Non-Overlap-Aware Egocentric Pose Estimation for Collaborative Perception in Connected Autonomy

Hong Huang[1], Dongkuan Xu[2], Hao Zhang[3], and Peng Gao[2]

***Abstract*— Egocentric pose estimation is a fundamental capability for multi-robot collaborative perception in connected autonomy, such as connected autonomous vehicles. During multi-robot operations, a robot needs to know the relative pose between itself and its teammates with respect to its own coordinates. However, different robots usually observe completely different views that contains similar objects, which leads to wrong pose estimation. In addition, it is unrealistic to allow robots to share their raw observations to detect overlap due to the limited communication bandwidth constraint. In this paper, we introduce a novel method for *Non-Overlap-Aware Egocentric Pose Estimation* (NOPE), which performs egocentric pose estimation in a multi-robot team while identifying the non-overlap views and satifying the communication bandwidth constraint. NOPE is built upon an unified hierarchical learning framework that integrates two levels of robot learning: (1) high-level deep graph matching for correspondence identification, which allows to identify if two views are overlapping or not, (2) low-level position-aware cross-attention graph learning for egocentric pose estimation. To evaluate NOPE, we conduct extensive experiments in both high-fidelity simulation and real-world scenarios. Experimental results have demonstrated that NOPE enables the novel capability for non-overlapping-aware egocentric pose estimation and achieves state-of-art performance compared with the existing methods.**

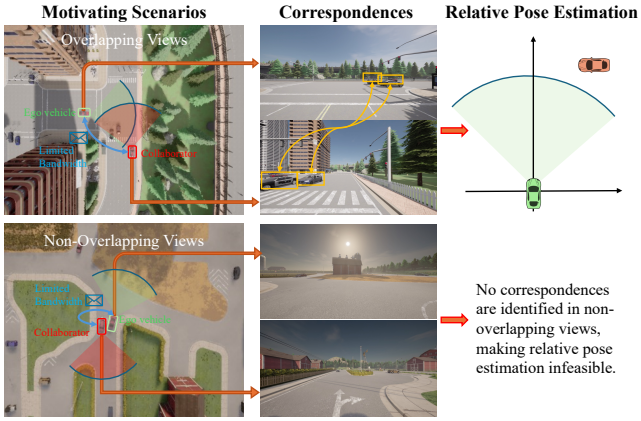


Fig. 1: A motivating scenario for egocentric pose estimation in connected autonomous driving. When two connected vehicles meet at an intersection, the ego vehicle must first estimate the pose of its teammate before merging its perception to enhance situational awareness. Meanwhile, it needs to address the challenges of limited communication bandwidth and non-overlapping views, where each vehicle observes a completely different perspective.

I. Introduction

Multi-robot systems have attracted wide attention in recent decades due to their scalability [1], parallelism [2], and reliability [3]. A fundamental capability in multi-robot systems is collaborative perception, which allows individual robots to share their own perception of the environments, thus leading to a shared situational awareness. It has lots of applications, such as connected autonomous driving [4], [5], collaborative simultaneous localization and mapping (SLAM) [6], [7], [8], and multi-robot search and rescue [9], [10].

To enable efficient collaborative perception, it is essential to achieve accurate egocentric pose estimation that estimates the relative pose between a robot and its teammates with respect to its own coordinate. This allows each robot to determine the poses of its teammates, facilitating the aggregation of multi-robot perception. As shown in Figure 1, when connected autonomous vehicles meet at an intersection, the ego vehicle must first estimate the poses of its collaborators before merging their perceptions to improve situational awareness. This is particularly crucial in urban areas where GPS is unreliable or even unavailable. However, achieving accurate egocentric pose estimation presents two key challenges. The first challenge arises from non-overlapping views, where different robots may observe totally different scenes while their different observations containing similar objects (e.g., traffic signs). This can lead to wrong pose estimations. The second challenge is limited communication bandwidth, which prevents vehicles from sharing raw observations to compare their observations to decide it they are overlapping or not.

Given the importance of egocentric pose estimation, a variety of methods have been studied. Previous techniques for multi-robot relative pose estimation often rely on SLAM, which assumes that robots share a global map [11] or utilize cross-robot loop closure to merge the local maps built by individual robots [12]. However, merging local maps is both time- and bandwidth-intensive and typically struggles to handle dynamic changes in the environment. Recently, vision-based methods have been developed using image registration through feature matching [13], [14] or geometric alignment [15], [16]. However, according to the real-world setting, the maximum bandwidth designated for vehicle-to-everything (V2X) communication is around 7.2 Mbps [17], which is infeasible to share raw observations among robots. In addition, there are non-overlapping views existing among multi-robot observations, which may contains similar objects. A unified

[1]Hong Huang works as a volunteer researcher with Prof. Peng Gao at NCSU. Email: hhong_@outlook.com. [2]Dongkuan Xu and Peng Gao are with the Department of Computer Science, North Carolina State University, Raleigh, NC, USA. Email:{dxu27,pgao5}@ncsu.edu. [3]Hao Zhang is with Human-Centered Robotics Lab at University of Massachusetts Amherst, Amherst, MA, USA Email: hao.zhang@umass.edu.

framework to address all these challenges has not been well addressed yet.

To address these challenges, we propose a novel hierarchical learning approach called *Non-Overlap-Aware Egocentric Pose Estimation* (**NOPE**), which performs egocentric pose estimation in a multi-robot team while identifying their non-overlap views. We represent each observation as a graph with nodes denoting objects associated with visual features extracted from the large vision model and edges denoting the spatial relationships of objects. Given the graph representations, our NOPE approach integrates two levels of robot learning into a hierarchical framework. The high-level NOPE performs correspondence identification (CoID) based on deep graph matching, which determines if two views are overlapped. The low level of NOPE utilizes a position-aware cross-attention network to capture the holistic context of observations for egocentric pose estimation.

The key contribution of the paper is the introduction of a novel approach to perform egocentric pose estimation while satisfying the communication bandwidth and identifying non-overlapping views. The specific novelties include:

- This work presents one of the first learning solutions for multi-robot egocentric pose estimation with the awareness of non-overlapping views. It enables a novel multi-robot capability, allowing the ego robot to estimate the poses of its teammates while detecting non-overlapping views and satisfying the communication bandwidth constraint, thus enhancing situational awareness.
- We introduce a novel hierarchical learning approach that integrates a high-level deep graph matching network for non-overlap detection and a low-level position-aware cross-attention graph learning network for egocentric pose estimation. Our approach achieves over **53**% and **78.6**% improvements on position and rotation estimations, as well as achieves over **96x** reduction in the shared data size that meets the realistic communication bandwidth constraint.

II. Related Work

A. Collaborative Perception

Collaborative perception has gained significant attention in recent research. From an application perspective, collaborative object localization surpasses single-view localization by leveraging multi-robot observations to consistently identify the same objects [18], [19], [20]. This technique enhances accuracy by fusing different viewpoints and addressing occlusions. In addition, collaborative perception plays a crucial role in trajectory forecasting [21], scene segmentation [22], [23], tracking, and object detection [24]. These tasks benefit from associating multi-robot observations to build a richer, more robust environmental understanding. However, existing methods often assume that robots share overlapping observations, such as connected vehicles meeting at an intersection, which may not always be the case in more dynamic and unstructured environments.

From a solution perspective, collaborative perception is typically categorized into three approaches. First, early fusion directly integrates raw sensor data from multiple robots before processing [25]. While this method retains the most information, it heavily depends on high-bandwidth communication, making it impractical in constrained network conditions. Second, intermediate fusion seeks a balance between information-sharing efficiency and computational cost by transmitting compressed feature representations instead of raw data. These methods include when2com [23], who2com [22], and where2com [26], which selectively share relevant features to optimize perception efficiency. However, these methods rely on coordinate transformations based on GPS or pre-existing maps to align observations, making them unreliable in GPS-denied environments or dynamically changing robot teams. Third, late fusion merges independent perception outputs from multiple robots, often using post-processing techniques like Non-Maximum Suppression (NMS) [27] and refined matching for pose consistency [28]. While this approach is robust to noise and requires minimal bandwidth, it ignores most of the useful information in the raw data, which limits its adaptability to unknown observations.

B. Multi-Robot Relative Pose Estimation

The existing methods of multi-robot relative pose estimation can be divided into three categories, including GPS-based methods, SLAM-based methods, and vision-based methods. First, GPS-based methods rely on accurate GPS signals to provide coordinate transformations for estimating relative poses among robots, such as in UAVs [29] and connected autonomous driving [30], [31]. However, GPS is often unreliable in highly dynamic environments and somtimes unavailable. Second, SLAM-based methods assume that the entire robot team maintains a shared global map, with loop closure detection used to estimate egocentric poses relative to this map [11], [32], [33]. However, global maps and reliable loop closures are not always available, especially in large-scale or dynamically changing environments, significantly limiting their applicability. Third, vision-based methods estimate relative poses by registering two observations (e.g., RGB images or point clouds) through feature matching [13], [14], [34], [35] or geometric alignment [15], [16], [36]. While these methods can provide high accuracy, they need substantial communication costs, making them impractical for real-time robotic applications with limited resources.

Recently, foundational models have been widely used as strong priors to various applications due to the generalizability. Foundation visual models, such as DINO [37] and CLIP [38], have been extensively used to extract meaningful and generalizable representations for relative pose estimation [39]. Even though it achieves promising performance on egocentric pose estimation, it still cannot address scenarios where the views of two robots have no overlap.

III. Approach

A. Problem Definition

For each robot, we represent its observations as a graph $\mathcal{G}(\mathcal{V}, \mathcal{F}, \mathcal{E})$. The node set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ contains the objects observed by the robot, where $v_i \in \mathcal{V}$ denotes

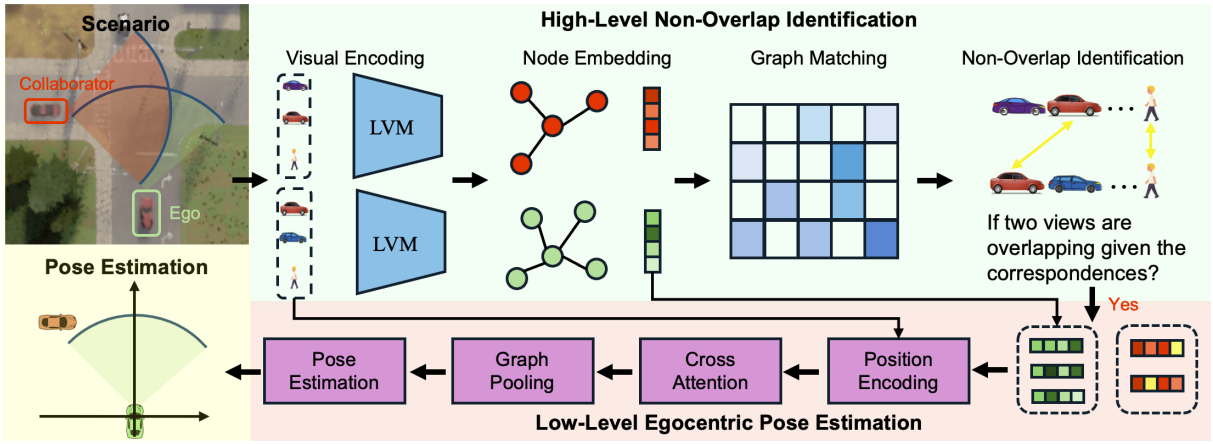


Fig. 2: Overview of our NOPE framework. NOPE represents the observation of each robot as a graph. The high-level NOPE performs CoID based on LVM-based deep graph matching. The identified correspondences are used to detect the overlapping views. The low-level of NOPE utilizes a position-aware cross-attention graph learning network to perform pose estimation between the ego robot and its teammate robot.

the 3D position of the i-th detected object (e.g., a vehicle or a pedestrian). Each object is associated with a visual feature vector, denoted as $\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n\}$, where $\mathbf{f}_i$ represents the visual feature vector of the i-th object. The edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ defines the spatial relationships of objects, where nodes v_i and v_j are connected if through the geometric rules of Delaunay triangulation, there exists a direct connection relationship between i-th and j-th objects in the geometric space. Consequently, the adjacency matrix $\mathbf{A}$ can be derived, where $\mathbf{A}_{i,j} = \|v_i - v_j\|_2$ if nodes v_i and v_j are connected; otherwise, $\mathbf{A}_{i,j} = 0$. The graph representation can significantly reduce the size of data shared among robots, thus satisfying the communication bandwidth constraints.

Given the graph representations $\mathcal{G}$ and $\mathcal{G}'$ provided by a pair of robots, we aim to address the problem of egocentric pose estimation with the identification of non-overlapping views:

1) **Non-Overlap Detection**: The capability of a robot team identifying if their views are overlapped or not solely based on their visual perception.
2) **Egocentric Pose Estimation**: The capability of a robot to estimate the pose of its robot teammates with respect to its own egocentric coordinates.

B. High-Level Correspondence Identification for Non-Overlap Detection

Given the graph representations $\mathcal{G}$ and $\mathcal{G}'$, we formulate non-overlap detection as a graph matching problem, which identifies the correspondences of objects in different views to determine if two views are overlapped or not. First, we utilize large vision models (LVMs), such as DINO [37] or CLIP [38], to extract visual features of objects. Formally, it is defined as

$$\mathbf{f}_i = \Phi(I, v_i), \tag{1}$$

where $\Phi(\cdot)$ denotes the LVM-based encoder, I is the observation of a robot, and v_i denotes the i-th object observed by the robot. Then, we employ a Transformer-based graph attention network $\{\mathbf{h}_i\}^n = \Psi(\mathcal{F}, \mathbf{A})$ to compute node embeddings, where $\mathbf{h}_i$ denotes the embedding of the i-th object, which does not just consider its own visual feature but also aggregating its neighbors. Formally, we compute the linear projection of the embedding of the i-th object as follows:

$$\mathbf{q}_i^l = \mathbf{W}^{l,q}\mathbf{h}_i, \quad \mathbf{k}_i^l = \mathbf{W}^{l,k}\mathbf{h}_i, \quad \mathbf{v}_i^l = \mathbf{W}^{l,v}\mathbf{h}_i, \tag{2}$$

where $\mathbf{q}_i^l$, $\mathbf{k}_i^l$, and $\mathbf{v}_i^l$ represent the query, key, and value vectors of the i-th object at layer l. The trainable weight matrices corresponding to these transformations are denoted as $\mathbf{W}^{l,q}$, $\mathbf{W}^{l,k}$, and $\mathbf{W}^{l,v}$. Notably, the initial input is defined as $\mathbf{h}_i^0 = \mathbf{f}_i$. The attention between pairs of nodes is computed as follows:

$$\alpha_{i,j}^l = \frac{\exp((\mathbf{q}_i^l)^\top(\mathbf{k}_j^l + \mathbf{W}^{l,e}\mathbf{A}_{i,j}))}{\sqrt{d}\cdot\sum_{k\in\mathcal{N}(i)}\exp((\mathbf{q}_i^l)^\top(\mathbf{k}_k^l + \mathbf{W}^{l,e}\mathbf{A}_{i,k}))}, \tag{3}$$

where $\mathcal{N}(i)$ denotes the set of neighboring nodes of v_i, $\mathbf{W}_e^l$ denotes a trainable weight matrix, and d denotes the length of query vector. $\alpha_{i,j}^l$ denotes the attention from the i-th node to the j-th node, which is computed by comparing the query of the i-th node and its neighbors. The adjacent matrix $\mathbf{A}_{i,k}$ is added into the learning process to encode the spatial relationships of nodes. Then, SoftMax is used to normalize the attention. The final node embedding is computed as follows:

$$\mathbf{h}_i^{l+1} = \text{LayerNorm}\left(\mathbf{W}^{l,h}\mathbf{h}_i^l + \Big\|_{m=1}^{M}\sum_{j\in\mathcal{N}(i)}\alpha_{i,j}^l\mathbf{v}_j^l\right), \tag{4}$$

where the $\|$ is the concatenation operation for M head attention, LayerNorm denotes layer normalization operation and $\mathbf{W}^{l,h}$ denotes a trainable weight matrix. The final node embedding is computed by aggregating the central node embedding and its neighborhood node embeddings weighted by attention coefficients. Multi-head mechanism generates a richer representation of the embedding by capturing different embedding spaces and layer normalization standardizes the embeddings to improve training stability and convergence.

Given the node embeddings $\mathbf{h}_i^L$ and $\mathbf{h}_j^L$ from the final layer L, we compute pairwise correspondences between graphs $\mathcal{G}$ and $\mathcal{G}'$. A similarity matrix $\mathbf{S}$ is computed as:

$$\mathbf{S}_{i,j} = \mathbf{h}_i^L (\mathbf{h}_j^L)^\top, \tag{5}$$

where $\mathbf{S} \in \mathbb{R}^{n \times m}$ represents the similarity matrix between two graphs containing n and m objects, respectively. To improve the robustness of CoID, a graph difference matrix $\mathbf{D}$ is to update the similarity matrix $\mathbf{S}$, which is defined as:

$$\mathbf{D} = \left(\mathbf{S}^\top \Psi(\mathbf{J}, \mathbf{A}) - \Psi(\mathbf{S}^\top \mathbf{J}, \mathbf{A}')\right)^\top, \tag{6}$$

where $\mathbf{J} \in \mathbb{R}^{n \times r}$ is a random matrix. According to graph consensus theorem [40], when the graphs $\mathcal{G}$ and $\mathcal{G}'$ represent the same graph, then $\mathbf{S}^\top \Psi(\mathbf{J}, \mathbf{A}) = \Psi(\mathbf{S}^\top \mathbf{J}, \mathbf{S}^\top \mathbf{A}\mathbf{S}) = \Psi(\mathbf{S}^\top \mathbf{J}, \mathbf{A}')$, thus $\mathbf{D}_{i,j} = 0$. The larger the difference between two graphs, the large values in the difference matrix $\mathbf{D}_{i,j} = 0$. The updated similarity matrix is defined as:

$$\hat{\mathbf{S}}_{i,j} = \epsilon\left(\left(\mathbf{S}_{i,j} + \mathbf{D}_{i,j}\right), \tau\right), \tag{7}$$

where $\hat{\mathbf{S}} \in \mathbb{R}^{n \times m}$ represents the final similarity matrix between two graphs. $\epsilon(\cdot)$ is an indicator function that outputs 1 when $\mathbf{S}_{i,j} + \mathbf{D}_{i,j} \geq \tau$, otherwise 0. τ denotes a threshold. The final correspondences of objects are identified as follows:

$$\begin{aligned} \mathbf{Y} &= \text{argmax}_{\mathbf{Y}} \sum_{i=1}^{n} \sum_{j=1}^{m} \hat{\mathbf{S}}_{ij} \cdot \mathbf{Y}_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{m} \mathbf{Y}_{ij} \leq 1, \qquad \sum_{i=1}^{n} \mathbf{Y}_{ij} \leq 1 \end{aligned} \tag{8}$$

where $\mathbf{Y} \in \{0,1\}^{n \times m}$ denotes the identified correspondences of objects observed by two robots, with $\mathbf{Y}_{i,j} = 1$ denoting the i-th object observed by the ego robot and the j-th object observed by its teammate are the same. The final correspondences are optimized by maximizing the overall similarity given the similarity matrix $\hat{\mathbf{S}}$. The constraint enforce that one object can at most have one corresponding object in the other observation, thus allowing to remove non-covisible objects that can only be observed by one robot. We use the Hungarian algorithm [41] to solve this optimization problem.

Given the correspondence matrix $\mathbf{Y}$, we determine whether there exists an overlap between two observations. Specifically, if the sum of all elements in $\sum_{i,j} \mathbf{Y} = 0$, it indicates that there is no overlap between two robots' views due to the lack of correspondences. If $\sum_{i,j} \mathbf{Y} \geq 1$, it implies an overlapping views between a pair of robots. As non-overlapping views significantly affect the pose estimation accuracy due to the lack of correlated contextual information, we only perform egocentric pose estimation when two observations are decided to be overlapping. We train the high-level CoID network with the following loss function:

$$\mathcal{L}^{high} = \frac{\sum_{i,j} (\hat{\mathbf{S}}_{i,j} - \mathbf{Y}^*_{i,j})}{n \cdot m}, \tag{9}$$

where $\mathbf{Y}^* \in \mathbb{R}^{n \times m}$ denotes the ground truth correspondence matrix. If the i-th object in one observation and the j-th object in the other observation are the same, then: $\mathbf{Y}^*_{i,j} = 1$, otherwise 0. The correspondence is optimal when the loss is minimum.

C. Low-Level Position-Aware Graph Learning for Egocentric Pose Estimation

Once two observations are decided to be overlapping given the high-level CoID results, we design a low-level network based on position-based cross-attention mechanism to estimate the relative poses between the ego robot and its teammates. To capture the holistic information of the observation for egocentric pose estimation, we compute the graph-level embeddings that captures the whole visual-spatial information of the observation as a single vector, meanwhile considering positional cues of node embeddings and the correlation between two observations, to improve the expressiveness of graph embeddings.

First, we explicit encode the order of node embeddings as $\mathbf{P}_i = \mathbf{U}[i, :]$, where $\mathbf{P}_i$ denotes the position embedding of the i-th object in observations $\mathcal{G}$ and $\mathbf{U}$ denotes learnable embedding matrices. During training, the parameters of $\mathbf{U}$ are optimized to capture meaningful positional information, allowing the model to learn an effective representation of position embeddings for each object in the sequence. By incorporating position embeddings, our model gains the ability to discern the relative positions of objects in observations, thereby enhancing the accuracy of egocentric pose estimation.

Second, we compute a set of node embeddings $\mathbf{H}$ of graph $\mathcal{G}$ by concatenating the node embeddings $\{\mathbf{h}_i\}^n$ and the further update it by combining the position embeddings:

$$\hat{\mathbf{H}}_i = \mathbf{H}_i + \mathbf{P}_i. \tag{10}$$

Third, we use cross-attention mechanism to capture the attention of the relevant objects between two observations with partial overlap, thus improving the egocentric pose estimation. Formally, given the sets of node embeddings $\hat{\mathbf{H}}$ and $\hat{\mathbf{H}}'$ of graphs $\mathcal{G}$ and $\mathcal{G}'$, we compute the cross attention to capture the correlations of nodes embeddings as follows:

$$\text{CrossAtt}(\hat{\mathbf{H}}, \hat{\mathbf{H}}') = \text{SoftMax}\left(\frac{\hat{\mathbf{H}}\mathbf{W}^Q (\hat{\mathbf{H}}'\mathbf{W}^K)^\top}{\sqrt{d}}\right), \tag{11}$$

where $\mathbf{W}^Q$, $\mathbf{W}^K$ and $\mathbf{W}^V$ denotes the trainable weight matrices. By comparing the similarity between the sets of node embeddings $\hat{\mathbf{H}}$ and $\hat{\mathbf{H}}'$, the cross attention between two observations $\mathcal{G}$ and $\mathcal{G}'$ is computed through a SoftMax function. Then we update the set of node embedding as follows:

$$\mathbf{H}_{\text{out}} = \text{LayerNorm}(\hat{\mathbf{H}} + \text{MLP}(\Big\|_{m=1}^{M} \text{CrossAttn}(\hat{\mathbf{H}}, \hat{\mathbf{H}}')_m \cdot \hat{\mathbf{H}}\mathbf{W}^V), \tag{12}$$

We use multi-head mechanism followed by a MLP to update the set of node embeddings based on cross attention. Then the updated graph embedding and the original graph embedding $\hat{\mathbf{H}}$ are added up and pass through a layer normalization to generate $\mathbf{H}_{\text{out}}$.

Finally, we employ an attention gate aggregation operation to compute the graph-level embeddings. Specifically, we apply self-attention to $\mathbf{H}_{\text{out}}$, which captures the weighted

relationships between nodes by considering the interactions and dependencies within the node embeddings.

$$\mathbf{H}_{\text{weight}} = \text{SoftMax}(\frac{\mathbf{H}_{\text{out}}\mathbf{W}^{Q}(\mathbf{H}_{\text{out}}\mathbf{W}^{K})^{\top}}{\sqrt{d}})\mathbf{H}_{\text{out}}\mathbf{W}^{V}, \quad (13)$$

Then, we pass $\mathbf{H}_{\text{weight}}$ through a multi-layer perceptron (MLP) followed by a SoftMax function to obtain the attention gate scores $\mathbf{g}$.

$$\mathbf{g} = \text{SoftMax}(\text{MLP}(\mathbf{H}_{\text{weight}})), \quad (14)$$

where $\mathbf{g}$ denotes the attention gate score, which captures the importance of each node for graph embedding. The graph embedding is computed by pooling node embeddings $\mathbf{H}_{\text{out},i}$ weighted by the attention gate scores $\mathbf{g}_i$.

$$\mathbf{h}_{\text{pooled}} = \sum_{i=1}^{n} \mathbf{g}_i \cdot \mathbf{H}_{\text{out},i}, \quad (15)$$

where $\mathbf{h}_{\text{pooled}}$ denotes the graph embedding of the graph $\mathcal{G}$, which captures all visual-spatial cues of nodes while considering the correlation of observations provided by the ego robot and its collaborator. Given the graph embedding, we estimate the egocentric pose as follows:

$$(\hat{\mathbf{p}}, \hat{\mathbf{R}}) = \text{MLP}(\mathbf{h}_{\text{pooled}}). \quad (16)$$

where $\hat{\mathbf{p}}$ and $\hat{\mathbf{R}}$ denote the position and rotation of the collaborator providing observation $\mathcal{G}'$. The egocentric pose of an ego vehicle's collaborator is estimated from the position-aware cross-attention graph embedding $\mathbf{h}_{\text{pooled}}$ followed by a MLP. The loss function to train the low-level egocentric pose estimation is defined as follows:

$$\mathcal{L}^{Low} = \overbrace{\|\hat{\mathbf{p}} - \mathbf{p}\|_2^2}^{\mathcal{L}_{pos}} + \overbrace{2 \cdot \|\hat{\mathbf{R}} - \mathbf{R}\|_2^2 \cdot (4 - \|\hat{\mathbf{R}} - \mathbf{R}\|_2^2)}^{\mathcal{L}_{rot}}. \quad (17)$$

The first term denotes the position loss which is compute by minimizing the Euclidean distance between the predicted position $\hat{\mathbf{p}}$ and the ground truth position $\mathbf{p}$. The second term denotes the loss of rotation estimation with respect to the ego robot coordinates. It is based on the chordal squared loss [39] to measure the difference between the quaternion-based rotation estimation $\hat{\mathbf{R}}$ and the ground truth $\mathbf{R}$.

IV. Experiments

A. Experimental Setup

We conducted experimental evaluation in both high-fidelity simulation and the real world. In the simulation, we utilize both CARLA [42] and SUMO [43] to create five connected autonomous driving (CAD) scenarios. In each scenario, a pair of connected vehicles are deployed. The behaviors of vehicles and pedestrians were controlled by SUMO in accordance with real-world rules, such as stopping at red lights and yielding to pedestrians. For each vehicle, it is equipped with a front-facing RGB-D camera and a Global Navigation Satellite System (GNSS) sensor. In the real-world application, we utilize the multi-modal autonomous driving dataset MARS [44], which was collected by a fleet of autonomous vehicles operating within a specific geographic area. Each vehicle follows its own route, with different vehicles potentially appearing in nearby locations. Each vehicle was equipped with one LiDAR, three narrow-angle RGB cameras, three wide-angle RGB fisheye cameras, one IMU, and one GPS. All sensor data were sampled at 10Hz to ensure synchronization.

In the simulation, we collect a total of 30,277 data instances, of which 27,247 were used for training and 3,030 for testing. Each data instance includes a pair of RGB-D images captured from different perspectives by two connected vehicles. The ground truth of object correspondences is provided by the CARLA simulation and the ground truth of positions and orientations of connected vehicles is provided by the GNSS sensor. In the real-world application, we select 201 data instances. Each data instance consists of a pair of RGB images captured from different perspectives by two connected vehicles, along with the positions and orientations of the vehicles provided by GPS.

For graph construction, we use YOLOv5 [45] to detect objects in each vehicle's view and extract visual features as node representations. Delaunay triangulation generates edges and DepthAnythingV2 [46] estimates depth information, which is used to compute edge attributes based on object locations. In simulation CAD, vehicle positions and rotations are represented in XYZ-pitch-roll-yaw. In real-world CAD, they are in XYZ format with quaternion rotation.

In the specific implementation details, we implemented the Transformer-based graph attention etwork Ψ using PyTorch and PyG [47]. In this network, we set the number of network layers as $L = 2$, with the number of heads as heads $= 4$, and all dimensions $d = 256$. Additionally, the edge feature dimension is set to dim $= 1$. After each attention layer, we applied a dropout with a probability of 0.5. For the MLP with two linear layers, each layer has a dropout probability of 0.2. In the position-aware cross-attention network, the number of network layers is $L = 4$, with heads $= 4$, and $d = 256$. The attention gate aggregation network has $L = 1$ layers, with heads $= 4$, and $d = 256$. In all experiments, we used Adam as the optimizer [48], with a learning rate of 0.001. We ran the training for 150 epochs.

We implement a baseline method **NOPE**$_{\text{high}}$ that just use the high-level network to evaluate the CoID performance. In addition, compare our method with six existing methods, including: 1) **GCN-GM** [49] aggregating visual-spatial information of objects and their neighborhoods via spline kernels for graph matching, 2) **DGMC** [40] identifying initial correspondences based on visual similarity and graph matching consensus for CoID, 3) **BDGM** [50] performing deep graph matching under a Bayesian framework to remove invisible objects given the quantified correspondence uncertainty, 4) **DMGM** [30] considering visual-spatial cues, matching consensus and uncertainty of correspondences for CoID, 5) **SuperGlue** [34] optimizing feature matching as a differentiable optimal transport problem, which leverages Transformer-based graph neural networks to estimate relative poses between two graphs, 6) **CoViS-Net** [39] is a foundation model for egocentric pose prediction, which utilizes DINOV2 as encoder on individual robots and generates bird-eye-

TABLE I: Quantitative results of egocentric pose estimation in both simulation and the real world based on metrics of position error (PE), rotation error (RE) and package size (PS). The improvements is computed w.r.t CoViS-Net [39]

Method	CAD Simulation		Real World		PS ↓
	PE ↓	RE ↓	PE ↓	RE ↓	
GCN-GM [49]	20.32	3.641	19.77	5.541	36.7KB
DGMC [40]	20.17	3.120	18.15	4.620	36.2kB
BDGM [50]	19.35	2.944	17.81	4.357	36.5KB
DMGM [30]	18.69	2.256	17.28	4.621	35.1KB
SuperGlue [34]	18.80	2.406	15.03	3.679	2.3MB
CoViS-Net [39]	13.92	2.037	14.52	3.120	0.75MB
NOPE	**6.42**	**0.435**	**13.63**	**2.226**	**27.0KB**
Improvements(%)	53.87	78.64	6.32	28.65	96.48

view map to encode poses of a robot team. As **GCN-GM**, **DGMC**, **BDGM** and **DMGM** do not have the capability of estimating poses, we use learning-free pose estimation method RANSAC [51] and essential matrix decomposition to estimate the poses given the correspondences identified.

We use the following metrics to evaluate our NOPE, including 1) Precision is defined as the ratio of correctly retrieved correspondences to the retrieved correspondences, 2) Recall is defined as the ratio of correctly retrieved correspondences to the ground truth correspondences, 3) F1-Score, which evaluates the overall performance of the CoID method, is calculated as F1=$\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$, 4) Position Error (**PE**) measures the Euclidean distance between the estimated and the ground truth position, 5) Rotation Error (**RE**) is defined as the geodesic distance [39] between the estimated rotation and the ground truth position, 6) Packet Size ((**PS**)) refers to the size of the data transmission packets shared between connected vehicles, to evaluate communication efficiency, 7) Non-Overlapping Detection Accuracy (**NDA**)) is defined as the ratio of correctly detected non-overlapping observation pairs to the total number of observation pairs, to evaluate accuracy of non-overlapping identification.

B. Results over Connected Autonomous Driving Simulations

The CAD simulation includes a lots of challenges to perform egocentric pose estimation, including highly dynamic street objects (e.g., pedestrians and vehicles) with ambiguous visual appearance caused by occlusion and long-distance observation, a large number of non-covisible objects, limited communication bandwidth, as well as non-overlapping views between pairs of connected vehicles. We run our approach on a Linux machine with an i7 32-core CPU and 16G memory. The average execution time is around 20Hz.

As shown in Figure 3(a), for CoID, we can clearly see that our NOPE outperforms existing methods BDGM and DMGM. This is because of the integration of LVMs and addressing non-covisible objects in NOPE. For pose estimation, the keypoint-based methods, SuperGlue and CoVisNet, can not well address the ambiguity in visual appearance of objects caused by long-distance observation and low resolution, which leads to poor pose estimations. Moreover, as none of these existing method can address non-overlapping views, NOPE achieves the best performance of egocentric pose estimation, which indicates the importance of addressing visual ambiguity, non-covisiblity and non-overlapping views for egocentric pose estimation in collaborative perception.

The quantitative results are shown in Table I. We observe that GCN-GM, DGMC, BDGM, and DMGM exhibit large pose errors. This is primarily because they focus on correspondences of objects and rely on RANSAC for pose estimation, which requires a large number of correct correspondences of objects. SuperGlue and CoVisNet achieve better performance by learning keypoint-based matching and pose estimation. However, these methods are highly sensitive to non-overlapping views with similar visual features. NOPE outperforms CoVisNet, the second-best method, by 53.9% in position estimation and 78.6% in rotation estimation, while requiring only 1/96 of the data size for sharing.

TABLE II: Quantitative results of CoID in the simulation CAD based on metrics of precision, recall and non-overlapping detection accuracy (NDA).

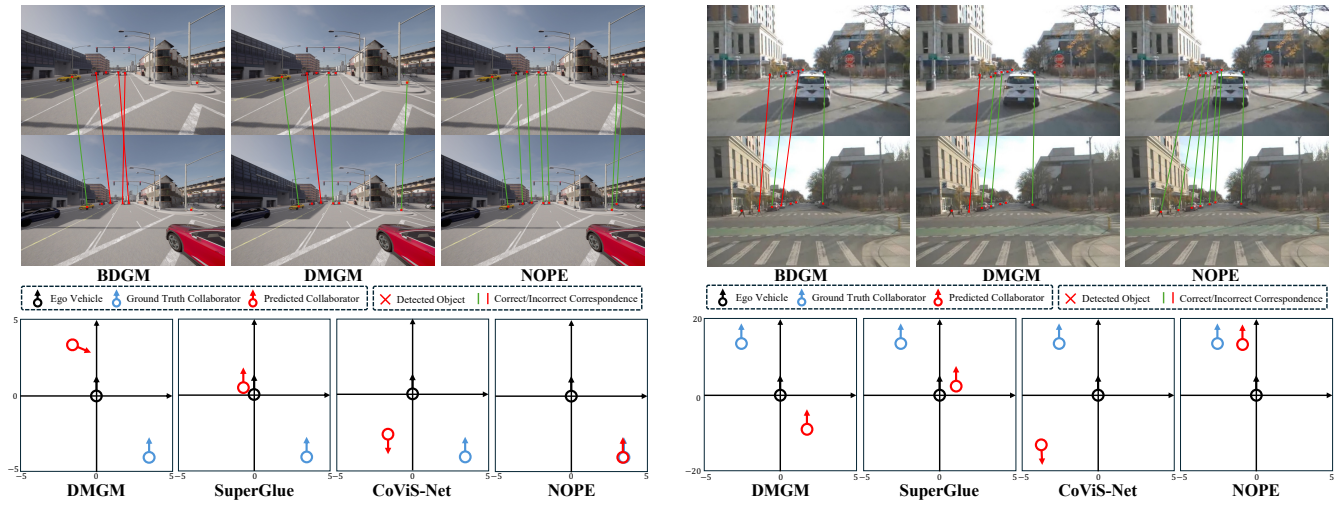
Method	Precision ↑	Recall ↑	F1-score ↑	NDA ↑
GCN-GM [49]	0.5001	0.6391	0.5611	0.6539
DGMC [40]	0.4736	0.6425	0.5453	0.6857
BDGM [50]	0.6817	0.6097	0.6437	0.7239
DMGM [30]	0.7859	0.8278	0.8063	0.7561
NOPE$_{\text{high}}$	**0.8224**	**0.8429**	**0.8325**	**0.8039**

We further evaluate NOPE's high-level CoID for non-overlapping view detection in CAD simulation. Table II indicate that GCN-GM and DGMC suffer from low recall due to their inability to handle non-covisible objects. BDGM prioritizes precision at the cost of recall based on the thresholding on the correspondence uncertainty but it uses hand-craft features for graph matching. By integrating LVMs and address non-covisible objects, NOPE surpasses all existing methods on all metrics with the improvements of 4.6%, 1.8%, and 3.2% on precision, recall and F1 score respectively. According to the metrics of NDA, we can see that our method is able to identify over 80% non-overlap views, which outperforms all the existing methods, which indicates the importance of identifying correct correspondences for the detection of non-overlapping views.

C. Results over Real-World Connected Autonomous Driving

The real-world connected autonomous driving scenario covers the challenges like low-resolution observations, limited communication bandwidth, non-covisible objects and non-overlapping views between connected vehicles. In addition, we do not fine tune NOPE with the real-world data and directly use the model learned from the CAD simulation to evaluate the generalizability of NOPE.

The qualitative results shown in Figure 3(b) illustrates that other methods shows significant errors in complex and noisy real-world scenarios in terms of both CoID or egocentric pose estimation. In addition, NOPE still outperforms the existing methods without the needs of fine tuning, which indicates its generalizability to real-world applications.



(a) Simulation CAD

(b) Real-world CAD

Fig. 3: Qualitative results on CoID and egocentric pose estimation from both simulation and real-world scenarios. The first row illustrates identified correspondences between the ego robot and its collaborator's observations. The second row compares the estimated and ground truth poses of the collaborator in the ego vehicle's coordinate frame.

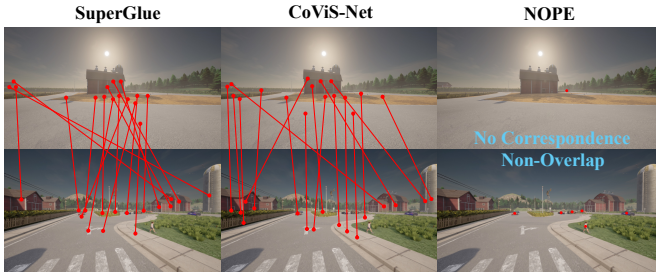


Fig. 4: Comparisons of CoID for non-overlap detection.

Table I provides the quantitative results of egocentric pose estimation. We can observe that NOPE continuously maintains a low pose error compared with the second best method CoViS-Net. Although its rotation error is 2.226 which slightly worse in the real-world scenario compared to its performance in the simulation, likely due to sensitivity to dynamic lighting, it still outperforms other methods. Furthermore, compared to feature-matching-based methods like SuperGlue and CoViS-Net, NOPE achieves the best results under realistic communication bandwidth constraints.

D. Discussion

Non-Overlapping Detection: Figure 4 demonstrates that NOPE can detect the non-overlapping views by making decisions on the identified correspondences. When there is no identified correspondences of objects, NOPE determinate that two observations are non-overlapping, thus will not estimate the poses between two views. Notably, SuperGlue and CoViS-Net rely on the keypoint feature matching, which can not work well with non-overlapping observations, particularly in different observations with similar visual context (e.g., traffic signs or buildings), which emphasizes the importance of CoID for non-overlapping view identification.

Hyperparameter Analysis: Figure 5 shows the performance of our high-level CoID for non-overlapping detection with the variance of threshold τ, defined in Eq. (7). We can see that the highest non-overlap detection accuracy is achieved when τ is in the range of $[0.6, 0.7]$ with small fluctuation. Similarly, the F1 score reach to the highest when τ is in the range of $[0.6, 0.7]$, which indicates positively correlation between the non-overlapping view detection and CoID.

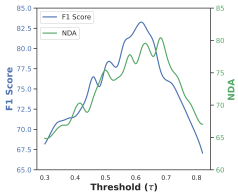


Fig. 5: Analysis of τ.

V. Conclusion

In this paper, we propose NOPE as a novel method to enable non-overlap-aware egocentric pose estimation for collaborative perception in multi-robot systems. NOPE integrates high-level deep graph matching to detect the overlap between two observations based on the identified correspondences, and low-level position-aware cross-attention network performs egocentric pose estimation. We conduct extensive experiments to evaluate NOPE in both high-fidelity simulation and real-world scenarios. The results demonstrate that NOPE enables new capability of non-overlap-aware egocentric pose estimation and significantly outperforms existing methods on bandwidth cost, non-overlap detection and egocentric pose estimation.

Our approach has several limitations that open avenues for future research. First, NOPE cannot estimate egocentric poses when observations are completely non-overlapping. A possible solution is to integrate Bayesian filters to estimate relative poses, using NOPE's outputs as corrections to refine these estimates. Second, NOPE does not ensure global pose consistency for teams larger than two. Future work could explore distributed consensus algorithms to enable large robot teams to collaboratively achieve consistent pose estimation.

REFERENCES

[1] J. Kuckling, R. Luckey, V. Avrutin, A. Vardy, A. Reina, and H. Hamann, "Do we run large-scale multi-robot systems on the edge? more evidence for two-phase performance in system size scaling," in *ICRA*, 2024.

[2] L. Chen, C. Liang, S. Yuan, M. Cao, and L. Xie, "Relative localizability and localization for multi-robot systems," *TRO*, pp. 1–19, 2025.

[3] H. Park and S. A. Hutchinson, "Fault-tolerant rendezvous of multirobot systems," *TRO*, vol. 33, no. 3, pp. 565–582, 2017.

[4] S. Liu, C. Gao, Y. Chen, X. Peng, X. Kong, K. Wang, R. Xu, W. Jiang, H. Xiang, J. Ma, *et al.*, "Towards vehicle-to-everything autonomous driving: A survey on collaborative perception," *arXiv preprint arXiv:2308.16714*, 2023.

[5] S. Hu, Z. Fang, Y. Deng, X. Chen, and Y. Fang, "Collaborative perception for connected and autonomous driving: Challenges, possible solutions and opportunities," *arXiv preprint arXiv:2401.01544*, 2024.

[6] X. Liu, S. Wen, J. Zhao, T. Z. Qiu, and H. Zhang, "Edge-assisted multi-robot visual-inertial slam with efficient communication," *IEEE Transactions on Automation Science and Engineering*, 2024.

[7] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone, and J. A. Castellanos, "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *TRO*, vol. 39, no. 3, pp. 1686–1705, 2023.

[8] D. Feng, Y. Qi, S. Zhong, Z. Chen, Q. Chen, H. Chen, J. Wu, and J. Ma, "S3e: A multi-robot multimodal dataset for collaborative slam," *RAL*, 2024.

[9] Y. Chang, K. Ebadi, C. E. Denniston, M. F. Ginting, A. Rosinol, A. Reinke, M. Palieri, J. Shi, A. Chatterjee, B. Morrell, *et al.*, "Lamp 2.0: A robust multi-robot slam system for operation in challenging large-scale underground environments," *RAL*, vol. 7, no. 4, pp. 9175–9182, 2022.

[10] G. S. Kashyap, D. Mahajan, O. C. Phukan, A. Kumar, A. E. Brownlee, and J. Gao, "From simulations to reality: enhancing multi-robot exploration for urban search and rescue," *arXiv preprint arXiv:2311.16958*, 2023.

[11] I. D. Miller, A. Cowley, R. Konkimalla, S. S. Shivakumar, T. Nguyen, T. Smith, C. J. Taylor, and V. Kumar, "Any way you look at it: Semantic crossview localization and mapping with lidar," *RAL*, vol. 6, no. 2, pp. 2397–2404, 2021.

[12] P. Yin, L. Xu, J. Zhang, and H. Choset, "Fusionvlad: A multi-view deep fusion networks for viewpoint-free 3d place recognition," *RAL*, vol. 6, no. 2, pp. 2304–2310, 2021.

[13] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers," in *CVPR*, 2021.

[14] T. Xie, K. Dai, K. Wang, R. Li, and L. Zhao, "Deepmatcher: a deep transformer-based network for robust and accurate local feature matching," *Expert Systems with Applications*, 2024.

[15] H. Jiang, M. Salzmann, Z. Dang, J. Xie, and J. Yang, "Se (3) diffusion model-based point cloud registration for robust 6d object pose estimation," *NeurIPS*, 2024.

[16] J. Wang, C. Rupprecht, and D. Novotny, "Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment," in *CVPR*, 2023.

[17] L. Gallo and J. Härri, "A lte-direct broadcast mechanism for periodic vehicular safety communications," in *IEEE VNC*, 2013.

[18] P. Gao, B. Reily, R. Guo, H. Lu, Q. Zhu, and H. Zhang, "Asynchronous collaborative localization by integrating spatiotemporal graph learning with model-based estimation," in *ICRA*, 2022.

[19] M. Ji, J. Gall, H. Zheng, Y. Liu, and L. Fang, "Surfacenet: An end-to-end 3d neural network for multiview stereopsis," in *ICCV*, 2017.

[20] P. Gao, R. Guo, H. Lu, and H. Zhang, "Multi-view sensor fusion by integrating model-based estimation and graph learning for collaborative object localization," in *ICRA*, 2021.

[21] H. Zhu, F. M. Claramunt, B. Brito, and J. Alonso-Mora, "Learning interaction-aware trajectory predictions for decentralized multi-robot motion planning in dynamic environments," *RAL*, vol. 6, no. 2, pp. 2256–2263, 2021.

[22] Y.-C. Liu, J. Tian, C.-Y. Ma, N. Glaser, C.-W. Kuo, and Z. Kira, "Who2com: Collaborative perception via learnable handshake communication," in *ICRA*, 2020.

[23] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira, "When2com: Multi-agent perception via communication graph grouping," in *CVPR*, 2020.

[24] C. Robin and S. Lacroix, "Multi-robot target detection and tracking: Taxonomy and survey," *AuRo*, 2016.

[25] E. Arnold, M. Dianati, R. de Temple, and S. Fallah, "Cooperative perception for 3d object detection in driving scenarios using infrastructure sensors," *Intelligence Transportation System*, vol. 23, no. 3, pp. 1852–1864, 2020.

[26] Y. Hu, S. Fang, Z. Lei, Y. Zhong, and S. Chen, "Where2comm: Communication-efficient collaborative perception via spatial confidence maps," *NeurIPS*, 2022.

[27] D. Forsyth, "Object detection with discriminatively trained part-based models," *Computer*, vol. 47, no. 02, pp. 6–7, 2014.

[28] Z. Song, F. Wen, H. Zhang, and J. Li, "A cooperative perception system robust to localization errors," in *IEEE IV*, 2023, pp. 1–6.

[29] X. An, A. Bellés, F. G. Rizzi, L. Hösch, C. Lass, and D. Medina, "Array ppp-rtk: A high precision pose estimation method for outdoor scenarios," *IEEE ITS*, vol. 25, no. 6, pp. 6223–6237, 2023.

[30] P. Gao, Q. Zhu, H. Lu, C. Gan, and H. Zhang, "Deep masked graph matching for correspondence identification in collaborative perception," in *ICRA*, 2023.

[31] P. Gao, Q. Zhu, and H. Zhang, "Uncertainty-aware correspondence identification for collaborative perception," *AuRo*, vol. 47, no. 5, pp. 635–648, 2023.

[32] P. Gao, J. Liang, Y. Shen, S. Son, and M. C. Lin, "Visual, spatial, geometric-preserved place recognition for cross-view and cross-modal collaborative perception," in *IROS*. IEEE, 2023, pp. 11 079–11 086.

[33] P. Gao and H. Zhang, "Long-term loop closure detection through visual-spatial information preserving multi-order graph matching," in *AAAI*, 2020.

[34] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *CVPR*, 2020.

[35] S. Zhang and J. Ma, "Diffglue: Diffusion-aided image feature matching," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024.

[36] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," in *CVPR*, 2022.

[37] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[38] 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 *ICML*, 2021.

[39] J. Blumenkamp, S. Morad, J. Gielis, and A. Prorok, "Covis-net: A cooperative visual spatial foundation model for multi-robot applications," *CoRL*, 2025.

[40] M. Fey, J. E. Lenssen, C. Morris, J. Masci, and N. M. Kriege, "Deep graph matching consensus," *arXiv preprint arXiv:2001.09621*, 2020.

[41] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, pp. 83–97, 1955.

[42] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *CoRL*, 2017.

[43] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "Sumo (simulation of urban mobility)-an open-source traffic simulation," in *MESM*, 2002.

[44] Y. Li, Z. Li, N. Chen, M. Gong, Z. Lyu, Z. Wang, P. Jiang, and C. Feng, "Multiagent multitraversal multimodal self-driving: Open mars dataset," in *CVPR*, 2024.

[45] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, Y. Kwon, J. Fang, K. Michael, D. Montes, J. Nadar, P. Skalski, *et al.*, "ultralytics/yolov5: v6. 1-tensorrt, tensorflow edge tpu and openvino export and inference," *Zenodo*, 2022.

[46] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *arXiv:2406.09414*, 2024.

[47] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[49] M. Fey, J. E. Lenssen, F. Weichert, and H. Müller, "Splinecnn: Fast geometric deep learning with continuous b-spline kernels," in *CVPR*, 2018.

[50] P. Gao and H. Zhang, "Bayesian deep graph matching for correspondence identification in collaborative perception," in *RSS*, 2021.

[51] M. FISCHLER AND, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, p. 381–395, June 1981.