projected into 3D space. To ensure localization accuracy and robustness, in this work, we employ a LiDAR-based SLAM algorithm (Cartographer [30]) to estimate robot poses and generate occupancy grid maps, which assist the construction of the COGraphs.

The proposed COGraph consists of the robot name, nodes, and edges. Each node contains the information presented in Tab. I while each edge only contains the information to identify the adjacency between nodes, including the robot name and IDs of the two nodes. To reduce data transmission, the 512-dimensional features of each node are stored locally while only the 3-dimensional features along with other information in the COGraphs are transmitted. Upon receiving COGraphs from other robots, the 512-dimensional features are reconstructed from the 3-dimensional features using a feature decoder. Additionally, only newly generated nodes and edges are shared during communication.

TABLE I
The Information One Node Contains in COGraphs

Symbol	Description	Size	Transmit
N	robot name	8 bits	yes
i	node ID	8 bits	yes
pos_i	3D center position	96 bits	yes
l_i	feature label	32 bits	yes
b_i	bounding box	24 bits	yes
$f^{3D}_{i,512}$	512-dimensional feature	4096 bits	no
$f^{3D}_{i,3}$	3-dimensional feature	24 bits	yes

B. Feature-object (FO) Nodes and Edges Generation

As illustrated in Fig. 2, given a sequence of RGB-D images, we run an open-vocabulary segmentation model to obtain the 2D mask m_k and 2D feature $f^{2D}_{k,512}$ for each object k in each frame. Since instance-aware segmentation provides a label list to project features into label IDs, we utilize it for subsequent clustering to generate nodes. Consequently, each object is associated with a unique 512-dimensional feature while multiple objects may share the same feature label. To facilitate accurate feature matching during node generation, we design a feature-object encoding strategy to generate FO images to differentiate between objects with identical labels. Each pixel of the FO images has 24 bits, in which 16 bits represent the feature label l_k and the other 8 bits represent the object index k. These FO images can be transmitted in the same way as existing image formats, and various lossless image compression methods can also be employed.

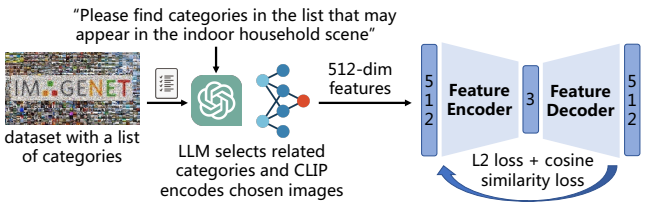


Fig. 3. Training Process of the Feature Encoder and Decoder.

3D back projection is conducted using FO images, depth images, and poses derived from SLAM. This module generates the 3D FO point clouds pc_k and the corresponding 2D feature $f^{2D}_{k,512}$ for each object k in every frame. We integrate semantic point clouds from adjacent frames according to the first 16 bits of information of each point in pc_k. Then clustering is performed using an approach similar to Hydra [12]. The output is a set of nodes, each characterized by a node ID i, a center position pos_i, and a bounding box b_i.

A separate thread is performed to compute the 3D semantic features for each object. We determine whether the semantic point clouds pc_k in a sequence of frames belong to the same object by analyzing the information encoded in FO images. If they do, and the overlap between these point clouds exceeds a predefined threshold, the 2D features $f^{2D}_{k,512}$ associated with these point clouds are averaged to produce 3D features $f^{3D}_{k,512}$. The corresponding point clouds are then merged. Given each 512-dimensional 3D feature $f^{3D}_{k,512}$ and its corresponding point clouds, we assign it to the nearest node by selecting the one closest to the center of the merged semantic point cloud, resulting in the node feature $f^{3D}_{i,512}$.

Edges are generated based on two criteria: 1) the distance between nodes is below a predefined threshold, and 2) there is no occlusion between them. While the constructed nodes include center positions and bounding boxes to describe spatial relationships, background elements such as walls are often undetected by the segmentation model due to their large size and minimal texture. To address this, we utilize 2D occupancy grid maps generated by 2D-LiDAR SLAM, which provide background information. Candidate edges are initially created based on the relative positions of nodes. These edges are then refined using the occupancy grid map to eliminate connections between nodes that belong to different rooms.

C. Data-driven Feature Compression

Compared to existing open-vocabulary 3D map representations, transmitting semantic nodes and edges generated in Section III-B significantly reduces the data volume. However, the 512-dimensional features of each node still pose a communication overhead in environments with limited bandwidth. To address this, we propose a data-driven, lightweight feature compression strategy to further reduce the feature dimension.

The feature encoder is implemented as a multi-layer perceptron (MLP), where fully connected layers sequentially reduce the dimension from 512 to 3. Specifically, the dimensions of the linear layers are as follows: 512, 256, 256, 128, 64, 32, 16, and finally 3. Each linear layer, except for the first one, is followed by BatchNorm1d and ReLU layers, which normalize and activate the features respectively.

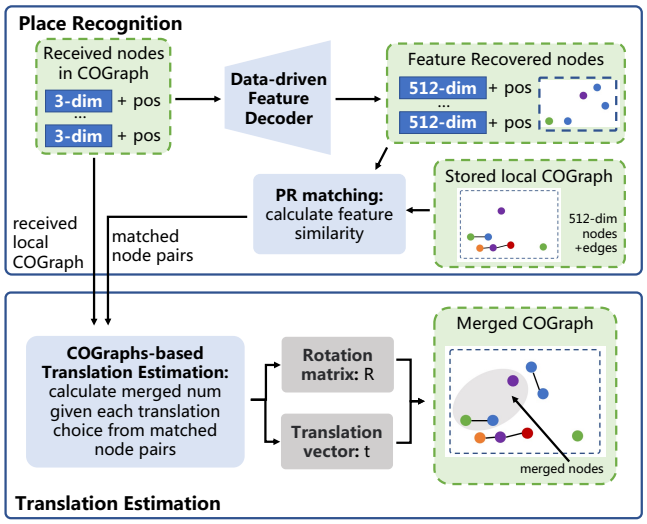


Fig. 4. COGraphs Merging.

In contrast, the feature decoder is structured as an inverse MLP, expanding the dimension from 3 back to 512 through fully connected layers. The layer dimensions are arranged in increasing order: 3, 8, 16, 32, 64, 128, 256, 256, and finally 512. Similar to the encoder, each linear layer, except for the first one, is followed by a ReLU activation, ensuring the non-linear transformation of the encoded features.

As shown in Fig. 3, we train the feature encoder and decoder using selected images from the ImageNet dataset [31]. The dataset provides a list of 1000 categories, and we extract category labels relevant to our environment via a Large Language Model (LLM) [32]. Corresponding images and their associated bounding boxes are chosen based on these labels. The 512-dimensional image features, extracted using the CLIP image encoder, are fed into the encoder and decoder for training. The network is optimized to effectively compress and reconstruct high-dimensional features. The encoder compresses the original 512-dimensional features, while the decoder reconstructs them, yielding a new 512-dimensional feature representation. The loss function combines L2 loss and cosine similarity loss between the original and the reconstructed 512-dimensional features, guiding the training process.

D. COGraphs Merging

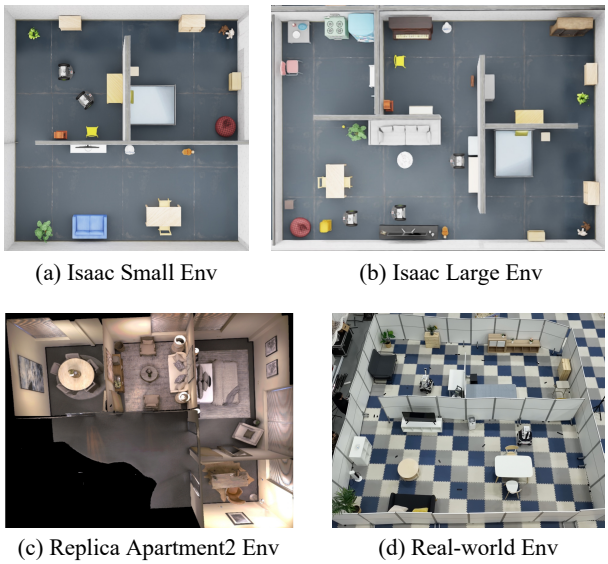
When a robot receives COGraphs from other robots, it first determines whether they have passed the same area and then estimates their relative positions to merge the COGraphs. An illustration of the merging process is shown in Fig. 4.

1) Place Recognition: Since the original 512-dimensional feature vectors are compressed to 3 dimensions before transmission, the feature decoder trained in Section III-C is used to recover each node's semantic feature $f^{3D}_{i',512}$ in received COGraphs. Place recognition is then performed by iteratively calculating the feature similarity between each

Algorithm 1: COGraphs Merging

Input: G: local COGraph , G': received COGraph, R: rotation matrix
Output: G^m: merged COGraph

pairs = []
G^m = G
for *node index* i *in* G **do**
  for *node index* i' *in* G' **do**
    sim = cos_similarity($f^{3D}_{i,512}$, $f^{3D}_{i',512}$)
    if *sim* $>$ *thred_sim* **then**
      pairs.append([i, i'])
if *num(pairs)* $>$ *thred_num* **then**
  for *[i, i'] in pairs* **do**
    $t_{i,i'}$ = calculate_translation(pos_i, $pos_{i'}$)
    $m_{i,i'}$ = merged_node_num($t_{i,i'}$, G, G')
  t = argmax($m_{i,i'}$)
  G^m = merge(G,G',t,R)
return G^m



(a) Isaac Small Env (b) Isaac Large Env

(c) Replica Apartment2 Env (d) Real-world Env

Fig. 5. Experiment Environments.

received node and nodes in the local COGraph. The cosine similarity is computed as:

$$Similarity = \frac{f^{3D}_{i',512} \cdot f^{3D}_{i,512}}{|f^{3D}_{i',512}| \cdot |f^{3D}_{i,512}|} \tag{1}$$

where i' represents node IDs in the received COGraph, and i represents node IDs in the local COGraph. When the similarity exceeds a predefined threshold, node i' and node i are marked as a matching pair. If the number of matching pairs between the received and the local COGraph exceeds a threshold, it strongly indicates that the two robots have passed through the same area, prompting the translation estimation step.

2) Translation Estimation: Each robot takes its starting position as the origin, and its orientation as the X-axis to establish a local coordinate system. When merging maps from multiple robots, the rotation R and translation t between coordinate systems need to be estimated. As the rotation of the robots can be directly obtained using a compass following the method in [23], only the translation vector t is estimated. Two nodes generated from different robots can be merged into one if their feature similarity and distance after the coordinate transformation both fall below their respective thresholds. We go through all the candidate translations $t_{i,i'}$ corresponding to the matching pairs and choose the candidate translation t that leads to the largest number of merged nodes. Using the chosen translation vector t and the rotation matrix R, the merged COGraph is generated. The merging process is detailed in Algorithm 1.

IV. Experiments

A. Experiment Setup

1) Replica Dataset: The Replica dataset [33] has been widely used in 3D scene construction and object retrieval studies. It provides 18 indoor environments and we select the Apartment2 environment shown in Fig. 5(c) as it features multiple rooms and is sufficiently large to support multi-robot mapping. To facilitate mapping and query evaluation, we develop a ROS wrapper [1] to extract RGB-D sequences and ground-truth poses from the dataset, transforming them into ROS bag files for seamless integration with our framework.

2) Simulation environment: We construct simulation experiments on the NVIDIA Isaac Sim platform [34], which offers high-fidelity environment rendering, physical modeling, and the ROS bridge interface. As shown in Fig. 5(a) and Fig. 5(b), we create a small and a large IKEA environment, including living rooms, bedrooms, kitchens, and so on. We also utilize the NVIDIA Carter robot [35], which is equipped with an RGB-D camera, a 2D-LiDAR, and an IMU.

3) Metrics: We use the object finding rate R_{obj} to evaluate the accuracy of the 3D Scene Graphs [12]. The query success rate R@n measures the object retrieval performance by considering the top-n most likely objects in the COGraph [9], with the retrieval counted successful if the correct object is among them. The root mean square error of the translation vector P_{trans} is used to evaluate the accuracy of map merging [22]. Importantly, the amount of data transmitted between robots is analyzed to assess communication efficiency.

4) Baselines: According to ablation studies in StructNav [15], the semantic segmentation module is the bottleneck in scene graph performance. Therefore, we compare different segmentation models (SAM [2], Detic [36], MaskRCNN [6]) in combination with the vision-language model (CLIP [3]) to construct open-vocabulary 3D scene graphs. Since existing multi-robot semantic mapping systems are closed vocabulary and mainly rely on point clouds, we evaluate our feature compression strategy by comparing 3-dimensional COGraphs with the direct transmission of 512-dimensional COGraphs and semantic point clouds.

[1] https://github.com/efc-robot/replica-ros-wrapper

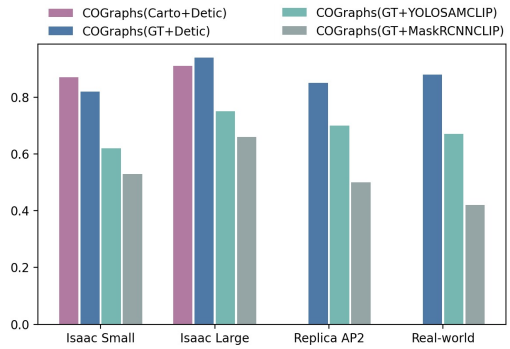
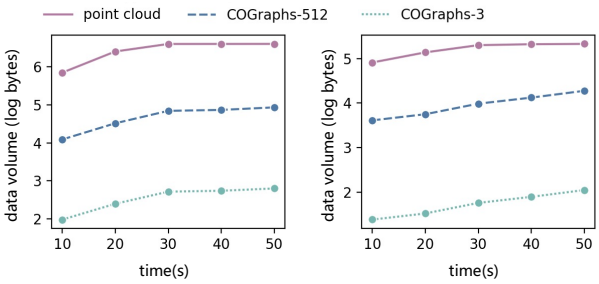


Fig. 6. Object Finding Rate.



(a) Isaac Small Env (b) Real-world Env

Fig. 7. Time-Amount of Data.

5) Real-world Dataset: As shown in Fig. 5(d), our real-world environment is 9m × 9m in size with 3 rooms. Two robots equipped with iPads are deployed to collect data. We control them remotely to collect RGB-D images and pose information using an APP called Record3D [37]. A custom script converts the recorded data into ROS bag files. This accurate and efficient data collection method can be adapted to various scenarios with minimal effort.

B. Object Finding Rate in COGraphs

We evaluate the performance of our scene graph construction on a Desktop PC (CPU: Intel I7-13700, GPU: Nvidia RTX 4080). For the two Isaac IKEA environments, Cartographer [30] is used for localization (shown as purple bars), while in the Replica environment, ground-truth (GT) poses are used. In the real-world dataset, poses are derived using the built-in localization algorithm in Record3D and they are labeled as "GT" for simplicity in Fig. 6. Each experiment is repeated three times, and the results are averaged.

As shown in Fig. 6, among the different segmentation methods, Detic achieves the highest object finding rate, with an average R_{obj} of 87.54%. It shows that Detic has excellent segmentation capability and real-time performance. In contrast, MaskRCNN, a closed-vocabulary model, can recognize fewer objects, leading to lower mapping accuracy. While the combination of YOLO [38], SAM, and the CLIP visual encoder offers superior segmentation performance, its high computational demands limit the segmentation frame rate to 4 frames per second (fps). This slower rate prevents it from keeping pace with the 10fps mapping rate, resulting in a lower mapping accuracy. Unlike MaskRCNN and SAM, Detic is able to output both masks and CLIP features for each detected object with an average frame rate of 8fps. Therefore, Detic is selected as the open-vocabulary segmentation model in our framework.

C. Data Transmission Evaluation

As shown in Fig. 7, we run multi-robot mapping experiments to analyze the data transmission between robots and take the logarithm base 10 of the amount of data. The point cloud map with semantics (point cloud), the COGraph with 512-dimensional semantic features (COGraphs-512), and the COGraph with 3-dimensional semantic features (COGraphs-3) are compared. In the Issac small environment, their average data volumes over time are 3058166, 274877, 411 bytes respectively, while they are 169691, 10342, 61 in the real-world environment. Our proposed method can significantly lower the data volume because we cluster semantic point clouds of the same object into multiple nodes and use 3-dimensional vectors to encode their semantic information. As a result, only a few bytes are required. This method can reduce the data volume by 99.98% compared with transmitting semantic point clouds, and 99.25% compared with 512-dimensional COGraphs.

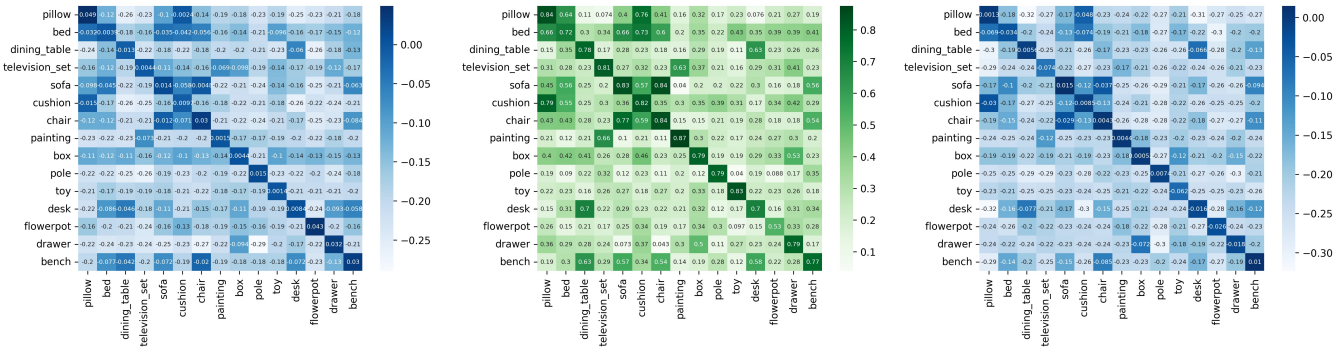
TABLE II
OBJECT RETRIEVAL BASED ON TEXT QUERIES

Environment	Query Type	R@1	R@2	R@3	# Queries
Isaac Small	Appeared	0.9	1.0	1.0	10
	Similar	0.4	0.5	0.5	10
	Descriptive	0.3	0.3	0.3	10
Isaac Large	Appeared	0.8	1.0	1.0	10
	Similar	0.6	0.7	0.7	10
	Descriptive	0.3	0.3	0.3	10
Replica AP2	Appeared	0.9	1.0	1.0	10
	Similar	0.7	0.7	0.7	10
	Descriptive	0.3	0.3	0.3	10
Real-world	Appeared	0.8	0.9	0.9	10
	Similar	0.4	0.6	0.7	10
	Descriptive	0.3	0.4	0.4	10

D. Object Retrieval Evaluation

The CLIP model is a multimodal neural network designed to align image and text representations. It can independently extract image features from visual inputs and generate text features from textual descriptions. Object retrieval is then conducted by calculating the cosine similarity between the image features and the text feature corresponding to the query.

We categorize the queried texts into three types and generate 10 queries for each type [9]. The object retrieval results are presented in Tab. II. The "appeared" queries are selected from feature labels, and our constructed COGraph achieves average retrieval success rates of 85%, 97.5%, and 97.5% for the R@1, R@2, and R@3 metrics across four environments. "Similar" queries are synonyms of feature labels, and approximately 65% of them in the R@3 metric can be successfully retrieved. For "descriptive" object phrases, only around 30% of the queries yield successful retrievals. These results indicate that CLIP-based queries perform well when the queried text has appeared in the feature label



(a) Text features & Image features (before) (b) Image features (before) & Image features (after) (c) Text features & Image features (after)

Fig. 8. Cosine Similarity between Text Features and Image Features (before feature encoding and after feature decoding).

TABLE III
MAP MERGING EVALUATION

Environment	Dim	Pose	$\mathbf{P_{trans}}$(m)	$\mathbf{R_{obj}}$
Isaac Small	512	GT	0.923	all
		Carto	1.063	all
	3	GT	0.084	all
		Carto	1.333	all
Isaac Large	512	GT	0.286	0.95
		Carto	0.559	0.875
	3	GT	0.138	0.95
		Carto	0.559	0.875
Replica AP2	512	GT	0.084	0.85
	3	GT	0.084	0.85
Real-world	512	GT	0.213	0.833
	3	GT	0.213	0.833

list. With the advance of visual-language models, the object retrieval performance of our framework can be improved, especially when the object query is a vague description statement.

Then, we analyze whether utilizing our feature encoding and decoding strategy will affect the query performance. In this work, we give the LLM [32] the category list of the ImageNet dataset, and input "Please find the categories that may appear in the indoor household scene" to get the training dataset of the encoder and decoder. In Fig. 8, we evaluate the object retrieval performance by performing auto-correlation and cross-correlation matching on features from 15 distinct objects. Fig. 8(a) shows the matching results between the original CLIP image features and text features. The deep color along the diagonal and the lighter colors in other areas indicate that the similarity between text and image features is low for different objects, while the similarity is high when comparing the same object's text and image features. Fig. 8(c) illustrates the matching results between the image features and text features after applying our feature compression strategy. The image features have been resized to 3 dimensions and then recovered to 512 dimensions. Similar to the original features, deep colors appear along the diagonal, and lighter colors are seen in other regions, showing that, after the data-driven compression process, the image features can still be effectively matched with the text features.

E. Map Merging Evaluation

The map merging results are presented in Tab. III. Compared to a merging approach without feature compression, the increase in translation estimation error is minimal. By appropriately setting the merging thresholds, the accuracy of the scene graph remains unaffected. In addition, we can see that localization errors emerge as the primary factor contributing to translation vector estimation inaccuracies. Interestingly, the object finding rate in the Isaac small environment is 100% when two robots perceive cooperatively, which is higher than Fig. 6. This is probably because objects missed by one robot are captured by another. Visualizations of the multi-robot merging process across the four environments are provided in the attached video.

Fig. 8(b) illustrates the matching results of the image features before and after applying the encoder and decoder. The deep color along the diagonal indicates that features processed through the encoder-decoder pipeline retain a high similarity to the original features while maintaining distinctiveness from other features. In summary, reducing feature dimensions during transmission has little impact on the precision of multi-robot scene graphs merging. This demonstrates that our method effectively reduces communication data volume without compromising mapping performance.

V. CONCLUSION AND FUTURE WORK

This paper presents a communication-efficient multi-robot mapping framework that allows for open-vocabulary object retrieval. We propose a method for constructing COGraphs, introducing a data-driven feature encoder to compress feature dimensions. To facilitate multi-robot collaboration, we develop place recognition and translation estimation strategies based on semantic features for efficient COGraph merging. Our framework is validated on both simulated environments and real-world datasets, demonstrating two orders of magnitude of reduction in data transmission while maintaining state-of-the-art 3DGS mapping accuracy and query success rates. In future work, we will leverage advanced vision-language models to further enhance open-vocabulary query capabilities and develop exploration strategies for autonomous semantic mapping in communication-constrained environments.

REFERENCES

[1] M. Corah, C. O'Meadhra, K. Goel, and N. Michael, "Communication-efficient planning and mapping for multi-robot exploration in large environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1715–1721, 2019.

[2] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[4] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto, "Volumetric instance-aware semantic mapping and 3d object discovery," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 3037–3044, 2019.

[5] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1689–1696.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[7] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, A. Maalouf, S. Li, G. Iyer, S. Saryazdi, N. Keetha *et al.*, "Conceptfusion: Open-set multimodal 3d mapping," *arXiv preprint arXiv:2302.07241*, 2023.

[8] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser *et al.*, "Openscene: 3d scene understanding with open vocabularies," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 815–824.

[9] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5021–5028.

[10] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola, "Distilled feature fields enable few-shot language-guided manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 405–424.

[11] S. Lu, H. Chang, E. P. Jing, A. Boularias, and K. Bekris, "Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data," in *Conference on Robot Learning*. PMLR, 2023, pp. 1610–1620.

[12] 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. [Online]. Available: https://api.semanticscholar.org/CorpusID:248913107

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

[14] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese, "3d scene graph: A structure for unified semantics, 3d space, and camera," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5664–5673.

[15] J. Chen, G. Li, S. Kumar, B. Ghanem, and F. Yu, "How to not train your dragon: Training-free embodied object goal navigation with semantic frontiers," *Proceedings of Robotics: Science and System XIX*, p. 075, 2023.

[16] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning," in *7th Annual Conference on Robot Learning*, 2023.

[17] Y. Chang, L. Ballotta, and L. Carlone, "D-lite: navigation-oriented compression of 3d scene graphs for multi-robot collaboration," *IEEE Robotics and Automation Letters*, 2023.

[18] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans," *Robotics: Science and Systems XVI*, 2020.

[19] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[20] H. Chang, K. Boyalakuntla, S. Lu, S. Cai, E. P. Jing, S. Keskar, S. Geng, A. Abbas, L. Zhou, K. Bekris *et al.*, "Context-aware entity grounding with open-vocabulary 3d scene graphs," in *Conference on Robot Learning*. PMLR, 2023, pp. 1950–1974.

[21] D. Maggio, Y. Chang, N. Hughes, M. Trang, D. Griffith, C. Dougherty, E. Cristofalo, L. Schmid, and L. Carlone, "Clio: Real-time task-driven open-set 3d scene graphs," *arXiv preprint arXiv:2404.13696*, 2024.

[22] J. Yu, J. Tong, Y. Xu, Z. Xu, H. Dong, T. Yang, and Y. Wang, "Smmr-explore: Submap-based multi-robot exploration system with multi-robot multi-target potential field exploration method," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8779–8785.

[23] Z. Zhang, J. Yu, J. Tang, Y. Xu, and Y. Wang, "Mr-topomap: Multi-robot exploration based on topological map in communication restricted environment," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 794–10 801, 2022.

[24] Y. Wu, Q. Gu, J. Yu, G. Ge, J. Wang, Q. Liao, C. Zhang, and Y. Wang, "Mr-gmmexplore: Multi-robot exploration system in unknown environments based on gaussian mixture model," in *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2022, pp. 1198–1203.

[25] Y. Chang, Y. Tian, J. P. How, and L. Carlone, "Kimera-multi: a system for distributed multi-robot metric-semantic simultaneous localization and mapping," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 210–11 218.

[26] Y. Yue, C. Zhao, Z. Wu, C. Yang, Y. Wang, and D. Wang, "Collaborative semantic understanding and mapping framework for autonomous systems," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 978–989, 2020.

[27] A. Takmaz, E. Fedele, R. W. Sumner, M. Pollefeys, F. Tombari, and F. Engelmann, "Openmask3d: open-vocabulary 3d instance segmentation," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023, pp. 68 367–68 390.

[28] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, "Langsplat: 3d language gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 051–20 060.

[29] S. Zhou, H. Chang, S. Jiang, Z. Fan, Z. Zhu, D. Xu, P. Chari, S. You, Z. Wang, and A. Kadambi, "Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 676–21 685.

[30] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[31] A. Howard, E. Park, and W. Kan, "Imagenet object localization challenge," https://kaggle.com/competitions/imagenet-object-localization-challenge, 2018, kaggle.

[32] "Kimi," 2024. [Online]. Available: https://kimi.moonshot.cn/

[33] "Replica Dataset," 2019. [Online]. Available: https://github.com/facebookresearch/Replica-Dataset

[34] "Nvidia isaac sim," 2024. [Online]. Available: https://developer.nvidia.com/isaac/sim

[35] "Nvidia carter robot," 2024. [Online]. Available: https://docs.omniverse.nvidia.com/isaacsim/latest/features/environment_setup/assets/usd_assets_robots.html

[36] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," in *European Conference on Computer Vision*. Springer, 2022, pp. 350–368.

[37] "Record3D," 2024. [Online]. Available: https://record3d.app/

[38] "Yolov8," 2024. [Online]. Available: https://docs.ultralytics.com/zh/models/yolov8/