2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Deep Masked Graph Matching for Correspondence Identification in Collaborative Perception

Peng Gao[1*], Qingzhao Zhu[2*], Hongsheng Lu[3], Chuang Gan[4], and Hao Zhang[5]

***Abstract*— Correspondence identification (CoID) is an essential component for collaborative perception in multi-robot systems, such as connected autonomous vehicles. The goal of CoID is to identify the correspondence of objects observed by multiple robots in their own field of view in order for robots to consistently refer to the same objects. CoID is challenging due to perceptual aliasing, object non-covisibility, and noisy sensing. In this paper, we introduce a novel deep masked graph matching approach to enable CoID and address the challenges. Our approach formulates CoID as a graph matching problem and we design a masked neural network to integrate the multimodal visual, spatial, and GPS information to perform CoID. In addition, we design a new technique to explicitly address object non-covisibility caused by occlusion and the vehicle's limited field of view. We evaluate our approach in a variety of street environments using a high-fidelity simulation that integrates the CARLA and SUMO simulators. The experimental results show that our approach outperforms the previous approaches and achieves state-of-the-art CoID performance in connected autonomous driving applications. Our work is available at: `https://github.com/gaopeng5/DMGM.git`.**

I. Introduction

Multi-robot systems have been widely investigated over the past decades due to their reliability and efficiency to address collaborative tasks, such as collaborative manufacturing [1], [2], multi-robot-assisted search and rescue [3], [4], and connected autonomous driving [5], [6]. To enable effective multi-robot collaboration, collaborative perception is a fundamental capability for multiple robots to share their perceptual data of the surrounding environment and to build a shared situational awareness among the robots.

As a critical component of collaborative perception, correspondence identification (CoID) aims to find the correspondence of the same objects observed by multiple robots in their own field of view. Figure 1 demonstrates an example of CoID. When a pair of connected vehicles meet at an intersection, they have to identify the correspondence of the street objects in order to correctly refer to the same objects when they share information about the objects.

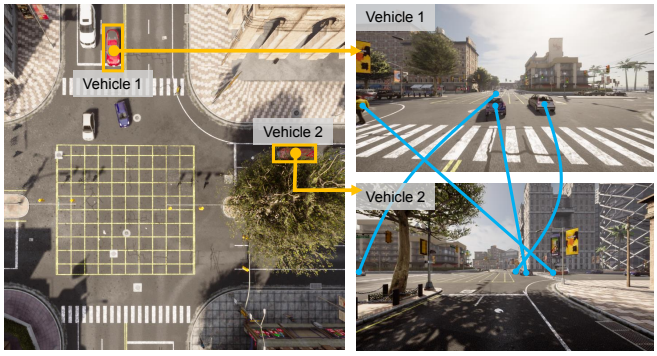


Fig. 1. A motivating scenario of correspondence identification to enable collaborative perception in connected autonomous driving. When two connected vehicles meet at an intersection, before they can share information of street objects, they need to identify the correspondence of the objects observed in their own field of view in order to consistently refer to the same objects.

Given the importance of CoID, a variety of approaches are developed, including learning-free and learning-based methods. Learning-free methods can be divided into three groups, including keypoint-based visual association [7], geometric-based spatial matching [8], [9], and synchronization methods that perform multi-view data association [10], [11]. Learning-based approaches are mainly based on deep learning, such as convolution neural network to perform object re-identification from different perspectives [12], [13] and graph neural network to perform deep graph matching [14], [15].

CoID is a challenging problem due to several reasons. First, enabling CoID must address the challenge of perceptual aliasing, i.e., street objects having similar or identical appearances, which often introduce visual ambiguity. The second challenge is caused by non-covisible objects that are only observed by a single robot due to occlusion or the robot's limited field of view. The third challenge is caused by noisy perception. For example, noisy depth sensing often causes inaccurate distance estimation.

To address the above challenges, we propose a novel deep masked graph matching method to perform CoID. We develop graph representations to encode visual-spatial information of street objects observed by each vehicle. Each node in a graph represents a street object and each edge represents the spatial relationship of a pair of objects. Given the graph representations of observations obtained by a pair of connected vehicles, we mathematically formulate CoID as a graph matching problem, and we develop a new masked graph neural network that integrates visual, spatial, and GPS cues to explicitly address object non-covisibility.

Authors with $*$ contributed equally to this paper.
[1]Peng Gao is with the Department of Computer Science, University of Maryland, College Park, MD 20742, USA. Email: gaopeng@umd.edu.
[2]Qingzhao Zhu is with the Department of Computer Science, Colorado School of Mines, Golden, CO 80401, USA. Email: zhuqingzhao@mines.edu.
[3]Hongsheng Lu are with Toyota Motor North America, Mountain View, CA 94043, USA. Email: hongsheng.lu@toyota.com.
[4]Chuang Gan is with MIT-IBM Watson AI Lab, Cambridge, MA 02142, USA. Email: ganchuang@csail.mit.edu.
[5]Hao Zhang is with the Manning College of Information and Computer Sciences (CICS), University of Massachusetts Amherst, Amherst, MA 01002, USA. Email: hao.zhang@cs.umass.edu.

The key contribution of this paper is the introduction of the masked deep graph matching method for CoID in collaborative perception. Specific novelties include

- We propose a multi-modal graph-based formulation that is able to integrate visual, spatial and GPS information to better represent street objects to improve CoID.
- We introduce a masked deep neural network with a novel loss design and non-covisible remover based on SoftMax variance thresholding to address object non-covisibility and improve CoID precision.

II. Related Work

A. Collaborative Perception

Given the popularity of multi-robot systems, collaborative perception attracts attention in recent studies. One of the well-known uses of collaborative perception is inter-robot loop closure detection in collaborative simultaneous localization and mapping (CSLAM), in which multiple robots need to recognize the same place by identifying the correspondences of landmarks or key points in order to merge their local maps [7], [16]. In addition, collaborative object localization can achieve better performance compared with single-view localization, in which multiple robots require to consistently localize the same objects given their own observations by identifying the correspondences of objects [17], [18], [19]. Furthermore, by associating multi-robot observations, collaborative perception also happens in trajectory forecasting [20], scene segmentation [21], [22], tracking and object detection [4]. In these applications, CoID plays an important role in collaborative perception, with the goal of associating key points or objects in various observations provided by multiple robots. However, most of the existing work assumes that the correspondences of objects are known or can be easily obtained via coordination transformation, such as based on GPS [21], [22] or point cloud registration [23]. The real-world cases without accurate GPS information or accurate poses have not been well studied yet to address CoID in collaborative perception.

In addition, there exist several open-source datasets on collaborative perception. CoMap uses CARLA [24] and SUMO [25] to generate a large-scale dataset for LiDAR-based object detection and semantic segmentation [26]. Similar datasets include OPV2V [27] and DAIRV2X [28], which use LiDAR points to evaluate object detection. V2X-Sim is a multi-modal multi-task dataset, which is used to evaluate object detection, segmentation and tracking [29]. Although these datasets provide ground truth for various collaborative perception tasks, they do not provide correspondences of street objects in multi-perspective observations obtained by connected vehicles. Our dataset provides the CoID ground truth that is computed from instance-level semantic segmentation in the simulations.

B. Correspondence Identification

CoID can be generally grouped into two categories, including learning-free and learning-based approaches.

Learning-free approaches can further be divided into three subgroups using visual appearance features, spatial relationship, and synchronization algorithms, respectively. Visual appearance features are commonly used for key-point matching to register adjacent frames or local-global mapping, such as SIFT [30] and ORB [31]. In addition, region-based visual features, such as HOG [32] and TransReID [33], are generally used to identify the same place observed at different times. Besides using visual features, spatial features are also used to identify correspondences of objects, such as ICP [34], graph matching [9], [8] and maximum clique [35]. Furthermore, synchronization algorithms are also highly related to the problem of CoID [10], [11]. These algorithms take the pairwise correspondences as inputs, and output multi-view correspondences by forcing circle consistency, e.g., using graph cut [10] and convex optimization [36].

Learning-based approaches typically focus on using deep neural networks to perform CoID. Specifically, these methods are typically based on convolution neural networks (CNN) or graph neural networks (GNN). CNN-based methods focus on extracting high-level visual features to recognize the same objects observed from different perspectives [37], [38], [39]. GNN-based approaches aim to learn a unique pattern surrounding objects by aggregating their spatial relationships [14], [40], [41]. In addition to the pure CNN or GNN-based approaches, there are several methods using the combination of them [1], [15]. By integrating visual-spatial information for CoID, it can improve robustness.

Although existing methods show promising performance, they often cannot fuse multi-modal information (visual, spatial, and GPS cues) for CoID. The problem of non-covisibility has also not been well addressed yet to enable CoID in connected driving scenarios.

III. Approach

Notation. Matrices are denoted as boldface capital letters, e.g., $\mathbf{M}=\{\mathbf{M}_{i,j}\} \in \mathcal{R}^{n\times m}$. $\mathbf{M}_{i,j}$ denotes the element in the i-th row and the j-th column of $\mathbf{M}$, and $\mathbf{M}_{i,:}$ denotes the i-th row of $\mathbf{M}$. Vectors are denoted as boldface lowercase letters $\mathbf{v} \in \mathcal{R}^n$, and scalars are denoted as lowercase letters.

A. Problem Formulation

Given an observation observed by a vehicle, we represent it as a graph $\mathcal{G}(\mathcal{V}, \mathbf{X}, \mathbf{A})$. $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ represents the node set with $\mathbf{v}_i \in \mathcal{V}$ denoting the 3D position of the i-th detected object (e.g. vehicle or pedestrian). Each object is associated with a visual feature vector, denoted as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the visual feature vector of the i-th object, and d is the vector length. $\mathbf{A}$ is the adjacent matrix indicating the node connection. All nodes are connected via Delaunay triangulation. If node i and node j are connected, then $\mathbf{A}_{i,j} = ||\mathbf{v}_i - \mathbf{v}_j||^2$; otherwise, $\mathbf{A}_{i,j} = 0$.

In collaborative perception, observations observed by a pair of vehicles can be represented as $\mathcal{G}$ and $\mathcal{G}'$, respectively. The correspondences of the objects observed by the two vehicles are represented by the correspondence matrix $\mathbf{Y} \in \mathcal{R}^{n\times n'}$. Then, we mathematically formulate CoID as a graph matching problem, which is defined as follows:

$$\arg\max_{\mathbf{Y}} \mathbf{S}^\top \mathbf{Y} \quad \text{s.t. } \mathbf{Y}\mathbf{1}_{n'\times 1} \leq \mathbf{1}_{n\times 1}, \mathbf{Y}^\top \mathbf{1}_{n\times 1} \leq \mathbf{1}_{n'\times 1} \tag{1}$$

where $\mathbf{S}$ encodes the similarity between $\mathcal{G}$ and $\mathcal{G}'$ and $\mathbf{1}$ is a all-one vector. Given Eq. (1), $\mathbf{Y}$ is optimal when the similarity $\mathbf{S}$ is maximum according to the identified correspondences. The constraint is used to force the one-to-one correspondences. In other words, an object in one observation can at most have one corresponding object in the other observation. The main technical problem we need to address is how to calculate the similarity $\mathbf{S}$ between a pair of graphs $\mathcal{G}$ and $\mathcal{G}'$.

B. Attentional Graph Embedding

To calculate the similarity between a pair of graphs, we do not just encode each object's own visual features but also its surrounding objects' appearance features given their spatial relationships. Formally, we use an attentional graph neural network to compute node embedding vectors as $\mathbf{H} = \Psi(\mathbf{X}, \mathbf{A})$, where $\mathbf{H}$ denotes the embedding matrix and $\Psi(\mathbf{X}, \mathbf{A})$ denotes the attentional graph neural network. Each row of $\mathbf{H}_{i,:}$ represents an embedding vector of the i-th object, which is defined as $\mathbf{h}_i$.

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

where $l = 1, 2, \ldots, L$ denotes the layer index, $\mathbf{q}_i^l$, $\mathbf{k}_i^l$ and $\mathbf{v}_i^l$ denote query, key and value, $\mathbf{W}_q^l, \mathbf{W}_k^l, \mathbf{W}_v^l$ denote their associating trainable weights. $\mathbf{h}_i^l$ denotes the feature embedding vector of the i-th object at the l-th layer, where $\mathbf{h}_i^0 = \mathbf{x}_i^0$. In the self-attention mechanism, the object feature $\mathbf{x}_i^l \in \mathbf{X}$ is first linearly transformed to query, key and value. Then the self-attention is computed as follows:

$$\alpha_{i,j}^l = \text{SoftMax}\left(\frac{(\mathbf{q}_i^l)^\top (\mathbf{k}_j^l + \mathbf{W}_e^l \mathbf{A}_{i,j})}{\sqrt{c^l}}\right) \tag{3}$$

where $\alpha_{i,j}^l$ is the attention from node j to node i at layer l. To encode spatial relationships of objects, we add edge attributes into the learning process, where W_e^l denotes the learnable parameter matrix that has the same dimension of key k_j^l. c^l is the number of output channels. This attention weight is obtained by comparing the query of the i-th node with its neighborhood keys and edge attributes. The final attention is normalized by the SoftMax function. The final node embedding vector is computed as follows:

$$\mathbf{h}_i^{l+1} = \mathbf{W}_x^l \mathbf{h}_i^l + \sum_{\mathbf{A}_{i,j}=1} \alpha_{i,j}^l (\mathbf{v}_j^l + \mathbf{W}_e^l \mathbf{A}_{i,j}) \tag{4}$$

$\mathbf{W}_x^l$ is a learnable parameter matrix. The final embedding vector $\mathbf{h}_i$ is computed via aggregating the object embedding feature and its neighborhood edge attributes weighted by attention weights. We also use a multi-head mechanism [42] to enable the network to catch a richer representation of the embedding. Multi-head embedding vectors are concatenated after intermediate attention layers and averaged after the last attention layer. Given the object embedding vectors, we compute the similarity between two graphs $\mathcal{G}$ and $\mathcal{G}'$ as follows:

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

where $\mathbf{S} \in \mathbb{R}^{m \times n}$ represents the similarity between the two graphs with m and n objects respectively.

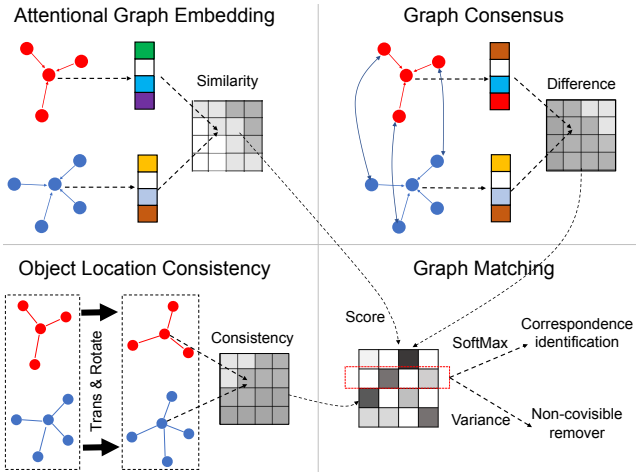


Fig. 2. Overview of the proposed masked deep graph matching approach for correspondence identification.

C. Graph Consensus

Due to the existence of noise in sensing information, such as noise in depth or RGB observations, it will introduce ambiguity into the CoID process based on visual and spatial information. Thus, we study the graph pruning technique to improve the robustness of CoID from two aspects. Inspired by the most recent work [41], we apply the graph consensus principle to the graphs according to the identified correspondences of objects. Formally,

$$\mathbf{S}_{i,j} = \mathbf{h}_i \mathbf{h}_j^\top + \varphi(\mathbf{D}_{i,j}) \tag{6}$$

where φ denotes a multi-layer perceptron with two linear layers followed by a ReLu non-linear activation function. $\mathbf{D}$ denotes the consensus difference between graphs $\mathcal{G}$ and $\mathcal{G}'$ given the correspondences $\mathbf{Y}$. Formally,

$$\mathbf{D} = (\mathbf{S}^\top \Psi(\mathbf{U}, \mathbf{A}) - \Psi(\mathbf{S}^\top \mathbf{U}, \mathbf{A}'))^\top \tag{7}$$

where $\mathbf{U} \in \mathcal{R}^{n \times r}$ is a random matrix with each row $\mathbf{U}_{i:}$ denoting a random feature vector with length r. When $\mathcal{G}$ and $\mathcal{G}'$ indicate the same graph (isomorphism), then $\mathbf{S}^\top \Psi(\mathbf{U}, \mathbf{A}) = \Psi(\mathbf{S}^\top \mathbf{U}, \mathbf{S}^\top \mathbf{A}\mathbf{S}) = \Psi(\mathbf{S}^\top \mathbf{U}, \mathbf{A}')$. In this case, $\mathbf{D}_{i,j} = \mathbf{0}$. Otherwise, $\varphi(\mathbf{D})$ that indicates the differences between two graphs will update the similarity matrix $\mathbf{S}$.

D. Object Location Consistency

In addition to pruning graph matching results given the consensus principle, our approach also incorporates GPS information to improve graph matching performance. We assume that the GPS information of two connected vehicles can be represented as the extrinsic parameters of cameras mounted on the vehicles with respect to the global coordinate. Then we can transfer the objects' 3D positions from the camera coordinates to the GPS global coordinates given the extrinsic parameters. We represent the positions of objects in the world 3D coordinates as $\{\mathbf{p}_i\}^n$ and $\{\mathbf{p}'_j\}^{n'}$ separately. Given the consistency of object positions in the world coordinates, we construct a position mask $\mathbf{G} \in \mathbb{R}^{n \times n'}$ based on the distances

Fig. 3. The five intersection scenarios that are implemented in our CAD simulator and used for approach evaluation in the experiments.

of corresponding objects in two different graphs. Each element in $\mathbf{G}$ can be calculated as:

$$\mathbf{G}_{i,j} = \frac{1}{\mathbf{p}_i - \mathbf{p}'_j} \tag{8}$$

where $\mathbf{G}$ is calculated via a reciprocal function, which represents the similarity between the positions of the corresponding objects in the world coordinates. If the corresponding objects denote the same object, then $\mathbf{G}_{i,j}$ will be large as their world coordinates are ideally the same. The final similarity score $\mathbf{S} \in \mathbb{R}^{n \times n'}$ is defined as follows:

$$\mathbf{S}_{i,j} = \mathbf{h}_i \mathbf{h}_j^\top + \varphi(\mathbf{D}_{i,j}) + \mathbf{G}_{i,j} \tag{9}$$

where the similarity score $\mathbf{S}$ containing similarities of visual-spatial features of objects, neighborhood structure consensus, and GPS-based object position consistency, as shown in Figure 2. Finally, the correspondences of objects can be identified as follows:

$$\mathbf{Y} = \text{SoftMax}(\mathbf{S}) \tag{10}$$

Even though SoftMax can not enforce one-to-one constraints on the correspondences, it has better consistency with the consensus pruning [41] and has similar performance as the one-to-one constrained assignment problem solver, such as Sinkhorn normalization.

E. Addressing Non-Covisible Objects

There may exist many non-covisible objects in two observations. To address this challenge, we propose a threshold-based approach to remove potential non-covisible objects based on SoftMax variance. Specifically, we do thresholding on the standard deviation of each row of $\mathbf{S}$, which is defined as $s_i = \sigma(\mathbf{S}_{i,:})$, where σ denotes the variance operator and s_i indicates the variance of the i-th identified correspondence. If the network is confident in the classification result computed via SoftMax, s_i should be large, otherwise; it should be small (as all the categories have similar confidence in the classification results). Then, we do thresholding on the variance. If $s_i \geq \theta$, then we preserve the i-th object in the correspondence matrix. If $s_i < \theta$, then we remove it from the identified correspondences by setting $\mathbf{Y}_{i,:} = 0$. Given the threshold on the SoftMax variance, we can significantly remove the non-covisible object from the identified correspondences, as they usually have low variance (even confidence) in the classification results. To train our network, we design the loss function as follows:

$$\mathcal{L} = \frac{1}{NN'} \sum_{i,j} (\mathbf{S}_{i,j} - \mathbf{Y}^*_{i,j})^2 \tag{11}$$

where $\mathbf{Y}^* \in \mathcal{R}^{n \times n'}$ denotes the ground truth correspondence. $\mathbf{Y}_{i,j} = 1$ denotes that the i-th object in one observation corresponds to the j-th object in the other observation. Otherwise, $\mathbf{Y}_{i,j} = 0$. The correspondence matrix $\mathbf{Y} \in \mathcal{R}^{m \times n}$ is optimal when the loss function is minimized.

IV. Experiments

In this section, we discuss our experimental setup, results in high-fidelity CAD simulations, and analysis of our approach.

A. Experimental Setup

We implement a high-fidelity connected autonomous driving (CAD) simulator that integrates CARLA [43] and SUMO [25]. CARLA is an open-source autonomous driving simulator that is able to simulate vehicle sensors, driving control and traffic scenarios. In our experiments, we design five different traffic scenarios at street interactions where connected vehicles more frequently meet from different driving directions. These five simulated scenarios are depicted in Figure 3.

In the simulations, each connected vehicle is equipped with a front-facing RGB camera, a front-facing depth camera, and a global navigation satellite system (GNSS) sensor. Examples of the RGB and depth images from a pair of connected vehicles are shown in Figure 4. Simulation of the GNSS sensor follows the technical specification of the real SBG Ellipse2-D sensor. Traffic patterns, including pedestrians and vehicles, are controlled by SUMO. Behaviors of vehicles and pedestrians are generated randomly and follow real-world rules, such as stopping at the red light and yielding to the pedestrian.

TABLE I

Description of our CAD dataset based on CARLA and SUMO simulations for CoID evaluation.

# Instances	69,469 from 5 different scenarios
Sensor	Color, depth, and GNSS sensors
RGBD specs	1920×1080 at 10 FPS
GNSS noise	$\mathcal{N}(0, 1.2m)$ in vertical and horizontal directions $\mathcal{N}(0, 0.2°)$ in yaw

Using our CAD simulator, we collect a large CAD dataset that is summarized in Table I. We collect a total of 69,469 data instances, of which 60,260 data instances are used for training,

3,747 data instances are used for validation, and 5,462 data instances are used for testing. Each data instance includes a pair of RGBD images observed by two connected vehicles from different perspectives, the GNSS positions and orientations of vehicles, and the ground truth of object correspondences directly obtained from the instance-level segmentation provided by CARLA (each object segmentation has a unique ID).

Fig. 4. Examples of the simulated color and depth images obtained by two connected vehicles at the same intersection from different perspectives.

In our graph construction, we use YOLOv5 [44] to detect objects and we extract the HOG feature [32] as each node's appearance feature. The edges are connected via Delaunay triangulation. The edge attributes (distances) are calculated from pairs of objects' positions, which are obtained from the depth images. The GNSS positions and orientations of each vehicle are represented as the XYZ-pitch-roll-yaw form, which can be rewritten as the transformation matrix.

In the implementation of our network, the attentional GNN Ψ is implemented based on the PyTorch geometric library. We set the number of network layers to be $L = 2$. In the first network layer, we set $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{900\times\{heads*256\}}$ where the multi-head number $heads = 4$. In addition, we set $\mathbf{W}_e \in \mathbb{R}^{dim\times\{heads*256\}}$ where the edge feature dimension $dim = 1$. Each attentional layer is followed by dropout with probability 0.5. For the MLP φ with two linear layers, each layer is followed by dropout with probability 0.2. In all the experiments, we use ADMM as the optimization method. We run 60 epochs to train our approach.

For comparison, we first implement a baseline line method that is our full approach but without using GPS information to generate an object location consistency matrix as defined in Eq. (8). In addition, we compare our method with three existing methods as follows:

- Graph convolutional neural network for graph matching (**GCN-GM**) [45] that use spline kernel to aggregate objects' themselves and their neighborhood visual-spatial information for graph matching.
- Deep graph matching consensus (**DGMC**) [46] that performs an iterative refinement process on the similarity matrix given consensus principle.
- Bayesian deep graph matching (**BDGM**) [1] that performs deep graph matching under Bayesian framework and reduces non-covisible objects based on correspondences uncertainties.

None of the comparison methods are capable of integrating GPS information and only ours and BDGM explicitly address non-covisible objects in CoID.

As we treat the CoID as a data retrieval process, we use the following metrics to evaluate the CoID performance.

- **Precision** is defined as the ratio of the retrieved correct correspondences over all the retrieved correspondences.
- **Recall** is defined as the ratio of the retrieved correct correspondences over the ground truth correspondences.
- **F1 Score** is a metric to evaluate the overall performance of CoID methods, which is defined as $(2 \times Precision \times Recall)/(Precision + Recall)$.

B. Results over Connected Autonomous Driving Simulations

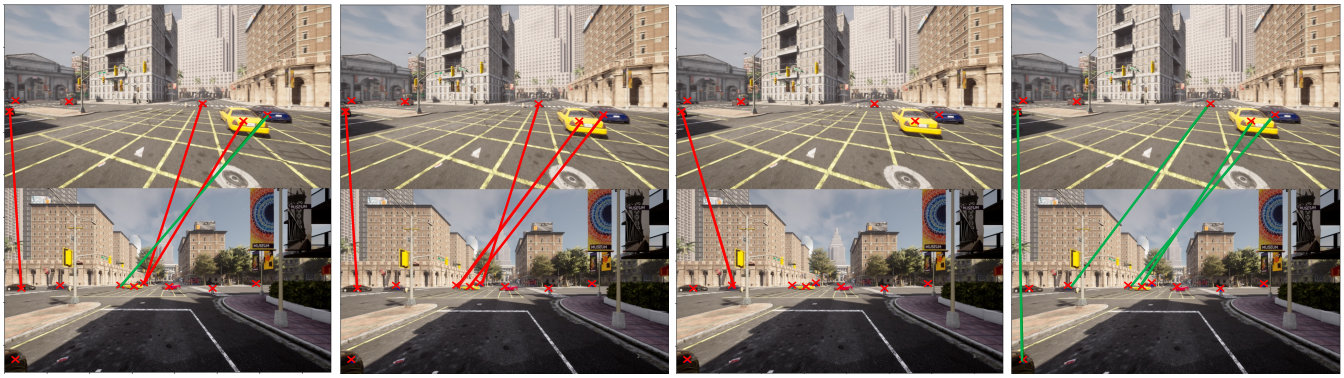
The CAD simulation includes a variety of technical challenges to perform CoID, including various street objects (e.g., pedestrians and vehicles) with ambiguous visual appearance, a large number of non-covisible objects, strong occlusion in the perception, noisy observation caused long-distance observing, as well as the realistic noisy GPS information, which follows the specification of the real-world GNSS. We run our approach on a Linux machine with an i7 16-core CPU and 16G memory. The average execution time is around 20Hz.

TABLE II
QUANTITATIVE RESULTS OF OUR APPROACH AND COMPARISONS WITH THREE PREVIOUS METHODS OVER THE CAD SIMULATIONS.

Method	Precision	Recall	F1
GCN-GM [45]	0.5001	0.6391	0.5611
DGMC [41]	0.4736	0.6425	0.5453
BDGM [1]	0.6817	0.6097	0.6437
Ours w/o GPS	0.7464	0.8006	0.7726
Ours	**0.7859**	**0.8278**	**0.8063**

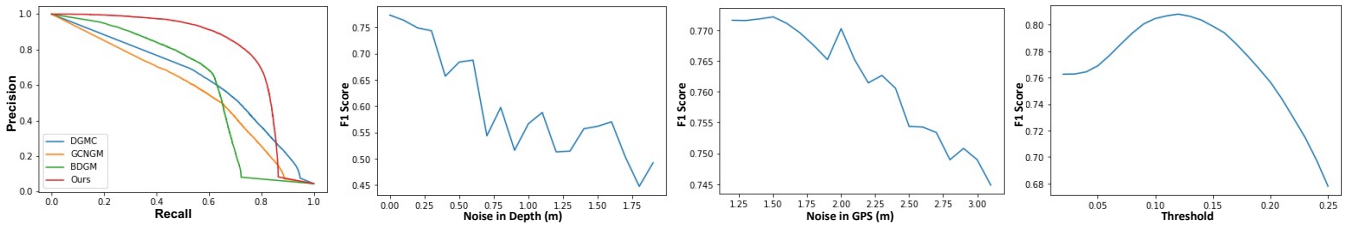
The quantitative results are presented in Table II. We can see that our baseline approach outperforms all the previous methods on precision, recall, and F1 score due to its capability of addressing non-covisible objects and integrating visual-spatial information for CoID. In addition, our full approach outperforms the baseline method, which indicates the importance of integrating GPS information. In the comparison with previous approaches, GCN-GM performs the worst as it is generally only focusing on integrating visual-spatial features of objects, and does not consider graph pruning and non-covisible object elimination. DGMC performs better than GCN-GM, as it is the first to propose graph matching consensus principle, the final identified correspondences can be refined by updating the consensus. BDGM achieves promising results on precision due to its consideration of removing non-covisible objects given correspondence uncertainties. By efficiently addressing visual ambiguity and non-covisible objects, as well as integrating GPS into graph matching, our approach achieves the best CoID performance.

The qualitative results of our approach are shown in Figure 5. We can clearly see that our approach correctly identified the correspondences of objects observed by connected vehicles. By comparing with other methods, we can also see that GCN-GM and DGMC perform badly as they aim to maximize the number of correspondences and ignore the influence caused by non-covisible objects. BDGM also performs badly as it removes most of the uncertain correspondences that are caused by sensing noise and low-resolution observations. By integrating multi-modal sensing information and addressing non-covisible objects, our approach can identify correspondences robustly in the CAD scenario.



(a) GCN-GM (b) DGMC (c) BDGM (d) Ours

Fig. 5. Qualitative experimental results obtained by our approach in the CAD simulations, and comparisons with GCN-GM, DGMC and BDGM. Green lines denote correct correspondences and red lines denote incorrect correspondences. Red cross symbols denote the detected street objects in different views of the vehicles. [Best viewed in color.]



(a) Precision-Recall Curve (b) Noisy Depth (c) Noisy GPS (d) Varying Thresholds

Fig. 6. Discussion of our CoID approach's characteristics, including the precision-recall curve, effects of noise in depth and GPS on the performance, and the analysis of threshold values.

C. Discussion

We further study our approach's characteristics, including the overall performance based on the precision-recall curve, robustness to the depth noise and the GPS noise, as well as the analysis of hyper-parameter threshold θ.

1) Overall Performance: In order to evaluate the overall performance of CoID, we draw the precision-recall curve as shown in Figure 6(a). We also use a single-value evaluation metric of Area Under the Curve (AUC) to evaluate the overall performance, which is defined as the area under the precision-recall curve. Its value is between [0, 1] with a greater value indicating better performance. It is observed that our approach obtains the AUC of 0.79, which is significantly larger than the BDGM with 0.6 and DGMC with 0.64.

2) Robustness to Noise: Figure 6(b) demonstrates the effect of varying noise in the sensing depth. We can see that as the increase of depth noise, the performance of our approach gradually decreases with small fluctuation. We can also see that our method obtains robust performance with $0.5m$ depth sensing noise. Figure 6(c) illustrates the effect of varying noise in the GPS. We can see that our approach generally shows good robustness to the noise in the GPS information. The GPS noise increasing from $1.2m$ to $3m$ cause $\%2.5$ decrease in the overall performance.

3) Threshold Analysis: We use hyperparameter θ to threshold the identified correspondences based upon the SoftMax variance in order to remove non-covisible objects. Figure 6(d) shows the sensitivity analysis of the performance influenced by θ based on the F1 score. We observe that our approach achieves the best performance when $\theta = 0.13$.

V. Conclusion

Correspondence identification is essential for collaborative perception in connected autonomous driving, with the goal of enabling consistent reference of street objects by connected vehicles. To address the key technical challenges, including perceptual aliasing, non-covisibility, and noisy perception, we introduce a novel deep masked graph matching approach for CoID. Through integrating multi-modal sensing information (including visual, spatial and GPS cues), our approach is able to robustly identify the correspondences of street objects. In addition, we implement a new technique to remove non-covisible objects by thresholding the SoftMax variance. Finally, we implement a connected autonomous driving simulator by integrating CARLA and SUMO, and employ it to collect a large-scale dataset that includes around 70K pairs of high-fidelity paired observations with ground truth correspondences for the training and evaluation of CoID methods. The experimental results show our approach outperforms previous methods and achieves state-of-the-art CoID performance in connected autonomous driving applications.

REFERENCES

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

[2] S. Zhang, Y. Chen, J. Zhang, and Y. Jia, "Real-time adaptive assembly scheduling in human-multi-robot collaboration according to human capability," in *ICRA*, 2020.

[3] B. Reily, J. G. Rogers, and C. Reardon, "Balancing mission and comprehensibility in multi-robot systems for disaster response," in *SSRR*, 2021.

[4] C. Robin and S. Lacroix, "Multi-robot target detection and tracking: taxonomy and survey," *AuRo*, vol. 40, no. 4, pp. 729–760, 2016.

[5] R. Guo, H. Lu, P. Gao, Z. Zhang, and H. Zhang, "Collaborative localization for occluded objects in connected vehicular platform," in *VTC*, 2019.

[6] S. Wei, D. Yu, C. L. Guo, L. Dan, and W. W. Shu, "Survey of connected automated vehicle perception mode: from autonomy to interaction," *ITS*, vol. 13, no. 3, pp. 495–505, 2018.

[7] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *CVPR*, 2016.

[8] P. Gao, R. Guo, H. Lu, and H. Z. Zhang, "Regularized graph matching for correspondence identification under uncertainty in collaborative perception," in *RSS*, 2021.

[9] P. Gao, Z. Zhang, R. Guo, H. Lu, and H. Zhang, "Correspondence identification in collaborative robot perception through maximin hypergraph matching," in *ICRA*, 2020.

[10] K. Fathian, K. Khosoussi, Y. Tian, P. Lusk, and J. P. How, "CLEAR: A consistent lifting, embedding, and alignment rectification algorithm for multiview data association," *TRO*, vol. 36, no. 6, pp. 1686–1703, 2020.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[12] H.-X. Yu, A. Wu, and W.-S. Zheng, "Unsupervised person re-identification by deep asymmetric metric embedding," *TPAMI*, 2018.

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

[14] R. Wang, J. Yan, and X. Yang, "Learning combinatorial embedding networks for deep graph matching," in *ICCV*, 2019.

[15] P. Gao, R. Guo, H. Lu, and H. Zhang, "Correspondence identification for collaborative multi-robot perception under uncertainty," *AuRo*, vol. 46, no. 1, pp. 5–20, 2022.

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

[23] J. Zhang and S. Singh, "LOAM: lidar odometry and mapping in real-time." in *RSS*, 2014.

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

[25] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "SUMO (simulation of urban mobility)-an open-source traffic simulation," in *The 4th middle East Symposium on Simulation and Modelling*, 2002.

[26] Y. Yuan and M. Sester, "COMAP: A synthetic dataset for collective multi-agent perception of autonomous driving," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 255–263, 2021.

[27] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, "OPV2V: an open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *ICRA*, 2022.

[28] H. Yu, Y. Luo, M. Shu, Y. Huo, Z. Yang, Y. Shi, Z. Guo, H. Li, X. Hu, J. Yuan, *et al.*, "DAIR-V2X: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection," in *CVPR*, 2022.

[29] Y. Li, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng, "V2X-Sim: A virtual collaborative perception dataset for autonomous driving," *RAL*, 2022.

[30] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *ECCV*, 2014.

[31] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *TRO*, vol. 31, no. 5, pp. 1147–1163, 2015.

[32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[33] S. He, H. Luo, P. Wang, F. Wang, H. Li, and W. Jiang, "Transreid: Transformer-based object re-identification," in *ICCV*, 2021.

[34] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *IEEE the 3rd International Conference on 3-D Digital Imaging and Modeling*, 2001.

[35] K. Adamczewski, Y. Suh, and K. Mu Lee, "Discrete tabu search for graph matching," in *ICCV*, 2015.

[36] N. Hu, Q. Huang, B. Thibert, and L. J. Guibas, "Distributable consistent multi-object matching," in *CVPR*, 2018.

[37] X. Jin, C. Lan, W. Zeng, G. Wei, and Z. Chen, "Semantics-aligned representation learning for person re-identification," in *AAAI*, 2020.

[38] A. Khatun, S. Denman, S. Sridharan, and C. Fookes, "Semantic consistency and identity mapping multi-component generative adversarial network for person re-identification," in *WACV*, 2020.

[39] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "MOTS: Multi-object tracking and segmentation," in *CVPR*, 2019.

[40] Z. Zhang and W. S. Lee, "Deep graphical feature learning for the feature matching problem," in *ICCV*, 2019.

[41] M. Fey, J. E. Lenssen, C. Morris, J. Masci, and N. M. Kriege, "Deep Graph Matching Consensus," in *ICLR*, 2019.

[42] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017.

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

[44] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, TaoXie, J. Fang, imyhxy, K. Michael, Lorna, A. V, D. Montes, J. Nadar, Laughing, tkianai, yxNONG, P. Skalski, Z. Wang, A. Hogan, C. Fati, L. Mammana, AlexWang1900, D. Patel, D. Yiwei, F. You, J. Hajek, L. Diaconu, and M. T. Minh, "ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference," Feb. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6222936

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

[46] M. Fey, J. E. Lenssen, C. Morris, J. Masci, and N. M. Kriege, "Deep graph matching consensus," in *ICLR*, 2020.