

# Multi-Robot Collaborative Perception With Graph Neural Networks

Yang Zhou<sup>ID</sup>, Graduate Student Member, IEEE, Jiuhong Xiao<sup>ID</sup>, Yue Zhou<sup>ID</sup>,  
and Giuseppe Loianno<sup>ID</sup>, Member, IEEE

**Abstract**—Multi-robot systems such as swarms of aerial robots are naturally suited to offer additional flexibility, resilience, and robustness in several tasks compared to a single robot by enabling cooperation among the agents. To enhance the autonomous robot decision-making process and situational awareness, multi-robot systems have to coordinate their perception capabilities to collect, share, and fuse environment information among the agents efficiently to obtain context-appropriate information or gain resilience to sensor noise or failures. In this letter, we propose a general-purpose Graph Neural Network (GNN) with the main goal to increase, in multi-robot perception tasks, single robots' inference perception accuracy as well as resilience to sensor failures and disturbances. We show that the proposed framework can address multi-view visual perception problems such as monocular depth estimation and semantic segmentation. Several experiments both using photo-realistic and real data gathered from multiple aerial robots' viewpoints show the effectiveness of the proposed approach in challenging inference conditions including images corrupted by heavy noise and camera occlusions or failures.

**Index Terms**—Aerial systems applications, deep learning for visual perception.

## I. INTRODUCTION

IN the past decade, robot perception played a fundamental role in enhancing robot autonomous decision-making processes in a variety of tasks including search and rescue, transportation, inspection, and situational awareness. Compared to a single robot, the use of multiple robots can speed up the execution of time-critical tasks while concurrently providing resilience to sensor noises and failures.

Multi-robot perception includes a wide range of collaborative perception tasks with multiple robots (e.g., swarms of aerial or ground robots) such as multi-robot detection, multi-robot tracking, multi-robot localization and mapping. They aim to achieve accurate, consistent, and robust environment sensing, which is crucial for collaborative high-level decision-making policies and

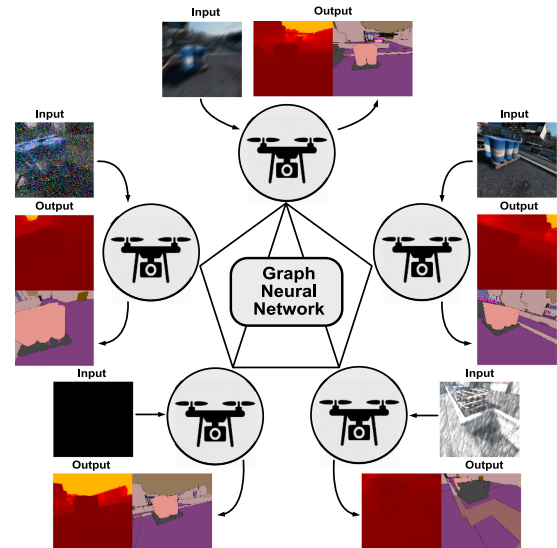


Fig. 1. Multi-robot collaborative perception with GNN. In this case, images are the inputs and monocular depth estimation and semantic segmentation are the downstream tasks.

downstream tasks such as planning and control. There are still several major challenges in multi-robot perception. Robots must efficiently collaborate with each others with minimal or even absence of communication. This translates at the perception level in the ability to collect and fuse information from the environments and neighbour agents in an efficient and meaningful way to be resilient to sensor noise or failures. Information sharing is fundamental to accurately obtain context-appropriate information and provides resilience to sensor noise, occlusions, and failures. These problems are often resolved by leveraging expert domain knowledge to design specific and handcrafted mechanisms for different multi-robot perception problems.

In this work, we address the multi-robot perception problem with GNNs. Recent trend of deep learning has produced a paradigm shift in multiple fields including robotics. Data-driven approaches [1] have outperformed classical methods in multiple robot perception problems, including monocular depth estimation, semantic segmentation, object detection, and object tracking without requiring expert domain knowledge. The single-robot system can benefit from the thrive of deep neural networks and collaboration with other agents. Graph Neural Networks (GNNs) can exploit the underlying graph structure of the multi-robot perception problem by utilizing the message passing among nodes in the graph. The node features are updated

Manuscript received September 9, 2021; accepted December 20, 2021. Date of publication January 10, 2022; date of current version January 25, 2022. This letter was recommended for publication by Associate Editor C. Wang and Editor C. Cadena Lerna upon evaluation of the reviewers' comments. This work was supported in part by Qualcomm Research, the NSF CPS under Grant CNS-2121391, and in part by the Technology Innovation Institute, Nokia, NYU Wireless. (Corresponding authors: Yang Zhou; Jiuhong Xiao; Yue Zhou; Giuseppe Loianno.)

The authors are with the New York University, Brooklyn, NY 11201 USA (e-mail: yangzhou@nyu.edu; jx1190@nyu.edu; yz1268@nyu.edu; loiannog@nyu.edu).

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2022.3141661>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3141661

by one or multiple rounds by aggregating node features from the neighbors. Various types of graph neural networks have been proposed, including GNN [2], Convolutional GNNs [3], Graph Attention Networks [4]. These methods have proved to be effective for node classification, graph classification, and link prediction. Recently, researchers also started to apply GNNs on multi-robot systems for communication [5] and planning [6]. However, there is still no work exploiting the expressiveness of GNNs for multi-robot perception problems.

This work presents multiple contributions. First, we propose a generalizable GNN-based perception framework for multi-robot systems to increase single robots' inference perception accuracy which is illustrated in Fig. 1. Our approach is flexible and takes into account different sensor modalities. It embeds the spatial relationship between neighbor nodes into the messages and employs the cross attention mechanism to adjust message weights according to the correlation of node features of different robots. Second, we show the proposed approach in two multi-view visual perception tasks: collaborative monocular depth estimation and semantic segmentation and we discuss how to employ the proposed framework with different sensing modalities. Finally, we show the effectiveness of the proposed approach in challenging multi-view perception experiments including camera sensors affected by heavy image noise, occlusions, and failures on photo-realistic as well as real image data collected with aerial robots. To the best of our knowledge, this is the first time that a GNN has been employed to solve a multi-view/multi-robot perception task using real robot image data.

The letter is organized as follows. Section II presents a literature review. Section III presents our GNN architecture whereas Section IV shows two instances of our method to address in two multi-view visual perception problems. In Section V, several experiments are presented to validate the proposed approach. Section VI concludes the letter.

## II. RELATED WORKS

Single-robot scene understanding benefited from the expressive power of deep neural networks. Among these frame-based perception tasks, semantic segmentation [7]–[9] and monocular depth estimation [10], [11] are among important perception problems which have been widely studied in computer vision and robotics community. For both tasks, researchers generally adopt an encoder-decoder structure deep neural network. [7] first employs a fully convolutional network for semantic segmentation. UNet [8] introduces the skip connection, and Deeplab [9] focuses on improving the decoder of the network architecture in order to exploit the feature map extracted from it. In [10], the authors also use deep neural networks to solve monocular depth estimation. Most recently, unsupervised learning methods such as Monodepth2 [11] still follow the encoder-decoder network structure for monocular depth estimation.

Many tasks of interest for the robotics and computer vision communities such as 3D shape recognition [12], [13], object-level scene understanding [14], and object pose estimation [15] can benefit from using multi-view perception. This can improve the task accuracy as well as increase robustness with respect to single robot sensor failures. Multi-robot perception also includes the problems related to communication and bandwidth

limitation among the robot network. Who2com [16] proposes a multi-stage handshake communication mechanism that enables agents to share information with limited communication bandwidth. The degraded agent only connects with the selected agent to receive information. When2com [17] as its following work further investigates the multi-agent communication problem by constructing communication groups and deciding when to communicate based on the communication bandwidth. They validated their approach on semantic segmentation with noisy sensor inputs. Conversely, in our work, we study the multi-robot perception problem employing a general GNN-based framework which directly operates on the graph structure. The proposed framework enables to resolve the multi-robot perception problem by leveraging the representation power benefits of modern GNN methods. We experimentally demonstrate the robustness to heavy exogenous (sensor and environment) noise or sensor corruption.

GNNs can model problems with structures of directional and un-directional graphs. The graph neural network propagates and aggregates messages around neighbor nodes to update each node feature [2]. In addition to nodes, the edges can also embed features [18]. Researchers focus on various approaches to create individual node embeddings. GraphSAGE [19] learns a function to obtain embedding by sampling and aggregation from neighbor nodes. Graph Attention Networks [4] introduces a multi-head attention mechanism to dynamically adjust weights of neighbor nodes in the aggregation step. MAGAT [20] introduces a key-query-like attention mechanism to improve the communication efficiency. GNNs have been applied in different fields of robotics [5], [6], [20]–[22] including multi-robot path planning, decentralized control, and policy approximation. These works show the potential of GNNs in the multi-robot system by modeling each robot as a node in the graph. Leveraging GNN, the method can learn efficient coordination in the multi-robot system and can be generalized to different graph topology with shared model parameters. There are few perception-related works exploiting the power of GNNs. [23] studies the multi-view camera relocalization problem whereas Pose-GNN [24] proposes an image-based localization approach. In these works, node features are initialized from image inputs by convolutional neural network encoders. Both methods treat their problems as supervised graph node regression problems.

## III. METHODOLOGY

### A. Preliminaries

We denote with  $N$  the number of robots in a multi-robot perception system. We consider each robot  $i$  as a node  $v_i$ . If robot  $i$  and robot  $j$  have communication connection, we construct an edge  $e_{ij}$  between  $v_i$  and  $v_j$ . Hence, we construct a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ,  $\mathcal{V} = \{v_i\}$ ,  $\mathcal{E} = \{e_{ij}\}$ ,  $i, j \in \{1, \dots, N\}$ , according to the communication connection between robots. The communication topology can be determined by considering a distance threshold between robots or the strength of the communication signal. The proposed multi-robot perception system, illustrated in Fig. 2, takes observations  $\{\mathbf{x}_i\}_{i=1\dots N}$  of sensors from all robots, and returns the output  $\{\mathbf{y}_i\}_{i=1\dots N}$  after the GNN processing. We denote the neighbor nodes of  $v_i$  as  $\mathcal{N}_{(i)}$ . We assume the graph structure does not change between the time that

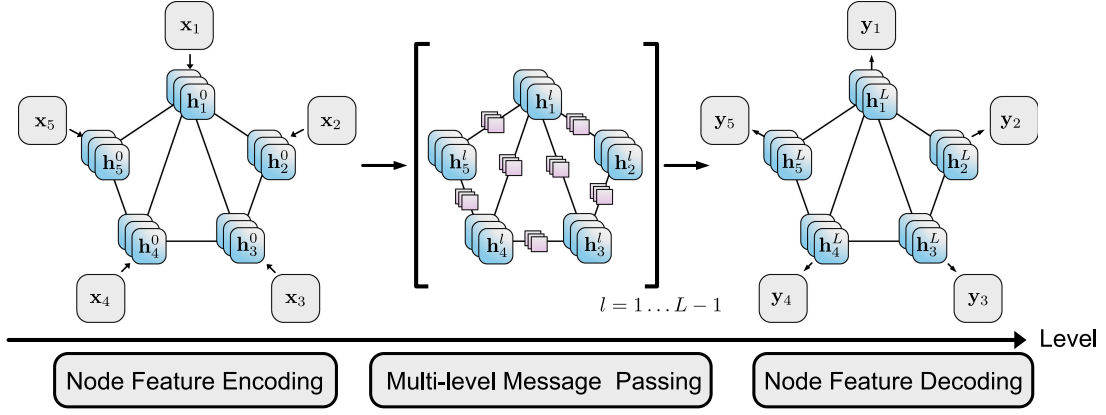


Fig. 2. System overview of the proposed GNN.

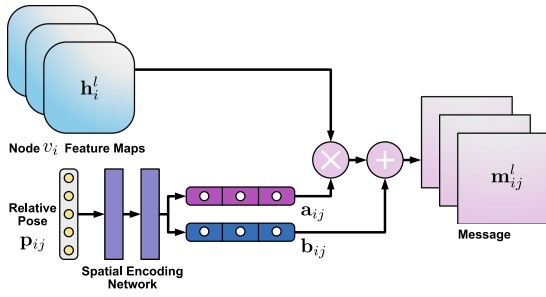


Fig. 3. Message with spatial encoding.

sensors capture information and the time that the GNN provides the results. The GNN first encodes the observation  $\{\mathbf{x}_i\}_{i=1\dots N}$  into the node feature  $\{\mathbf{h}_i\}_{i=1\dots N}$  where  $\mathbf{h}_i^0 = f_{\text{Encode}}(\mathbf{x}_i)$ . At each level  $l \in 1 \dots L$  of the message passing, each node  $v_i$  aggregates messages  $\mathbf{m}_{ij}^l$  from its neighbor nodes  $\{v_j\}_{j \in \mathcal{N}(i)}$ . We consider two different message encoding mechanisms: a) spatial encoding and b) dynamic cross attention encoding. These mechanisms make the messages on bidirectional edges of the graph nonidentical. After each level of the message passing, the node feature  $\mathbf{h}_i^l$  is updated. Once the last level  $L$  of message passing is executed, the final result  $\mathbf{y}_i$  is obtained by decoding the node feature  $\mathbf{h}_i^L$ .

In the following, we detail our proposed method. We explain how to use the determined spatial encoding to utilize the relative spatial relationship between two nodes in Section III-B. We also show the way to encode feature map correlation into messages by cross attention in Section III-C. The message passing mechanism is introduced in Section III-D whereas the design of feature decoder in Section III-E.

### B. Messages With Spatial Encoding

For robotics application, relative spatial relationship of the multi-robot network can be obtained by the multi-robot system itself [25] or from other systems [26]. If we have access to the relative spatial relationship between nodes, we can encode it into the messages shared between the node pairs. We illustrate the message generation mechanism with spatial encoding in Fig. 3. We can represent the relative pose between robot  $i$  and robot  $j$  which contains rotation  $\mathbf{R}_{ij}$  and translation  $\mathbf{t}_{ij} = [t_{ij}^x, t_{ij}^y, t_{ij}^z]$ .

In [27], a continuous rotation representation  $\mathbf{R}'_{ij}$  which takes first two columns of  $\mathbf{R}_{ij}$

$$\mathbf{R}_{ij} = \begin{bmatrix} r_1 & r_4 & r_7 \\ r_2 & r_5 & r_8 \\ r_3 & r_6 & r_9 \end{bmatrix}, \mathbf{R}'_{ij} = \begin{bmatrix} r_1 & r_4 \\ r_2 & r_5 \\ r_3 & r_6 \end{bmatrix}, \quad (1)$$

$$\mathbf{p}_{ij} = [t_{ij}^x, t_{ij}^y, t_{ij}^z, r_1, r_2, r_3, r_4, r_5, r_6]. \quad (2)$$

Compared to common rotation representations including rotation matrix, quaternion and axis angles, this continuous rotation representation is easier to learn for neural networks [27]. We feed the continuous relative pose representation  $\mathbf{p}_{ij}$  into the spatial encoding network to produce  $\{\mathbf{a}_{ij}, \mathbf{b}_{ij}\} = \{(a_{ij})_k, (b_{ij})_k\}_{k=1,\dots,C}$ ,  $(a_{ij})_k \in \mathbb{R}$ ,  $(b_{ij})_k \in \mathbb{R}$  where  $C$  is number of channel of the node feature  $\mathbf{h}_i^l$ . For simplicity, we implement the spatial encoding network as two fully connected layers with ReLU. Inspired by FiLM [28], we embed the relative robot pose  $\mathbf{p}_{ij}$  and the node feature  $\mathbf{h}_i^l$  to compose the message from the node  $v_i$  to the node  $v_j$ . The  $k$ -th channel of the message transforms  $\mathbf{m}_{ij}$  as

$$(\mathbf{a}_{ij}, \mathbf{b}_{ij}) = \text{FiLM}(\mathbf{p}_{ij}), \quad (3)$$

$$(\mathbf{m}_{ij}^l)_k = (a_{ij})_k (\mathbf{h}_i^l)_k + (b_{ij})_k \mathbf{1}, \quad (4)$$

where  $\mathbf{1}$  is a matrix of ones with the same dimension as the node feature.  $\{(a_{ij})_k, (b_{ij})_k\}_{k=1,\dots,C}$  serves as a set of affine transformation parameter applied on the node feature  $\mathbf{m}_{ij}$ . It embeds the relative spatial relationship between robot  $i$  and robot  $j$  onto the node feature  $\mathbf{h}_i^l$  to compose the message  $\mathbf{m}_{ij}^l$  shared from the node  $v_i$  to the node  $v_j$  in the GNN.

### C. Messages With Dynamic Cross Attention Encoding

We also propose another message encoding mechanism with dynamic cross attention between different node features inspired by Graph Attention Networks [4]. This message encoding mechanism takes the dynamic feature relationship of neighbor robots into account and weights the messages from neighbor nodes accordingly. We illustrate this message encoding mechanism in Fig. 4. Consider the node  $v_i$  and the node  $v_j$ , we take cross attention of the node features  $\mathbf{h}_i^l$  and  $\mathbf{h}_j^l$  to obtain a single scaled value as the attention weight of node feature  $\mathbf{h}_i^l$ . We

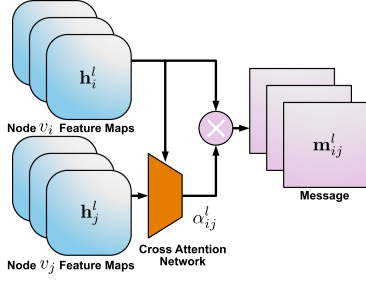


Fig. 4. Message passing with cross attention.

first concatenate both feature maps  $\mathbf{h}_i^l, \mathbf{h}_j^l$ , and transform the concatenated feature map to  $\alpha_{ij}^l \in \mathbb{R}$  by a linear transform module  $F(\cdot)$  and a nonlinear layer  $\text{LeakyReLU}(\cdot)$ . The output of the transform module is a single scaled value. We use  $\text{Softmax}$  to normalize the attention score  $\alpha_{ij}^l$  on the incoming edges  $e_{ij}$  of the node  $v_j$  in order to make attention scores more comparable across different neighbor nodes as

$$\begin{aligned} \mathbf{m}_{ij}^l &= \text{Softmax}_i(\alpha_{ij}^l) \mathbf{h}_i^l, \\ \alpha_{ij}^l &= \text{LeakyReLU}(F[\mathbf{h}_i^l \parallel \mathbf{h}_j^l]). \end{aligned} \quad (5)$$

In order to enrich the model capacity, we also introduce a multi-head attention mechanism. We use  $D$  different transform layer  $F^d$  instead of a single-head attention mechanism to embed the concatenated node feature  $[\mathbf{h}_i^l \parallel \mathbf{h}_j^l]$ . Each attention head generates a message  $(\mathbf{m}_{ij}^l)^d$ . The final message is the average of the  $D$  messages

$$\mathbf{m}_{ij}^l = \frac{1}{D} \sum_{d=1}^D (\mathbf{m}_{ij}^l)^d, \quad (6)$$

$$(\mathbf{m}_{ij}^l)^d = \text{Softmax}_i((\alpha_{ij}^l)^d) \mathbf{h}_i^l, \quad (7)$$

$$(\alpha_{ij}^l)^d = \text{LeakyReLU}(F^d[\mathbf{h}_i^l \parallel \mathbf{h}_j^l]). \quad (8)$$

#### D. Message Passing Mechanism

The message passing mechanism aggregates messages  $\mathbf{m}_{ji}$  among neighbor nodes  $v_j$  and  $v_i$ , and update the node next level feature  $\mathbf{h}_i^{l+1}$ . We simply use an average operation as the aggregation operation instead of other options with extra parameters to keep computation and memory consumption suitable for real-time robotics application

$$\mathbf{h}_i^{l+1} = \frac{1}{\#\mathcal{N}(i)} \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ji}^l. \quad (9)$$

#### E. Feature Decoder

We introduce the feature decoding mechanism in Fig. 5. After the final levels of message passing, the node feature of each node has aggregated the information from other related robots according to spatial relationship by determined spatial message encoding or feature relationship by dynamic cross attention message encoding. We concatenate the original node feature  $\mathbf{h}_i^0$  with the last updated node feature  $\mathbf{h}_i^L$  without losing the original

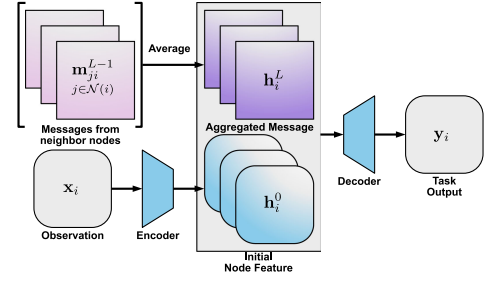


Fig. 5. Node feature maps aggregation and feature decoder.

encoded information as

$$\mathbf{y}_i = f_{\text{Decode}}[\mathbf{h}_i^0 \parallel \mathbf{h}_i^L]. \quad (10)$$

#### F. Task-Related Loss

We train our entire framework with supervised learning. The task-related loss is designed as the mean of objective functions of all robots. The objective function  $F_{\text{loss}}(\cdot)$  can be designed according to the specific sensor modalities and tasks. In Section IV, we show the design for the multi-view visual perception case. The term  $\mathbf{y}_i$  is the prediction obtained from the proposed method,  $\mathbf{y}_i^*$  is the target ground truth.

$$L = \frac{1}{N} \sum_{i=1}^N F_{\text{loss}}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{y}_i^*). \quad (11)$$

### IV. MULTI-VIEW VISUAL PERCEPTION CASE STUDY

Our framework is generalizable to different sensor modalities. In order to show its effectiveness, we demonstrate the proposed approach in two multi-view visual perception problems, semantic segmentation and monocular depth estimation. We leverage Deep Graph Library (DGL) [29] Python package and PyTorch [30] to design our approach.

We take 2D images  $\mathbf{I} \in \mathbb{R}^{H \times W}$  as sensor observations  $\mathbf{x}_i$ . Therefore, we use a differentiable 2D Convolutional Neural Network (CNN) as feature encoder. In order to meet the real-time requirement of the embedded robotic application, we employ MobileNetV2 [31] as a lightweight network for feature encoding. The dimension of the node feature  $\mathbf{h}_i^l$  is  $C \times h \times w$ , where  $(h, w)$  are the scaled dimension of original image size  $(H, W)$ . The message encoding presented in Section III-B and Section III-C uses a 2D convolution operator and matches the dimension of the node feature. The  $k$ -th channel of the message transforms  $\mathbf{m}_{ij}$  as

$$(\mathbf{m}_{ij}^l)_k = (a_{ij}^l)_k (\mathbf{h}_i^l)_k + (b_{ij}^l)_k \mathbf{1}^{h \times w}, \quad (12)$$

where  $\mathbf{1}^{h \times w}$  is a matrix of ones with the same dimension as the node feature  $\mathbf{h}_i^l \in \mathbb{R}^{C \times h \times w}$  when the input modality is a 2D image. The transform module  $F(\cdot)$  described in Section III-C is based on two 2D convolution layers to reduce the dimension of the feature map to 1/8 of the original size, then it flattens it to 1D feature vector. Subsequently, the transform module applies a fully connected layer to transform the 1D feature vector to a single scaled value.

At the feature decoder stage introduced in Section III-E, in order to recover the concatenated feature map to the original size,





Fig. 6. Real data collected by five aerial robots with cameras.

we use 2D transposed convolution with  $\text{ReLU}(\cdot)$  as a nonlinear activation function to enlarge the size of the feature map through 5 layers. We keep the design of the decoder simple compatible with our case studies: semantic segmentation and monocular depth estimation.

As a task-related loss, we employ objective functions which are widely used for semantic segmentation [7] and monocular depth estimation [11]. For monocular depth estimation, we use smooth L1 loss and edge-aware smooth loss

$$F_{\text{depth}}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{y}_i^*) = L_{L1}(\mathbf{y}_i, \mathbf{y}_i^*) + \alpha L_s(\mathbf{y}_i, \mathbf{x}_i),$$

$$L_{L1}(\mathbf{y}_i, \mathbf{y}_i^*) = \begin{cases} 0.5 (\mathbf{y}_i - \mathbf{y}_i^*)^2 / \beta, & \text{if } |\mathbf{y}_i - \mathbf{y}_i^*| < \beta \\ |\mathbf{y}_i - \mathbf{y}_i^*| - 0.5\beta, & \text{otherwise} \end{cases},$$

$$L_s(\mathbf{y}_i, \mathbf{x}_i) = |\partial_x \mathbf{y}_i| e^{-|\partial_x \mathbf{x}_i|} + |\partial_y \mathbf{y}_i| e^{-|\partial_y \mathbf{x}_i|}. \quad (13)$$

where  $\partial_x(\cdot), \partial_y(\cdot)$  represent the gradient operations that apply on 2D depth prediction  $\mathbf{y}_i$  and original 2D image  $\mathbf{x}_i$ . The edge-aware loss encourages the gradient of the depth prediction to be consistent with the original image.

Conversely, for semantic segmentation, we use a cross entropy loss function

$$F_{\text{semantic}}(\mathbf{y}_i, \mathbf{y}_i^*) = L_{CE}(\mathbf{y}_i, \mathbf{y}_i^*). \quad (14)$$

Our method can also be easily employed with different sensor modalities by changing the dimension of node features and the design of encoder and decoder structures. For example, it is possible to use a recurrent neural network (RNN) to encode IMU measurements or a 3D convolutional neural network to encode 3D LiDAR measurements. The user also needs to adapt the task-related loss for other tasks. Takes visual object pose estimation for example, the user should use the distance between points on the ground truth pose and the estimated pose as the task-related loss.

## V. EXPERIMENTS

### A. Dataset

We collected photo-realistic simulation and real-world multi-robot data. We simulate different categories of sensor noises similarly to [32]. We consider four datasets with different properties to validate our approach in different scenarios. All simulated datasets contain depth and semantic segmentation ground truth. The real-world dataset contains depth ground truth. In the four datasets, we simulate the sensor corruption or sensor noise by corrupting images and varying the number of affected cameras as described below.

1) *Photo-Realistic Data*: **Airsim-MAP** dataset was first proposed in When2com [17]. It contains images from five robots

flying in a group in the city environment and rotating along their own z-axis with different angular velocities. In the noisy version of the dataset, **Airsim-MAP** applies Gaussian blurring with random kernel size from 1 to 100 and Gaussian noise [16]. Each camera has over 50% probability of being corrupted and each frame has at least one noisy image in five views [17]. We use this dataset to show that our approach is robust to sensor corruption.

**Industrial-pose**, **Industrial-circle** and **Industrial-rotation** datasets are generated leveraging the Flightmare [33] simulator. We use five flying robots in an industrial harbor with various buildings and containers. In all the datasets, the robots form a circle. **Industrial-pose** dataset has more variations in terms of relative pose combination among the vehicles and the robots fly at a higher altitude compared to the other two datasets. In the **Industrial-circle** dataset, the overlapping region sizes across the robots' Field of View (FoV) are larger compared to the **Industrial-rotation** dataset. Specifically, robots in the **Industrial-circle** are pointing to the center of the team whereas robots in the **Industrial-rotation** are pointing out of the circle. We project the robots' FoV to the 2D ground plane and calculate the ratio between the union and the sum of the projected FoVs, where the ratio of **Industrial-circle** is 24.4% and the ratio of **Industrial-rotation** is 18.4%. Industrial datasets use [32] to generate noise. The noise types include motion blur, Gaussian noise, shot noise, impulse noise, snow effects, and JPEG compression effects. These datasets select the first  $N(0-2)$  cameras to perturb. Compared to **Airsim-MAP**, noise on industrial datasets is less severe, but presents a larger variety. We use **Industrial-pose** dataset to show that our approach is robust to different sensor noises and the **Industrial-circle** and **Industrial-rotation** to verify the robustness of the proposed method to different overlap areas between robots' FoV.

2) *Real Data*: **ARPL-r4** dataset was collected in an indoor flying arena of  $10 \times 6 \times 4m^3$  at the Agile Robotics and Perception Lab (ARPL) lab at New York University using multiple aerial robots based on our previous work [34] equipped with RGB camera that autonomously fly and observe a feature-rich object (see Fig. 6). We use this dataset to validate our method in real-world scenarios. The vehicles are performing extreme maneuvers creating motion blur. The robots form a circle and point their front camera towards the object in the center to maximize their overlapping visual area. We simulate different noise patterns on one or two robots.

### B. Evaluation Protocol

In our experiments, we compare the performance of different methods on the monocular depth estimation and semantic segmentation. We use Abs Relative difference (Abs Rel), Squared Relative difference (Sq Rel) and Root Mean Squared Error (RMSE) as the metric for monocular depth estimation [10]. We use mIoU as the metric of semantic segmentation. The bolded numbers in the table are the best case, and the underlined numbers in the table are the second-best case. We compare different variants of our proposed methods and baseline with different numbers of noisy/corrupted cameras (ranging from 0 to 2). We use a single-robot **baseline** with the same encoder and decoder structure. The baseline is trained on the clean dataset

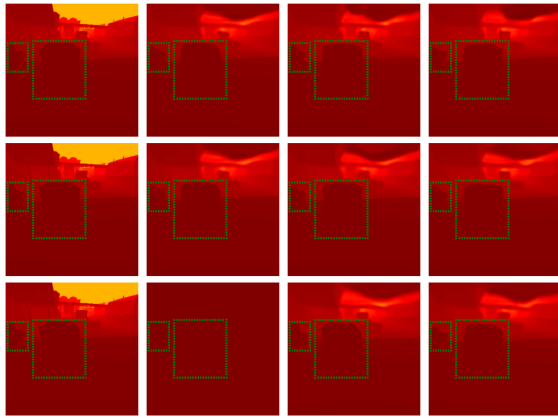


Fig. 7. Monocular depth estimation qualitative results varying the number of noisy cameras from a corrupted sensor. Methods from left to right: ground truth, baseline, mp, mp-pose. From top to bottom: 0, 1 and 2 noisy cameras.

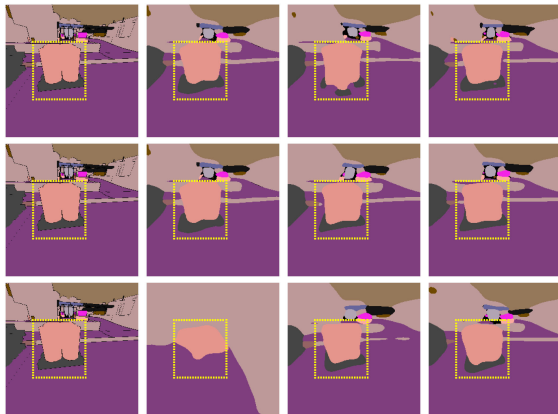


Fig. 8. Semantic segmentation qualitative results varying the number of noisy cameras. Methods from left to right: ground truth, baseline, mp, mp-pose. From top to bottom: 0, 1 and 2 noisy cameras.

where the number of noisy cameras is 0, and we test the baseline with all datasets with different number of noisy cameras. Our methods are trained and tested on the same datasets with all different numbers of noisy cameras. We also use a multi-robot **baseline-mp** which takes all the images as inputs and produces the desired outputs of all inputs. We study three different variants of our method: **mp-pose** represents multi-robot perception with spatial encoding messages, **mp-att** represents multi-robot perception with cross attention encoding messages, and **mp** represents multi-robot perception using messages without encoding, which are the previous-level node features. We use **mp** for ablation study to show the spatial and cross attention encoding effectiveness.

### C. Qualitative Result

In Figs. 7 and 8, we show the result of **baseline**, **mp**, **mp-pose** with different number of noisy cameras. By increasing the number of noisy cameras, the performance of **baseline** decreases, and our methods: **mp-pose**, **mp-att** still preserve high-quality results. We can also see the effects of message encoding by comparing the details of the object boundaries and the perception accuracy of small and thin objects. In Figs. 9 and 10, we use

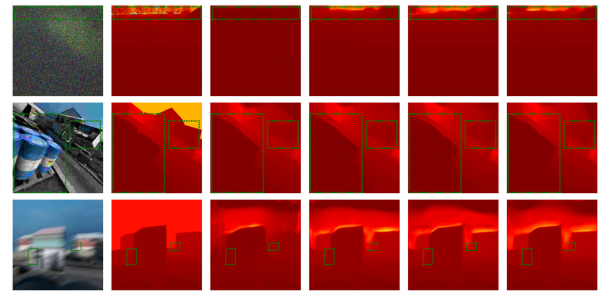


Fig. 9. Monocular depth estimation qualitative results for different types of sensor noise. Methods from left to right: images, ground truth, baseline, mp, mp-pose, mp-att.

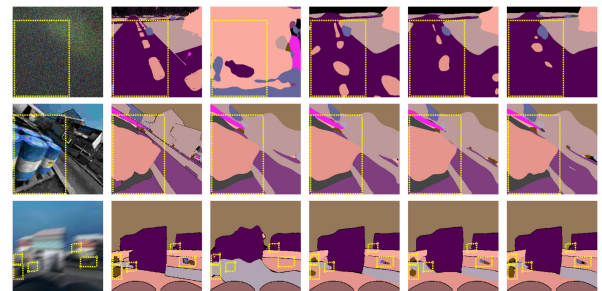


Fig. 10. Qualitative results of semantic segmentation for different types of sensor noises. Methods from left to right: images, ground truth, baseline, mp, mp-pose, mp-att.

images from different datasets with different types of sensor corruption and noise. We highlight the area of object boundaries and small objects in the figures. Single-robot baseline creates wrong object boundaries and misses small and thin objects while our methods recover the information from corrupted images and preserve the perception details on both tasks. These results show that the proposed method outperforms single-robot baseline and is robust to different types of image corruptions and noises. This qualitative result shows that our method is robust to different numbers of noisy cameras in the robot network.

### D. Quantitative Result

1) *Effects of Message Encoding*: To show the effects of the message passing with spatial encoding, we use **Airsim-MAP** dataset with sensor corruption and **Industrial-pose** dataset with sensor noises. In Table I, we observe that **mp-pose** outperforms **baseline** on both tasks among all datasets with different number of noisy cameras. Comparing **mp-pose** and **mp**, we find that the spatial encoding improves the performance of both tasks in most of the experiments. We also demonstrate the effects of the cross attention message encoding on **Airsim-MAP** dataset and **Industrial-pose** dataset. In Table I, we show that **mp-att** outperforms **baseline** on all cases in both tasks, and **mp-att** outperforms **mp** on most cases in both tasks. We also demonstrate the necessity of GNN in the multi-robot perception task by comparing **baseline-mp** with other methods, which shows that **baseline-mp** are outperformed by all other methods, since it cannot learn the correspondence of features by message passing. We can conclude that both message encoding including the spatial encoding and the cross attention encoding improves the

TABLE I  
EFFECTS OF SPATIAL ENCODING AND CROSS ATTENTION. OUR METHOD IS ROBUST TO SENSOR NOISE

# noisy cameras		0				1				2			
dataset	method	Abs Rel↓	Sq Rel↓	RMSE↓	mIoU↑	Abs Rel↓	Sq Rel↓	RMSE↓	mIoU↑	Abs Rel↓	Sq Rel↓	RMSE↓	mIoU↑
Airsim-MAP	baseline	0.0555	2.6438	18.6380	0.6170	0.3359	8.0912	37.1608	0.4154	-	-	-	-
	baseline-mp	0.1556	6.7748	36.2965	0.3514	0.2489	18.1093	44.1963	0.2244	-	-	-	-
	mp-pose	<b>0.0521</b>	<b>2.4841</b>	<b>18.3737</b>	0.6186	<b>0.0799</b>	<b>4.3344</b>	<b>25.4724</b>	0.5055	-	-	-	-
	mp-att	0.0542	2.5365	18.4385	<b>0.6251</b>	0.0900	4.6000	25.9189	<b>0.5081</b>	-	-	-	-
	mp	0.0565	2.7646	18.6658	0.6151	0.0912	4.4830	26.7753	0.5033	-	-	-	-
Industrial-pose	baseline	0.1290	0.5918	3.8945	0.7718	0.2184	3.9593	13.6328	0.6853	0.3085	7.5872	19.8783	0.6000
	baseline-mp	0.3515	3.9684	10.5106	0.5096	0.3793	5.0247	12.7207	0.4903	0.4147	6.5149	15.8023	0.4654
	mp-pose	<b>0.1174</b>	<b>0.5608</b>	<b>3.7336</b>	<b>0.7808</b>	<b>0.1251</b>	0.6564	4.1251	0.7446	<b>0.1277</b>	<b>0.6276</b>	<b>4.1556</b>	<b>0.7436</b>
	mp-att	0.1214	0.5817	3.7995	0.7800	0.1287	<b>0.5881</b>	<b>4.1064</b>	<b>0.7681</b>	0.1385	0.6551	4.2095	0.7403
	mp	0.1262	0.6032	3.8019	0.7791	0.1294	0.6322	4.1794	0.7469	0.1317	0.6360	4.2854	0.7380

TABLE II  
STUDY OF THE EFFECT OF SIZES OF OVERLAPPING AREAS ACROSS ROBOTS' FoV BETWEEN ROBOTS. LARGER OVERLAP AREAS BETWEEN ROBOTS FoV NATURALLY PROVIDE MORE BENEFITS FROM MESSAGE PASSING WITH SPATIAL ENCODING

# noisy cameras		0				1				2			
dataset	method	Abs Rel↓	Sq Rel↓	RMSE↓	mIoU↑	Abs Rel↓	Sq Rel↓	RMSE↓	mIoU↑	Abs Rel↓	Sq Rel↓	RMSE↓	mIoU↑
Industrial-circle	baseline	0.0596	0.0995	0.7935	0.7623	0.1741	0.4763	1.9639	0.6895	0.2941	0.8912	2.7556	0.6144
	baseline-mp	0.4376	1.4737	2.4441	0.5397	0.5530	2.4935	2.7593	0.5132	0.7293	4.1689	3.3767	0.4886
	mp-pose	<b>0.0576</b>	0.0930	<b>0.7880</b>	<b>0.7662</b>	<b>0.0827</b>	<b>0.1359</b>	<b>0.8923</b>	<b>0.7362</b>	<b>0.1269</b>	0.2934	<b>0.9288</b>	<b>0.7340</b>
	mp	0.0582	<b>0.0914</b>	0.7939	0.7597	0.0851	0.1564	0.9111	0.7355	0.1314	<b>0.2714</b>	0.9415	0.7220
Industrial-rotation	baseline	<b>0.1107</b>	0.1340	0.7338	0.7631	0.3182	0.5354	1.5238	0.7107	0.4531	0.8693	2.0525	0.6618
	baseline-mp	0.4761	0.8329	1.8128	0.6524	0.7458	1.3347	2.0131	0.6259	1.1000	2.0564	2.3251	0.5985
	mp-pose	0.1173	<b>0.1259</b>	<b>0.7295</b>	<b>0.7657</b>	0.2228	0.4834	<b>0.9540</b>	<b>0.7387</b>	<b>0.2277</b>	0.2918	<b>1.0274</b>	<b>0.7276</b>
	mp	0.1157	0.1349	0.7432	0.7639	<b>0.2115</b>	<b>0.2893</b>	0.9812	0.7368	0.2307	<b>0.2706</b>	1.0404	0.7212

TABLE III  
REAL-WORLD MONOCULAR DEPTH ESTIMATION EXPERIMENTS. OUR METHODS OUTPERFORM THE BASELINE ON BOTH TASKS

# noisy cameras		0			1			2		
dataset	method	Abs Rel↓	Sq Rel↓	RMSE↓	Abs Rel↓	Sq Rel↓	RMSE↓	Abs Rel↓	Sq Rel↓	RMSE↓
ARPL-r4	baseline	0.1604	0.3416	0.4221	0.1886	0.3812	0.5480	0.2194	0.4206	0.6591
	mp-pose	<b>0.1566</b>	<b>0.3370</b>	<b>0.4079</b>	0.1822	0.3540	<b>0.4479</b>	<b>0.1700</b>	<b>0.3454</b>	<b>0.4351</b>
	mp	0.1844	0.3562	0.4614	<b>0.1797</b>	<b>0.3516</b>	0.4668	0.1999	0.3718	0.4955

accuracy of both tasks, and its robustness with respect to different numbers of corrupted and noisy image cameras.

2) *Effects of Overlap Between Robots*: In Table II, we study how the overlapping areas between robots' FoV affect the proposed message passing with spatial encoding. We use **Industrial-circle** dataset and **Industrial-rotation** for this study. We find that our method **mp-pose** outperforms **baseline** for both datasets, which indicates that our method is robust to different sizes of overlapping areas across robots' FoV in the robot network. Comparing the performance gap between **mp-pose** and **mp** on both datasets, we can find that this is smaller for **Industrial-circle** dataset rather than **Industrial-rotation**, especially for the mIoU metric in the semantic segmentation case. This result shows that the message passing with spatial encoding can be affected by the overlapping pattern. Therefore, for limited overlapping areas, we correctly obtain reduced benefits from the message passing mechanism.

3) *Real-World Experiments*: We demonstrate our approach on a real-world dataset **ARPL-r4**. We use onboard visual-inertial odometry to calculate the relative pose between robots. Since we have access to the relative spatial relationship between robots, we use messages with spatial encoding. In Table. III, we show that our method **mp-pose** with spatial encoded message

outperforms the baseline as well as the one without spatial encoding in most cases.

## E. Discussion

In the following, we discuss the bandwidth requirements for the proposed and the corresponding benefits compared to a direct exchange of sensor data among the robots as well as the advantages and disadvantages for the two encoding mechanisms. We employ MBytes per frame (MBpf) as the metric to measure the bandwidth requirements of our approach. By learning and communicating encoded messages, the communication bandwidth for one message between two robots for one level is 2.5 Mbpf, while the communication bandwidth of directly sharing raw sensor measurements is 6 Mbpf. The levels of the GNN are usually limited to prevent over-smoothing of the GNN. In the demonstrated case, we do not have more than 2 levels which makes the approach communication-efficient than sharing the original information. Also, the message passing can reduce the computation requirements by spreading the computations to each agent, which makes the computation requirements feasible for the robot team. This further shows the ability to scale the proposed approach to larger teams. It is possible to further reduce



communication bandwidth by adopting smaller network encoder or learning communication groups as shown in [17]. For other sensing modalities, similar bandwidth trade-offs are expected.

The bandwidth requirements is equivalent for the two proposed encoding mechanisms, but these have different properties. The results in Section V confirm that the spatial encoding can utilize relative poses when this information is accessible, but it cannot dynamically capture the correlation between the sensor measurements. Conversely, the dynamic cross attention encoding can utilize the correlation relationship among features when the relative pose is not accessible. Geometric-aware tasks like monocular depth estimation benefits more by spatial encoding since the relative poses contain important geometric cues, while semantic-aware tasks like semantic segmentation do not favor either one of the mechanism. In the future, it would be interesting to investigate how to combine the advantages of both mechanisms.

## VI. CONCLUSION

In this work, we proposed a multi-robot collaborative perception framework with graph neural networks. We utilize message passing mechanism with spatial encoding and cross attention encoding to enable information sharing and fusion among the robot team. The proposed method is a general framework which can be applied to different sensor modalities and tasks. We validate the proposed method on semantic segmentation and monocular depth estimation tasks on simulated and real-world datasets collected from multiple aerial robots' viewpoints including different types of severe sensor corruptions and noises. The experiment shows that the proposed method increases the perception accuracy and robustness to sensor corruption and noises. It can be an effective solution for different multi-view tasks with swarms of aerial or ground robots.

In the future, we will extend this framework to other sensor modalities and tasks. We will also explore how to use the graph neural network to integrate perception, control and planning framework for aerial swarm systems.

## REFERENCES

- [1] S. Garg *et al.*, "Semantics for robotic mapping, perception and interaction: A survey," *Found. Trends Robot.*, vol. 8, no. 1, pp. 1–224, 2020.
- [2] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
- [3] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.
- [4] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018.
- [5] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph neural networks for decentralized multi-robot path planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 11785–11792.
- [6] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro, "Learning decentralized controllers for robot swarms with graph neural networks," in *Proc. Conf. Robot Learn. PMLR*, 2020, pp. 671–682.
- [7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
- [8] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervention*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., 2015, pp. 234–241.
- [9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
- [10] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2366–2374.
- [11] C. Godard, O. M. Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3827–3837.
- [12] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 945–953.
- [13] X. Wei, R. Yu, and J. Sun, "View-GCN: View-based graph convolutional network for 3D shape analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1847–1856.
- [14] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield, "Multi-view fusion for multi-level robotic scene understanding," Mar. 2021, *arXiv:2103.13539*.
- [15] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "CosyPose: Consistent multi-view multi-object 6D pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 574–591.
- [16] Y.-C. Liu, J. Tian, C.-Y. Ma, N. Glaser, C.-W. Kuo, and Z. Kira, "Who2com: Collaborative perception via learnable handshake communication," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6876–6883.
- [17] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira, "When2com: Multi-agent perception via communication graph grouping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4106–4115.
- [18] M. S. Schlichtkrull *et al.*, "Modeling relational data with graph convolutional networks," in *Proc. Eur. Semantic Web Conf.*, 2018, pp. 593–607.
- [19] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.
- [20] Q. Li, W. Lin, Z. Liu, and A. Prorok, "Message-aware graph attention networks for large-scale multi-robot path planning," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5533–5540, Jul. 2021.
- [21] A. Khan, A. Ribeiro, V. Kumar, and A. G. Francis, "Graph neural networks for motion planning," Dec. 2020, *arXiv:2006.06248*.
- [22] R. Kortvelesy and A. Prorok, "ModGNN: Expert policy approximation in multi-agent systems with a modular graph neural network architecture," in *Proc. IEEE Int. Conf. Robot. Auto.*, 2021, pp. 9161–9167.
- [23] F. Xue, X. Wu, S. Cai, and J. Wang, "Learning multi-view camera relocation with graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11372–11381.
- [24] A. Elmooghy, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Pose-GNN: Camera pose estimation system using graph neural networks," Mar. 2021, *arXiv:2103.09435*.
- [25] M. Pavliv, F. Schiano, C. Reardon, D. Floreano, and G. Loianno, "Tracking and relative localization of drone swarms with a vision-based headset," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1455–1462, Apr. 2021.
- [26] S. Güler, M. Abdelkader, and J. S. Shamma, "Infrastructure-free multi-robot localization with ultrawideband sensors," in *Proc. Amer. Control Conf.*, 2019, pp. 13–18.
- [27] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5738–5746.
- [28] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, vol. 32, no. 1.
- [29] M. Wang *et al.*, "Deep graph library: A graph-centric, highly-performant package for graph neural networks," 2020, *arXiv:1909.01315*.
- [30] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.
- [31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.
- [32] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *Proc. Int. Conf. Learn. Representations*, 2019.
- [33] Y. Song, S. Naji, E. Kaufmann, A. Loquercio, and D. Scaramuzza, "Flightmare: A flexible quadrotor simulator," in *Proc. Conf. Robot Learn.*, 2020.
- [34] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, "Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and IMU," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 404–411, Apr. 2017.