Abstract

To promote better performance-bandwidth trade-off for multi-agent perception, we propose a novel distilled collaboration graph (DiscoGraph) to model trainable, pose-aware, and adaptive collaboration among agents. Our key novelties lie in two aspects. First, we propose a teacher-student framework to train DiscoGraph via knowledge distillation. The teacher model employs an early collaboration with holistic-view inputs; the student model is based on intermediate collaboration with single-view inputs. Our framework trains DiscoGraph by constraining post-collaboration feature maps in the student model to match the correspondences in the teacher model. Second, we propose a matrix-valued edge weight in DiscoGraph. In such a matrix, each element reflects the inter-agent attention at a specific spatial region, allowing an agent to adaptively highlight the informative regions. During inference, we only need to use the student model named as the distilled collaboration network (DiscoNet). Attributed to the teacher-student framework, multiple agents with the shared DiscoNet could collaboratively approach the performance of a hypothetical teacher model with a holistic view. Our approach is validated on V2X-Sim, a large-scale multi-agent perception dataset that we synthesized using CARLA and SUMO co-simulation. Our quantitative and qualitative experiments in multi-agent 3D object detection show that DiscoNet could not only achieve a better performance-bandwidth trade-off than the state-of-the-art collaborative perception methods, but also bring more straightforward design rationale. Our code is available on https://github.com/ai4ce/DiscoNet.

1 Introduction

Perception, which involves organizing, identifying and interpreting sensory information, is a crucial ability for intelligent agents to understand the environment. Single-agent perception [4] has been studied extensively in recent years, e.g., 2D/3D object detection [18, 27], tracking [23, 22] and segmentation [25, 16], etc. Despite its great progress, single-agent perception suffers from a number of shortcomings stemmed from its individual perspective. For example, in autonomous driving [8], the LiDAR-based perception system can hardly perceive the target in the occluded or long-range areas. Intuitively, with an appropriate collaboration strategy, multi-agent perception could fundamentally upgrade the perception ability over single-agent perception.

To design a collaboration strategy, current approaches mainly include raw-measurement-based early collaboration, output-based late collaboration and feature-based intermediate collaboration.

*Corresponding authors.

Early collaboration [3] aggregates the raw measurements from all the agents, promoting a holistic perspective; see Fig. 1 (a). It can fundamentally solve the occlusion and long-range issues occurring in the single-agent perception; however, it requires a lot of communication bandwidth. Contrarily, late collaboration aggregates each agent’s perception outputs. Although it is bandwidth-efficient, each individual perception output could be noisy and incomplete, causing unsatisfying fusion results. To deal with the performance-bandwidth trade-off, intermediate collaboration [19, 34, 20] has been proposed to aggregate intermediate features across agents; see Fig. 1 (b). Since we can squeeze representative information to compact features, this approach can potentially both achieve communication bandwidth efficiency and upgrade perception ability; however, a bad design of collaboration strategy might cause information loss during feature abstraction and fusion, leading to limited improvement of the perception ability.

To achieve an effective design of intermediate collaboration, we propose a distilled collaboration graph (DiscoGraph) to model the collaboration among agents. In DiscoGraph, each node is an agent with real-time pose information and each edge reflects the pair-wise collaboration between two agents. The proposed DiscoGraph is trainable, pose-aware, and adaptive to real-time measurements, reflecting dynamic collaboration among agents. It is novel from two aspects. First, from the training aspect, we propose a teacher-student framework to train DiscoGraph through knowledge distillation [1, 10, 26]; see Fig. 1 (c). Here the teacher model is based on early collaboration with holistic-view inputs and the student model is based on intermediate collaboration with single-view inputs. The knowledge-distillation-based framework enhances the training of DiscoGraph by constraining the post-collaboration feature maps in the student model to match the correspondences in the teacher model. With the guidance of both output-level supervision from perception and feature-level supervision from knowledge distillation, the distilled collaboration graph promotes better feature abstraction and aggregation, improving the performance-bandwidth trade-off. Second, from the modeling aspect, we propose a matrix-valued edge weight in DiscoGraph to reflect the collaboration strength with a high spatial resolution. In the matrix, each element represents the inter-agent attention at a specific spatial region. This design allows the agents to adaptively highlight the informative regions and strategically select appropriate partners to request supplementary information.

During inference, we only need to use the student model. Since it leverages DiscoGraph as the key component, we call the student model as the distilled collaboration network (DiscoNet). Multiple agents with the shared DiscoNet could collaboratively approach the performance of a hypothetical teacher model with the holistic view.

To validate the proposed method, we build V2X-Sim 1.0, a new large-scale multi-agent 3D object detection dataset in autonomous driving scenarios based on CARLA and SUMO co-simulation platform [6]. Comprehensive experiments conducted in 3D object detection [37, 28, 29, 17] have shown that the proposed DiscoNet achieves better performance-bandwidth trade-off and lower communication latency than the state-of-the-art intermediate collaboration methods.
Figure 2: Overall teacher-student training framework. In the student model, for Agent 1 (in red), its input single-view point cloud can be converted to a BEV map, and then be consumed by a shared encoder $\Theta^s$ to obtain the feature map $F^s_i$. Based on the collaboration graph $G_{\Pi}$, Agent 1 aggregates the neural messages from other agents and obtains the updated feature map $H^s_i$. The shared header following the shared decoder outputs the 3D detection results. In the teacher model, we aggregate the points collected from all agents to obtain a holistic-view point cloud. We crop the region and align the pose to ensure the BEV maps in both teacher and student models are synchronized. We constrain all the post-collaboration feature maps in the student model to match the correspondences in the teacher model through knowledge distillation, resulting in a collaborative student model.

2 Multi-Agent Perception System with Distilled Collaboration Graph

This work considers a collaborative perception system to perceive a scene, where multiple agents can perceive and collaborate with each other through a broadcast communication channel. We assume each agent is provided with an accurate pose and the perceived measurements are well synchronized. Then, given a certain communication bandwidth, we aim to maximize the perception ability of each agent through optimizing a collaboration strategy. To design such a strategy, we consider a teacher-student training framework to integrate the strengths from both early and intermediate collaboration. During training, we leverage an early-collaboration model (teacher) to teach an intermediate-collaboration model (student) how to collaborate through knowledge distillation. Here the teacher model and the student model are shared by all the agents, but each agent would input raw measurements from its own view to either model; see Fig. 2. During inference, we only need the student model, where multiple agents with the shared student model could collaboratively approach the performance of the teacher model with the hypothetical holistic view.

In this work, we focus on the perception task of LiDAR-based 3D object detection because the unifying 3D space naturally allows the aggregation of multiple LiDAR scans. Note that in principle the proposed method is generally-applicable in collaborative perception if there exists a unified space to aggregate raw data across multiple agents.

2.1 Student: Intermediate collaboration via graphs

Feature encoder. The functionality is to extract informative features from raw measurements for each agent. Let $X_i$ be the 3D point cloud collected by the $i$th agent ($M$ agents in total), the feature of the $i$th agent is obtained as $F^s_i \leftarrow \Theta^s(X_i)$, where $\Theta^s(\cdot)$ is the feature encoder shared by all the agents and the superscript $s$ reflects the student mode. To implement the encoder, we convert a 3D point cloud to a bird’s-eye-view (BEV) map, which is amenable to classic 2D convolutions. Specifically, we quantize the 3D points into regular voxels and represent the 3D voxel lattice as a 2D pseudo-image, with the height dimension corresponding to image channels. Such a 2D image is virtually a BEV map, whose basic element is a cell that is associated with a binary vector along the vertical axis. Let $B_i \in \{0, 1\}^{K \times K \times C}$ be the BEV map of the $i$th agent associated with $X_i$. With this map, we can apply four blocks composed of 2D convolutions, batch normalization and ReLU activation to gradually reduce the spatial dimension and increase the number of channels for the BEV, to obtain the feature map $F^s_i \in \mathbb{R}^{K' \times K \times C'}$ with $K' \times K'$ the spatial resolution and $C'$ the number of channels.
where the transformation $i$ we leverage the collaboration graph to achieve feature transformation across agents. For the

**Remark**

agents share the same teacher model to guide one student model; however, each agent provides the pipeline has the feature encoder $\Theta$. During the training phase, an early-collaboration model, as the teacher, is introduced to guide the

2.2 Teacher: Early collaboration

the number of channels. Finally, we use an output header to generate the final detection outputs, the corresponding feature map in the encoder and then uses a

Decoder and header. After collaboration, each agent decodes the updated BEV-based feature map. The decoded feature map is $M^s_i \leftarrow \Psi^s(H^s_i)$. To implement the decoder $\Psi^s(\cdot)$, we progressively up-sample $H^s_i$ with four layers, where each layer first concatenates the previous feature map with the corresponding feature map in the encoder and then uses a $1 \times 1$ convolutional operation to halve the number of channels. Finally, we use an output header to generate the final detection outputs, $Y^s_i \leftarrow \Phi^s(M^s_i)$. To implement the header $\Phi^s(\cdot)$, we use two branches of convolutional layers to classify the foreground-background categories and regress the bounding boxes.

2.2 Teacher: Early collaboration

During the training phase, an early-collaboration model, as the teacher, is introduced to guide the intermediate-collaboration model, which is a student. Similar to the student model, the teacher’s pipeline has the feature encoder $\Theta^t$, feature decoder $\Psi^t$ and the output header $\Phi^t$. Note that all the agents share the same teacher model to guide one student model; however, each agent provides the inputs with its own pose, and its inputs to both the teacher and student models should be well aligned.
Feature encoder. Let $\mathcal{X} = \mathcal{A}(\xi_1 \odot \mathcal{X}_1, \xi_2 \odot \mathcal{X}_2, \ldots, \xi_M \odot \mathcal{X}_M)$ be a holistic-view 3D point cloud that aggregates all the points from all $M$ agents in the global coordinate system, where $\mathcal{A}(\cdot, \ldots, \cdot)$ is the aggregation operator of multiple 3D point clouds, and $\xi_1$ and $\mathcal{X}_i$ are the pose and the 3D point cloud of the $i$th agent, respectively. To ensure the inputs to the teacher model and the student model are aligned, we transform the holistic-view point cloud $\mathcal{X}$ to an agent’s own coordinate based on the pose information. Now, for the $i$th agent, the input to the teacher model $\xi_i^{-1} \odot \mathcal{X}$ and the input to the student model $\mathcal{X}_i$ are in the same coordinate system. Similarly to the feature encoder in the student model, we convert the 3D point cloud $\xi_i^{-1} \odot \mathcal{X}$ to a BEV map and use 2D convolutions to obtain the feature map of the $i$th agent in the teacher model, $\mathbf{H}_i^t \in \mathbb{R}^{K \times K \times C}$. Here we crop the BEV map to ensure it has the same spatial range and resolution with the BEV map in the student model.

Decoder and header. Similarly to the decoder and header in the student model, we adopt $\Theta^t$ to $\Psi^t(\mathbf{H}_i^t)$ to obtain the decoded BEV-based feature map and $\hat{Y}_i^t \leftarrow \Phi^t(\mathbf{M}_i^s)$ to obtain the predicted foreground-background categories and regressed bounding boxes.

Teacher training scheme. As in common teacher-student frameworks, we train the teacher model separately. We employ the binary cross-entropy loss to supervise foreground-background classification and the smooth $L_1$ loss to supervise the bounding-box regression. Overall, we minimize the loss function $\mathcal{L}_t = \sum_{i=1}^{M} \mathcal{L}_{\text{det}}(\hat{Y}_i^t, \mathbf{Y}_i^t) + \lambda_{\text{kd}} \mathcal{L}_{\text{kd}}(\mathbf{H}_i^t, \mathbf{H}_i^s) + \lambda_{\text{kd}} \mathcal{L}_{\text{kd}}(\mathbf{M}_i^s, \mathbf{M}_i^t)$.

The detection loss $\mathcal{L}_{\text{det}}$ is similar to that of the teacher, including both foreground-background classification loss and the bounding box regression loss, pushing the detection result of each agent to be close to its local ground-truth. The second and third terms form a knowledge distillation loss, regularizing the student model to generate similar feature maps with the teacher model. The hyperparameter $\lambda_{\text{kd}}$ controls the weight of the knowledge distillation loss $\mathcal{L}_{\text{kd}}$ defined as follows:

$$\mathcal{L}_{\text{kd}}(\mathbf{H}_i^s, \mathbf{H}_i^t) = \sum_{n=1}^{K \times K} \mathcal{D}_{KL}(\sigma((\mathbf{H}_i^s)_n)||\sigma((\mathbf{H}_i^t)_n)),$$

where $\mathcal{D}_{KL}(p(x)||q(x))$ denotes the Kullback-Leibler (KL) divergence of distribution $q(x)$ from distribution $p(x)$, $\sigma(\cdot)$ indicates the softmax operation of the feature vector along the channel dimension, and $(\mathbf{H}_i^s)_n$ and $(\mathbf{H}_i^t)_n$ denote the feature vectors at the $n$th cell of the $i$th agent’s feature map in the student model and teacher model, respectively. Similarly, the loss on decoded feature maps $\mathcal{L}_{\text{kd}}(\mathbf{M}_i^s, \mathbf{M}_i^t)$ can be introduced to enhance the regularization.

Remark. As mentioned in Section 2.1, the feature maps output by the collaboration graph process in the student model is computed by $\{\mathbf{F}_i^s\}_{i=1}^M \leftarrow \mathcal{M}_{\mathcal{G}_i}((\mathbf{F}_i^s)^{M}_{i=1})$. Intuitively, the feature map of the teacher model $\{\mathbf{H}_i^t\}_{i=1}^M$ would be the desired output of the collaboration graph process $\mathcal{M}_{\mathcal{G}_i}(\cdot)$. Therefore, we constrain all the post-collaboration feature maps in the student model to match the correspondences in the teacher model through knowledge distillation. This constraint would further regularize the upfront trainable components: i) the distilled student encoder $\Theta^s$, which abstracts the features from raw measurements and produces the input to $\mathcal{M}_{\mathcal{G}_i}(\cdot)$, and ii) the edge-weight encoder $\Pi$ in the distilled collaboration graph. Consequently, through knowledge distillation and back-propagation, the distilled student encoder would learn to abstract informative features from raw data for better collaboration; and the distilled edge-weight encoder would learn how to control the collaboration based on agents’ features. In a word, our distilled collaboration network (DiscoNet) can comprehend feature abstraction and fusion via the proposed knowledge distillation framework.

3 Related Work

Multi-agent communication. As core components in a multi-agent system, communication strategy among agents has been actively studied in previous works [30, 7, 31]. For example, CommNet [32] adopted the averaging operation to achieve continuous communication in multi-agent system; VAIN [11] considered an attention mechanism to determine which agents would share information;
Figure 3: V2X-Sim 1.0 dataset. (a) Point clouds from multiple agents in a top-down view (each color indicates an individual agent). (b) Snapshot of the rendered CARLA-SUMO co-simulation. (c) The map of Town05 which is a squared-grid town with cross junctions and multiple lanes per direction.

ATOC [12] exploited an attention unit to determine what information to share with other agents; and TarMAC [5] implicitly learned a signature-based soft attention mechanism to let agents actively select which agents should receive the messages. In this work, we focus on the 3D perception task and propose a trainable, dynamic collaboration graph to control the communication among agents.

Collaborative perception. Collaborative perception is an application of multi-agent communication system to perception tasks [9, 36, 14]. Who2com [20] exploited a handshake communication mechanism to determine which two agents should communicate for image segmentation; When2com [19] introduced an asymmetric attention mechanism to decide when to communicate and how to create communication groups for image segmentation; V2VNet [34] proposed multiple rounds of message passing on a spatial-aware graph neural network for joint perception and prediction in autonomous driving; and [33] proposed a pose error regression module to learn to correct pose errors when the pose information from other agents is noisy.

Knowledge distillation. Knowledge distillation (KD) is a widely used technique to compress a larger teacher network to a smaller student network by pushing the student to approach the teacher in either output or intermediate feature space [10], and it has been applied to various tasks such as semantic segmentation [21], single-agent 3D detection [35], and object re-identification [13]. In this work, we apply the KD technique to a new application scenario: multi-agent graph learning. The teacher is an early-collaboration model with privileged information, and the student is an intermediate-collaboration model with limited viewpoint. The distilled collaboration graph could enable effective and efficient inference for the student model without teacher’s supervision.

Advantages and limitation of DiscoNet. First, previous collaborative perception only rely on the final detection supervision; while DiscoNet leverages both detection supervision and intermediate feature supervision, acquiring more explicit guidance. Second, the collaboration attention in previous methods is a scalar, which cannot reflect the significance of each region; while DiscoNet uses a matrix, leading to more flexible collaboration at various spatial regions. Third, previous methods uses multiple-round collaboration: When2com needs at least one more round of communication after the first handshake, and V2VNet claims three rounds to ensure reliable performance; while DiscoNet only requires one round, suffering less from latency. Admittedly, DiscoNet has a limitation by assuming accurate pose for each agent, which could be improved by method like [33].

4 Experiment

4.1 V2X-Sim 1.0: A multi-agent 3D object detection dataset

Since there is no public dataset to support the research on multi-agent 3D object detection in self-driving scenarios, we build such a dataset named as V2X-Sim 1.0\(^2\), based on both SUMO [15], a micro-traffic simulation, and CARLA [6], a widely-used open-source simulator for autonomous driving research. SUMO is firstly used to produce numerically-realistic traffic flow and CARLA is

\(^2\)V2X (vehicle-to-everything) denotes the collaboration between a vehicle and other entities such as vehicle (V2V) and roadside infrastructure (V2I). V2X-Sim dataset is maintained on https://ai4ce.github.io/V2X-Sim/, and the first version of V2X-Sim used in this work includes the LiDAR-based V2V scenario.
employ v2x-sim 1.0 follows the same storage format of nuScenes [2], a widely-used autonomous driving dataset. nuScenes collected real-world single-agent data; while we simulate multi-agent scenarios. Each scene includes a 20 second traffic flow at a certain intersection of Town05, and the LiDAR streams are recorded every 0.2 second, so each scene consists of 100 frames. We generate 100 scenes with a total of 10,000 frames, and in each scene, we randomly select 2-5 vehicles as the collaboration agents. We use 8,000/900/1,100 frames for training/validation/testing. Each frame has multiple streams are recorded every 0.2 second, so each scene consists of 100 frames. We generate 100 scenes among intermediate collaboration. Even with 16 times compression, DiscoNet(16) still outperforms V2VNet without compression.

Table 1: Detection performance by regularizing various layers via knowledge distillation (KD). \( M^i \{ n \} (n = 1, 2, 3, 4) \) denotes different layers in the decoder. The performance significantly boosts once the KD regularization is applied. Further regularization has a slight effect on the detection performance.

<table>
<thead>
<tr>
<th>Collaboration Approach</th>
<th>Average Precision (AP) @IoU=0.5</th>
<th>Average Precision (AP) @IoU=0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Late</td>
<td>63.3</td>
<td>59.7</td>
</tr>
<tr>
<td>Early</td>
<td></td>
<td>60.2</td>
</tr>
<tr>
<td>Lower-bound</td>
<td></td>
<td>55.8</td>
</tr>
<tr>
<td>DiscoNet(8)</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>DiscoNet(16)</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>V2VNet [34]</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Who2com [20]</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Whom2com [19]</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

4.2 Quantitative evaluation

Implementation and evaluation. We crop the points located in the region of \([-32, 32] \times [-32, 32] \times [-3, 2]\) meters defined in the ego-vehicle \(XYZ\) coordinate. We set the width/length of each voxel as 0.25 meter, and the height as 0.4 meter; therefore the BEV map input to the student/teacher encoder has a dimension of \(256 \times 256 \times 13\). The intermediate feature output by the encoder has a dimension of \(32 \times 32 \times 256\), and we use an autoencoder (AE) for compression to save bandwidth: the sent message is the embedding output by AE. After the agents receive the embedding, they will decode it to the original resolution \(32 \times 32 \times 256\) for intermediate collaboration. The hyperparameter \(k_{AE}\) is set as \(10^5\). We train all the models using NVIDIA GeForce RTX 3090 GPU. We employ the generic BEV detection evaluation metric: Average Precision (AP) at Intersection-over-Union (IoU) threshold of 0.5 and 0.7. We target the detection of the car category and report the results on the test set.

Baselines and existing methods. We consider the single-agent perception model; called Lower-bound, which consumes a single-view point cloud. The teacher model based on early collaboration with a holistic view is considered as the Upper-bound. Both lower-bound and upper-bound can involve late collaboration, which aggregates all the boxes and applies a global non-maximum suppression (NMS). We also consider three intermediate-collaboration methods, Who2com [20], Whom2com [19] and V2VNet [34]. Since the original Who2com and Whom2com do not consider pose information, we consider both pose-aware and pose-agnostic versions to achieve fair comparisons. All the methods use the same detection architecture and conduct collaboration at the same intermediate feature layer.

Results and comparisons. Table 1 shows the comparisons in terms of AP (@IoU = 0.5/0.7). We see that i) early collaboration (first row) achieves the best detection performance, and there is an obvious improvement over no collaboration (last row), i.e., AP@0.5 and AP@0.7 are increased by 38.2% and 42.3% respectively, reflecting the significance of collaboration; ii) late collaboration improves the lower-bound (second last row), but hurts the upper-bound (second row), reflecting the unreliability of late collaboration. This is because the final, global NMS can remove a lot of noisy boxes for the lower-bound, but would also remove many useful boxes for the upper-bound; and iii) among the intermediate collaboration-based methods, the proposed DiscoNet achieves the best performance. Comparing to the pose-aware Whon2com, DiscoNet improves by 31.9% in AP@0.5 and 29.3% in
Table 3: Comparison of various intermediate collaboration strategies. KD indicates knowledge distillation. With KD and matrix-valued edge weights, our DiscoGraph outperforms the others.

<table>
<thead>
<tr>
<th>Collaboration Strategy</th>
<th>No</th>
<th>Sum</th>
<th>Average Weight</th>
<th>Weighted Average</th>
<th>Max</th>
<th>Cat DiscoGraph</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o KD (AP@IoU 0.5/0.7)</td>
<td>45.8/42.3</td>
<td>55.7/50.9</td>
<td>55.7/50.4</td>
<td>56.1/51.4</td>
<td>56.7/51.4</td>
<td>55.0/50.2</td>
</tr>
<tr>
<td>w/ KD (AP@IoU 0.5/0.7)</td>
<td>46.5/42.9</td>
<td>54.4/46.1</td>
<td>56.4/51.1</td>
<td>56.7/50.9</td>
<td>56.7/51.8</td>
<td>57.5/52.6</td>
</tr>
<tr>
<td>Gain</td>
<td>0.7/0.6</td>
<td>-1.3/-4.6</td>
<td>0.6/0.5</td>
<td>1.1/-1.0</td>
<td>0.0/0.4</td>
<td>2.5/2.4</td>
</tr>
<tr>
<td>Gain (%)</td>
<td>1.5/1.4</td>
<td>-2.3/-9.0</td>
<td>1.3/1.4</td>
<td>1.1/-1.0</td>
<td>0.0/0.7</td>
<td>3.3/4.6</td>
</tr>
</tbody>
</table>

Figure 4: Performance-bandwidth trade-off. DiscoNet(64) achieves 192 times less communication volume and still outperforms V2VNet [34].

AP@0.7. Comparing to V2VNet, DiscoNet improves by 6.2 \% in AP@0.5 and 6.3 \% in AP@0.7. Even with 16-times compression by the autoencoder, DiscoNet(16) still outperforms V2VNet.

Performance-bandwidth trade-off analysis. Fig. 4 thoroughly compares the proposed DiscoNet with the baseline methods in terms of the trade-off between detection performance and communication bandwidth. In Plots (a) and (b), the lower-bound with no collaboration is shown as a dashed line since the communication volume is zero, and the other method is shown as one dot in the plot. We see that i) DiscoNet with feature-map compression by an autoencoder achieves a far-more superior trade-off than the other methods; ii) DiscoNet(64), which is DiscoNet with 64-times feature compression by autoencoder, achieves 192 times less communication volume and still outperforms V2VNet for both AP@0.5 and 0.7; and iii) feature compression does not significantly hurt the detection performance.

Comparison of training times with and without KD. Incorporating KD will increase the training time a little bit, from \textasciitilde1200s to \textasciitilde1500s per epoch (each epoch consists of 2,000 iterations). However, during inference, DiscoNet does not need the teacher model anymore and can work alone without extra computations. Note that our key advantage is to push the computation burden into the training stage in pursuit of efficient and effective inference which is crucial in the real-world deployment.

4.3 Qualitative evaluation
Visualization of edge weight. To understand the working mechanism of the proposed collaboration graph, we visualize the detection results and the corresponding edge weights; see Fig. 5 and Fig. 7. Note that the proposed edge weight is a matrix, reflecting the collaboration attention in a cell-level resolution, which is shown as a heat map. Fig. 5 (a), (b) and (c) show three exemplar detection results of Agent 1 based on the lower-bound, upper-bound and the proposed DiscoNet, respectively. We see that with the collaboration, DiscoNet is able to detect boxes in those occluded and long-range regions. To understand why, Fig. 5 (d) and (e) provide the corresponding ego edge weight and the edge weight from agent 2 to agent 1, respectively. We clearly see that with the proposed collaboration graph, Agent 1 is able to receive complementary information from the other agents. Take the first row in Fig. 5 as an example, we see that in Plot (d), the bottom-left spatial region has a much darker color, indicating that Agent 1 has less confidence about this region; while in Plot (e), the bottom-left spatial region is much brighter, indicating that Agent 1 is receiving confidence from Agent 2. This shows the effectiveness of the proposed collaboration framework.
Figure 5: Detection and matrix-valued edge weight for Agent 1. Green and red boxes denote ground-truth (GT) and predictions, respectively. (a-c) shows the outputs of lower-bound, upper-bound and DiscoNet compared to GT. (d) Ego matrix-valued edge weight for Agent 1. Attention to the region with sparse/no measurement is suppressed. (e) Matrix-valued edge weight from Agent 2 to Agent 1. Attention to the region with complementary information from Agent 2 is enhanced.

Figure 6: DiscoNet qualitatively outperforms the state-of-the-art methods. Green and red boxes denote ground-truth and detection, respectively. (a) Output of when2com. (b) Output of V2VNet. (c) Output of DiscoNet. (d)-(f) Matrix-valued edge weights. (Ego agent: 1; neighbour agents: 2 and 3.)

Comparison with When2com and V2VNet. Fig. 6 shows two examples to compare the proposed DiscoNet with When2com, V2VNet. We see that DiscoNet is able to detect more objects. The reason is that both V2VNet and when2com employ a scalar to denote the agent-to-agent attention, which cannot distinguish which region is more informative; while DiscoNet can adaptively find beneficial region in a cell-level resolution; see the visualization of matrix-valued edge weights in Fig. 6 (d)-(f).

4.4 Ablation study

Effect of collaboration strategy. Table 3 compares the proposed method with five baselines using different intermediate collaboration strategies. *Sum* simply sums all the intermediate feature maps. *Average* calculates the mean of all the feature maps. *Max* selects the maximum value of all the feature maps at each cell to produce the final feature. *Cat* concatenates the mean of others’ features with its own feature, and then uses a convolution layer to halve the channel number. *Weighted Average* appends one convolutional layer to our edge encoder \( \Pi \), to generate a scalar value as the agent-wise edge weight. We use a matrix-valued edge weight to model the collaboration among agents. Without knowledge distillation, *Max* has achieved the second best performance probably because it ignores
Figure 7: Detection and matrix-valued edge weight for Agent 1. Green/red boxes denote GT/predictions. (a)(b) show the outputs of upper-bound and DiscoNet. (c) Ego edge weight for Agent 1. (d) Edge weight from Agent 2 to Agent 1. The spatial regions containing complementary information for Agent 1 are highlighted by green circles in (d) and red circles in (a).

the noisy information with lower value. We see that with the proposed matrix-valued edge weight and knowledge distillation, DiscoGraph significantly outperforms all the other collaboration choices.

**Effect of knowledge distillation.** In Table 3 we investigate the versions with and without knowledge distillation. We see that i) for our method, the knowledge distillation can guide the learning of collaboration graph, and the agents can work collaboratively to approach the teacher’s performance; ii) for max without a learnable module during collaboration, knowledge distillation has no impact; and iii) for cat and average, their performances are improved a little bit, as knowledge distillation can influence the feature abstraction process. Table 2 further shows the detection performances when we apply knowledge-distillation regularization at various network layers. We see that i) once we apply knowledge distillation to regularize the feature map, the proposed method starts to achieve improvement; and ii) applying regularization on four layers has the best performance.

5 Conclusion

We propose a novel intermediate-collaboration method, called distilled collaboration network (DiscoNet), for multi-agent perception. Its core component is a distilled collaboration graph (DiscoGraph), which is novel in both the training paradigm and the edge weight setting. DiscoGraph is also pose-aware and adaptive to perception measurements, allowing multiple agents with the shared DiscoNet to collaboratively approach the performance of the teacher model. To validate, we build V2X-Sim 1.0, a large-scale multi-agent 3D object detection dataset based on CARLA and SUMO. Comprehensive quantitative and qualitative experiments show that DiscoNet achieves appealing performance-bandwidth trade-off with a more straightforward design rationale.

**Acknowledgment.** This research is partially supported by the NSF CPS program under CMMI-1932187, the National Natural Science Foundation of China under Grant 6217010074, and the Science and Technology Commission of Shanghai Municipal under Grant 21511100900. The authors gratefully acknowledge the useful comments and suggestions from anonymous reviewers.
References


[28] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. IEEE transactions on pattern analysis and machine intelligence, 2020. 2


