What Makes Good Collaborative Views? Contrastive Mutual Information Maximization for Multi-Agent Perception

Wanfang Su[*,1,2], **Lixing Chen**[*,1,2], **Yang Bai**[†,1], **Xi Lin**[1,2], **Gaolei Li**[1,2], **Zhe Qu**[3], **Pan Zhou**[4]

[1]School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China
[2]Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai, China
[3]School of Computer Science and Engineering, Central South University, Changsha, China
[4] Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, Huazhong University of Science of Technology, Wuhan, China
{77susu, lxchen, ybai, linxi234, gaolei_li}@sjtu.edu.cn, zhe_qu@csu.edu.cn, zhoupannewton@gmail.com

Abstract

Multi-agent perception (MAP) allows autonomous systems to understand complex environments by interpreting data from multiple sources. This paper investigates intermediate collaboration for MAP with a specific focus on exploring "good" properties of collaborative view (i.e., post-collaboration feature) and its underlying relationship to individual views (i.e., pre-collaboration features), which were treated as an opaque procedure by most existing works. We propose a novel framework named CMiMC (Contrastive Mutual Information Maximization for Collaborative Perception) for intermediate collaboration. The core philosophy of CMiMC is to preserve discriminative information of individual views in the collaborative view by maximizing mutual information between pre- and post-collaboration features while enhancing the efficacy of collaborative views by minimizing the loss function of downstream tasks. In particular, we define multi-view mutual information (MVMI) for intermediate collaboration that evaluates correlations between collaborative views and individual views on both global and local scales. We establish CMiMNet based on multi-view contrastive learning to realize estimation and maximization of MVMI, which assists the training of a collaboration encoder for voxel-level feature fusion. We evaluate CMiMC on V2X-Sim 1.0, and it improves the SOTA average precision by 3.08% and 4.44% at 0.5 and 0.7 IoU (Intersection-over-Union) thresholds, respectively. In addition, CMiMC can reduce communication volume to $1/32$ while achieving performance comparable to SOTA. Code and Appendix are released at https://github.com/77SWF/CMiMC.

Introduction

Multi-Agent perception (MAP) (Wang et al. 2023; Xu et al. 2023a) refers to the process by which multiple entities work together to gather a interpret sensory information and establish a collective comprehension of the environment. MAP enhances perception capabilities over the single-agent perception by mitigating inherent limitations arising from individual perspectives (Li et al. 2021). For example, using MAP in autonomous driving (Cui et al. 2022; Chen et al. 2017) effectively addresses challenges caused by occlusions and sparse observations in long-distance areas.

Collaboration strategy is one of the fundamental components of MAP, which involves protocols for communication, data sharing, and coordination among multiple agents. State-of-the-art collaboration strategies can be categorized into three types: early collaboration, late collaboration, and intermediate collaboration. *Early Collaboration* (Chen et al. 2019; Arnold et al. 2020) aggregates the raw measurement from agents to generate a comprehensive view. However, it requires substantial communication bandwidth and incurs privacy leakage risk. *Late Collaboration* (Miller et al. 2020) aggregates perception results of individual agents, which is bandwidth-efficient. However, individual perception results are noisy and coarse-grained, and therefore may cause inferior performance for collaborative perception. To strike a balance between performance and bandwidth usage, *Intermediate Collaboration* has been proposed (Liu et al. 2020b; Hu et al. 2022) to aggregate intermediate features of agents. The intermediate features distill compact knowledge representations from raw measurement, offering the potential for both communication efficiency and enhanced perception capability. Nevertheless, intermediate collaboration strategies require judicious design, otherwise, may result in substantial information loss and adversely impact the effectiveness of MAP.

Recent endeavors have been dedicated to the advancement of intermediate collaboration strategies. For example, V2VNet (Wang et al. 2020) proposes to exchange intermediate features and integrate them via a spatial-aware graph neural network for improving motion prediction performance; When2com (Liu et al. 2020a) constructs a communication group based on general attention mechanisms to determine the optimal timing for intermediate collaboration; DiscoNet (Li et al. 2021) trains a distilled collaboration network to push the results of intermediate collaboration close to that of early collaboration. The majority of current works treat the intermediate collaboration as an opaque process and approximate it with deep neural networks that are trained to optimize the performance of downstream tasks or other specified design goals. However, such a process can eas-

[*]These authors contributed equally.
[†]Corresponding authors.

ily encounter performance bottlenecks as it runs the risk of losing valuable features solely in the pursuit of optimizing the pre-defined objectives. Evidence in Hjelm et al. (2019) demonstrates that features generated by unsupervised learning outperform those generated by supervised learning that minimizes downstream loss. This paper aims to delve into fundamental principles of feature aggregation in intermediate collaboration, exploring the characteristics that contribute to good collaborative views (i.e., post-collaboration features) and how good collaborative views relate to views of individual agents (i.e., pre-collaboration features).

Intuitively, it is desirable for collaborative views to aggregate all discriminative information in individual views. Drawing upon this intuition, we introduce the concept of mutual information (MI) maximization into intermediate collaboration. MI maximization is initially used in representation learning (Hjelm et al. 2019; Linsker 1988) to find features that capture the underlying dependencies in the data. We extend its ability to intermediate collaboration for obtaining collaborative views that retain informative parts of individual views and discard irrelevant or redundant ones. The **contributions** of this paper are summarized as follows:

1. We propose a novel framework named CMiMC (Contrastive Mutual Information Maximization for Colloabrative perception) for intermediate collaboration. CMiMC defines multi-view mutual information (MVMI) that properly measures the global and local dependencies between a collaborative view and multiple individual views. We design an MVMI maximization strategy and plug it into the supervised learning process of MAP. This enables CMiMC to construct collaborative views that retain discriminative information of individual views and also provide enhanced effectiveness to downstream tasks.

2. We establish CMiMNet based on multi-view contrastive learning to estimate and maximize MVMI in an unsupervised fashion by drawing close the collaborative view and individual views that come from the same scene (i.e., positive pairs) and pushing away individual views from different scenes (i.e., negative pairs). CMiMNet enables the intermediate collaboration strategy to identify critical regions in individual views and complete fine-grained feature aggregation at the voxel resolution.

3. We evaluate our method on V2X-Sim 1.0 (Li et al. 2021). Experimental results show that CMiMC outperforms state-of-the-art (SOTA) benchmarks in terms of average precision (AP) and performance-bandwidth trade-offs. CMiMC improves SOTA AP by 3.08%\4.44% at Intersection-over-Union (IoU) $0.5\backslash 0.7$. It can reduce communication volume to $1/32$ while achieving AP comparable to SOTA.

Related Work

Collaborative Perception The existing works of collaborative strategies can be categorized into three types: Early Collaboration (Chen et al. 2019) involves sharing raw measurement, Late Collaboration (Miller et al. 2020) involves sharing perception results, and Intermediate Collaboration involves sharing intermediate features. Among these approaches, intermediate collaboration has attracted significant attention as it strikes a balance between perception performance and communication bandwidth. For example, When2com (Liu et al. 2020a) forms a communication group based on the general attention mechanism to determine the optimal timing for intermediate collaboration. V2VNet (Wang et al. 2020) broadcasts and receives intermediate features using a spatially aware GNN to improve motion prediction performance. DiscoNet (Li et al. 2021) utilizes knowledge distillation to push the results of intermediate collaboration close to that of early collaboration. Where2comm (Hu et al. 2022) proposes a spatial confidence map to reflect the spatial heterogeneity of perceptual information. V2X-ViT (Xu et al. 2022) fuses information across heterogeneous agents using a multi-agent attention module. CRCNet (Luo et al. 2022) studies feature selection by sequentially incorporating features that differ the most from fused ones. Most of the above works treat feature aggregation of intermediate collaboration as a black box and directly use deep neural networks to learn collaboration strategies that optimize downstream tasks or other specified goals. By contrast, our work focuses on the feature aggregation stage, aiming to unravel the characteristics that contribute to the formation of a good collaborative view.

Mutual Information Estimation and Maximization Mutual information (MI) is commonly used for quantifying the correlation between random variables. This paper leverages MI maximization to enhance the performance of intermediate collaboration. Existing works have studied MI estimation and maximization. Early non-parametric methods include binning (Darbellay and Vajda 1999), kernel density estimator (Silverman 1981), and density-ratio-based likelihood estimator (Hido et al. 2011). However, these works are limited to handling low-dimensional data. Recent works introduce neural networks into MI estimation, which has proven effective for handling high-dimensional data. For example, MINE (Belghazi et al. 2018) and InfoNCE (Oord, Li, and Vinyals 2018) estimate MI by learning to maximize variational bounds. Deep InfoMax (Hjelm et al. 2019), Info3D (Sanghi 2020) studies unsupervised representation learning by maximizing MI between encoder input and output. However, these works only consider the MI estimation for two variables. To address this issue, Contrastive Multiview Coding (CMC) (Tian, Krishnan, and Isola 2020) is proposed to capture dependencies among multiple views. Inspired by these works, we introduce MI maximization to intermediate collaboration for constructing collaborative views that retain discriminative information of individual views. We designed multi-view mutual information (MVMI) that is tailed for intermediate collaboration to measure the dependencies of multiple variables in a one-to-many setting (i.e., one collaborative view to many individual views). MVMI also takes into account MI evaluations on both global and local scales. In addition, we design CMiMNet to estimate and maximize MVMI based on contrastive learning.

Methods

For ease of presentation, we introduce CMiMC in the context of LiDAR-based 3D object detection. However, we note that CMiMC is also compatible with other MAP scenarios.

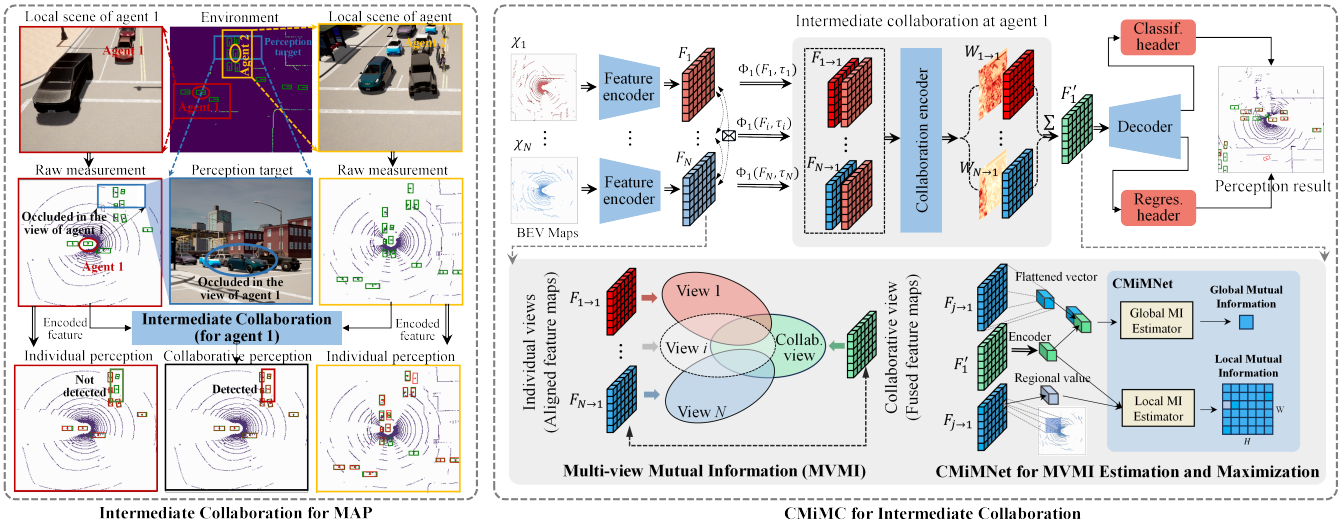


Figure 1: Scenario of intermediate collaboration for LiDAR-based 3D object detection and the framework of CMiMC.

The CMiMC Framework

We illustrate the scenario of intermediate collaboration for LiDAR-based 3D object detection and the framework of CMiMC in Fig. 1. The upper part of CMiMC shows the pipeline of MAP, which comprises three parts: 1) a feature encoder for extracting features from raw measurements; 2) a collaboration encoder for aggregating features received from other agents; 3) a decoder-and-header module for decoding fused features and generating perception results. The bottom part of CMiMC illustrates the MI maximization module which encompasses the definition of Multi-View Mutual Information (MVMI) on global/local scales and CMiMNet for estimating and maximizing MVMI.

All agents utilize the same feature encoder, collaboration encoder, and decoder-header module to perform collaborative perception. CMiMNet is exclusively used during training, and agents do not need to execute CMiMNet during inference. In the following, we give the procedures for performing the above components. The architectures of encoders/decoders are given in Appendix A.1.

Feature Encoder Each agent possesses a feature encoder to extract features from raw measurements. In the context of LiDAR-based 3D object detection, the raw measurements are 3D point clouds which will be converted into Bird's-Eye View (BEV) maps (see Appendix A.2 for details) and fed to the feature encoder. The BEV map of agent i is denoted by χ_i. The feature map extracted by the feature encoder is $F_i = \Psi_{\texttt{enc}}(\chi_i) \in \mathbb{R}^{H\times W\times C}$, where $\Psi_{\texttt{enc}}$ denotes the feature encoder and H, W, and C denote the height, width, and channel of the intermediate feature map. We also call the feature map F_i the *individual view* of agent i.

Collaboration Encoder Agents exchange individual views via peer-to-peer communication and utilize a collaboration encoder to generate a collaborative view (i.e., a fused feature map). Because each agent perceives the environment using a local coordinate system with itself as the origin, an agent needs to align the received views before feature aggregation. We consider that agents exchange pose information along with intermediate features, and let τ_i denote the pose information of agent i. The feature alignment at agent i for feature F_j with pose information τ_j is denoted as $F_{j\to i} = \Phi_i(F_j, \tau_j)$, where Φ_i is a feature pose transformation function based on pose information τ_i. Given the aligned feature maps, the collaboration encoder outputs spatial weight matrices for voxel-level feature aggregation. The weight matrix for aggregating agent j's feature at agent i is $W_{j\to i} = \Psi_{\texttt{col}}(F_{j\to i}, F_i)$, where $\Psi_{\texttt{col}}$ is the collaboration encoder. The weight matrix $W_{j\to i}$ has the same size as $F_{j\to i}$. Each element in $W_{j\to i}$ corresponds to a voxel at the same position in $F_{j\to i}$ and reflects the importance of the regional feature. Finally, agent i can obtain a fused feature map by

$$F'_i = \sum_{j=1}^{N} \Psi_{\texttt{col}}(F_{j\to i}, F_i) \otimes \Phi_i(F_j, \tau_j), \qquad (1)$$

where N is the number of agents in the multi-agent perception task and $\otimes$ denotes element-wise multiplication.

Decoder-Header Module Given collaborative view F'_i, the decoder-header module first decodes it to a feature map $\varrho_i = \Psi_{\texttt{dec}}(F'_i)$ via decoder $\Psi_{\texttt{dec}}$. The decoded feature is then fed to a classification header (for classifying the foreground-background categories) and a regression header (for generating the bounding boxes). The perception results at agent i can be represented by $Y_i = \Psi_{\texttt{head}}(\varrho_i)$, where $\Psi_{\texttt{head}}$ denotes a combination of headers.

Multi-view Mutual Information Maximization for Intermediate Collaboration Agents in MAP capture diverse views of the environment, each offering a unique insight. Empirical evidence in numerous real-world scenarios (Tian, Krishnan, and Isola 2020) supports the conventional belief that valuable information often resides in different views. However, how to gather and consolidate valuable informa-

tion dispersed across individual views with minimal information loss is still challenging. To address this challenge, we introduce mutual information maximization into intermediate collaboration. Unlike previous approaches that construct collaborative views by minimizing the loss function of downstream tasks, our method aims to retain the discriminative features of individual views in the collaborative view by maximizing mutual information between them.

We use mutual information (MI) to quantify the dependencies between individual views and collaborative views. Consider an individual view $F \in \{F_{1\to i}, \ldots, F_{N\to i}\}_{i=1}^{N}$ and a collaborative view $F' \in \{F'_i\}_{i=1}^{N}$, the MI between F and F' is defined by $\mathcal{I}(F, F') = \int_{F\times F'} \log \frac{\mathrm{d}\mathbb{P}_{FF'}}{\mathrm{d}\mathbb{P}_F \mathbb{P}_{F'}} \mathrm{d}\mathbb{P}_{FF'}$. The Kullback-Leibler (KL) divergence between $\mathbb{P}_{FF'}$ and $\mathbb{P}_F\mathbb{P}_{F'}$ equals the above MI definition (Belghazi et al. 2018), and hence we can use KL divergence to reflect the overall dependency between an individual view and a collaborative view, which we call global MI: $\mathcal{I}_{\mathrm{G}}(F, F') = D_{\mathrm{KL}}(\mathbb{P}_{FF'} || \mathbb{P}_F\mathbb{P}_{F'})$. Global MI has been widely used in representation learning, however, it also has been shown to be insufficient for perception tasks when the structure of features matter (Bachman, Hjelm, and Buchwalter 2019). Therefore, we further use local MI to evaluate the dependencies between the collaborative view and regional feature patches of the individual views. By doing so, we can properly capture the spatial structure of individual views. The local MI can be obtained by:

$$\mathcal{I}_{\mathrm{L}}(F, F') = \frac{1}{H \times W} \sum\nolimits_{k=1}^{H\times W} \mathcal{I}(F(k), F'), \tag{2}$$

where H and W are the height and width of F and $F(k)$ denote k-th regional feature patch in F.

Vanilla MI is designed to measure the dependency between two random variables and does not match the scenario of intermediate collaboration where an agent needs to measure MI between one collaborative view and multiple individual views. To fill this gap, we define Multi-View Mutual Information (MVMI) which is tailored to this one-to-many MI evaluation setting. To be specific, the collaborative view F'_i is the core view for the collaboration encoder to optimize. MVMI first builds pair-wise MI (both global MI and local MI) between F'_i and each individual view $F_{j\to i}$. Then, pair-wise MIs are averaged to form MVMI. Mathematically, MVMI for agent i is calculated as:

$$\mathcal{I}_{\mathrm{MV},i} = \underbrace{\frac{\beta_{\mathrm{G}}}{N} \sum_{j=1}^{N} \mathcal{I}_{\mathrm{G}}(F_{j\to i}, F'_i)}_{\text{Global MVMI}} + \underbrace{\frac{\beta_{\mathrm{L}}}{N} \sum_{j=1}^{N} \mathcal{I}_{\mathrm{L}}(F_{j\to i}, F'_i)}_{\text{Local MVMI}}, \tag{3}$$

where β_{G} and β_{L} are weights for global and local MVMI. However, MI is notoriously difficult to compute as underlying distributions of features (i.e., $\mathbb{P}_F$, $\mathbb{P}_{F'}$ and $\mathbb{P}_{FF'}$) are unknown in practice, particularly for continuous and high-dimensional intermediate features in our work. In the following, we build CMiMNet based on contrastive learning to realize the estimation and maximization of MVMI.

CMiMNet

We now present CMiMNet for MVMI maximization.

MVMI Maximization via Contrastive Learning The goal of contrastive learning is to learn a discriminator $T_\theta(\cdot)$ (parameterized by θ) to distinguish if two views, e.g., F and F', come from different distributions $\mathbb{P}_{FF'}$ and $\mathbb{P}_F\mathbb{P}_{F'}$. The discriminator $T_\theta(\cdot)$ is trained to minimize the contrastive loss by assigning a high score to positive sample pair (i.e., samples from the same distribution) $(F_{j\to i}, F'_i)_{\forall j\in[1,N]} \sim \mathbb{P}_{FF'}$ and low score to negative sample pair (i.e., samples from different distributions) $(F_{j\to k}, F'_i)_{\forall j\in[1,N], k\neq i} \sim \mathbb{P}_F\mathbb{P}_{F'}$. To put it understandable in our context (Fig. 2), T_θ draws close collaborative views and individual views that come from the same scene and pushes away individual views from different scenes. The loss of T_θ is termed as contrastive loss $\mathcal{L}_{\mathrm{CTRS}}$, which can be used to estimate the lower bound of MI (Oord, Li, and Vinyals 2018; Poole et al. 2019), i.e.,

$$\mathcal{I}(F, F') \geq \log(K) - \mathcal{L}_{\mathrm{CTRS}}(T_\theta(\cdot, \cdot)) = \hat{\mathcal{I}}(F, F'), \tag{4}$$

where K represents the number of negative sample pairs. Minimizing the contrastive loss equivalently maximizes the lower bound of MI $\mathcal{I}(\cdot)$. Therefore, we can define the negative value of estimated MI $\hat{\mathcal{I}}(\cdot)$ as the contrastive loss.

Now, the problem becomes estimating MI $\hat{\mathcal{I}}(\cdot)$. Existing works often use KL-divergence-based MI estimators, e.g., MINE (Belghazi et al. 2018) and InfoNCE (Oord, Li, and Vinyals 2018), however, these methods require a large number of negative samples (Hjelm et al. 2019) to avoid unstable estimations. Therefore, CMiMNet utilizes an MI estimator based on Jensen-Shannon (JS) divergence (Nowozin, Cseke, and Tomioka 2016), which delivers a similar capability as KL divergence-based MI estimators but exhibits higher stability (Hjelm et al. 2019). The MI estimator is constructed on discriminator T_θ, formally defined by $\hat{\mathcal{I}}(F, F'; \theta) = \mathbb{E}_{\mathbb{P}_{FF'}}[-\delta(-T_\theta(F, F'))] - \mathbb{E}_{\mathbb{P}_F\mathbb{P}_{F'}}[\delta(T_\theta(\hat{F}, F'))]$, where (F, F') and $(\hat{F}, F')$ are positive and negative sample pairs; $\delta(x) = \log(1 + e^x)$ is the softplus function. Based on MI estimator $\hat{\mathcal{I}}(\cdot)$ and definitions of global/local MI, we can write global MI estimator and local MI estimator as $\hat{\mathcal{I}}_{\mathrm{G}}(F, F'; \theta_{\mathrm{G}}) = \hat{\mathcal{I}}(F, F'; \theta_{\mathrm{G}})$ and $\hat{\mathcal{I}}_{\mathrm{L}}(F, F'; \theta_{\mathrm{L}}) = \frac{1}{H\times W} \sum_{k=1}^{H\times W} \hat{\mathcal{I}}(F(k), F'; \theta_{\mathrm{L}})$, where θ_{G} and θ_{L} are the parameters of global and local discriminators, respectively. Given the definition of MVMI in (3), the objective of CMiMNet can be written as:

$$\max_{\theta_{\mathrm{G}}, \theta_{\mathrm{L}}} \frac{\beta_{\mathrm{G}}}{N} \sum_{j=1}^{N} \hat{\mathcal{I}}_{\mathrm{G}}(F_{j\to i}, F'_i; \theta_{\mathrm{G}}) + \frac{\beta_{\mathrm{L}}}{N} \sum_{j=1}^{N} \hat{\mathcal{I}}_{\mathrm{L}}(F_{j\to i}, F'_i; \theta_{\mathrm{L}}). \tag{5}$$

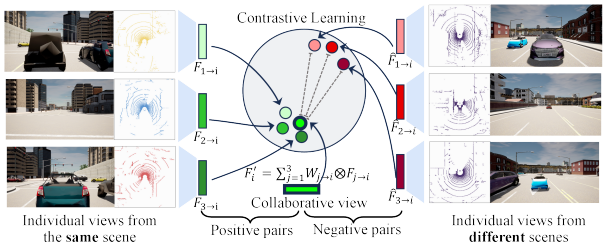


Figure 2: Contrastive learning over positive/negative pairs.

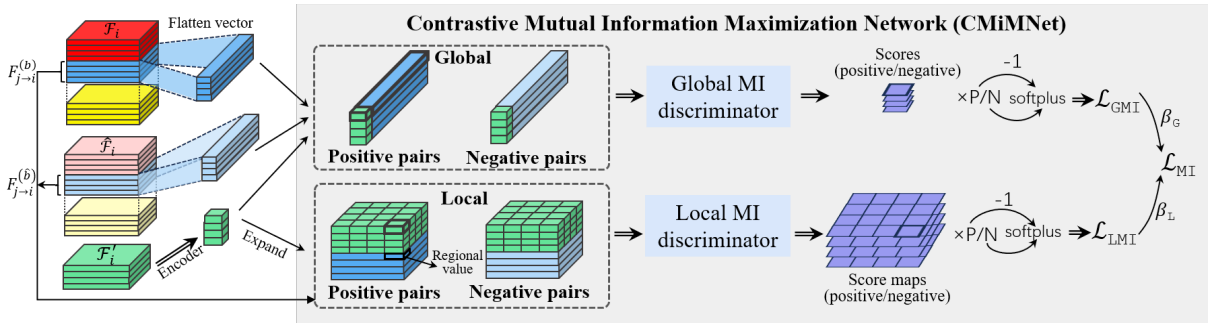


Figure 3: The structure of CMiMNet for estimating and maximizing MVMI.

Positive and Negative Pairs for CMiMNet CMiMNet learns in an unsupervised fashion using positive and negative sample pairs. In our problem, we pair collaborative views and individual views sampled from the same perception scene as positive pairs, and pair the collaborative views and individual views from different scenes as negative pairs (see Fig. 2 as an example). In each training iteration, we sample B scenes, denoted by $\mathcal{B} = \{1, 2, \ldots, B\}$. For each agent i, we collect its collaborative views of B scenes in set $\mathcal{F}'_i = \{F'^{(b)}_i\}_{b\in\mathcal{B}}$, and then collect the individual views for generating these collaborative views in set $\mathcal{F}_i = \{F^{(b)}_{1\to i}, F^{(b)}_{2\to i}, \ldots, F^{(b)}_{N\to i}\}_{b\in\mathcal{B}}$. We sample another B scenes, denoted by $\hat{\mathcal{B}}$, which are different from that in $\mathcal{B}$, i.e., $\hat{\mathcal{B}} \cap \mathcal{B} = \emptyset$. We get individual views of N agents about scenes in $\hat{\mathcal{B}}$, and collect them in $\hat{\mathcal{F}}_i = \{F^{(\hat{b})}_{1\to i}, F^{(\hat{b})}_{2\to i}, \ldots, F^{(\hat{b})}_{N\to i}\}_{\hat{b}\in\hat{\mathcal{B}}}$. Then, for b-th scene in $\mathcal{B}$, we can construct N positive pairs $\{(F^{(b)}_{j\to i}, F'^{(b)}_i)\}_{j=1}^N$ and N negative pairs $\{(F^{(\hat{b})}_{j\to i}, F'^{(b)}_i)\}_{j=1}^N$ for training global and local discriminators, i.e., $T_{\theta_{\text{G}}}$ and $T_{\theta_{\text{L}}}$, and consequently maximizing the objective in (5).

Global and Local MVMI Estimator The structure of the global and local MVMI estimator is presented in Fig. 3. For each sample pair (F, F'), the global MVMI estimator first encodes the collaborative view F' into a feature vector via a linear layer. Individual view F are flattened and concatenated to the feature vector, and then fed to the global discriminator $T_{\theta_{\text{G}}}$. The outputs of $T_{\theta_{\text{G}}}$ are scores for positive/negative sample pair, which can be used for global MVMI estimation. For the local MVMI estimator, the collaborative view F' is also converted to a feature vector using the encoder. Then, the generated feature vector is concatenated with regional values of individual features, which gives a set of vectors $\{[F(k), \omega(F')]\}_{k=1}^{H\times W}$ (where $\omega(\cdot)$ denotes the encoder). These vectors are fed to the local MI discriminator $T_{\theta_{\text{L}}}$, generating a $H \times W$ score map. The score maps of all positive and negative pairs can be used to estimate local MVMI. The architectures of global and local discriminators are given in Appendix A.1.

Loss Function During training, CMiMC updates the parameters of CMiMNet and other components in MAP simultaneously in a direction of minimizing overall loss $\mathcal{L}(\boldsymbol{\Psi}, \theta_{\text{G}}, \theta_{\text{L}})$ (where $\boldsymbol{\Psi} = (\Psi_{\texttt{enc}}, \Psi_{\texttt{col}}, \Psi_{\texttt{dec}}, \Psi_{\texttt{head}})$):

$$\mathcal{L}(\boldsymbol{\Psi}, \theta_{\text{G}}, \theta_{\text{L}}) = (1-\alpha)(\mathcal{L}_{\texttt{CLS}} + \mathcal{L}_{\texttt{REG}}) + \alpha\mathcal{L}_{\texttt{MI}}, \tag{6}$$

In Eqn. (6), $\mathcal{L}_{\texttt{CLS}}$ and $\mathcal{L}_{\texttt{REG}}$ are the losses for foreground-background classification and bounding box regression. They are associated with downstream tasks (i.e., LiDAR-based 3D object detection). $\mathcal{L}_{\texttt{MI}}$ is the loss for MVMI estimation. The variable $\alpha \in [0, 1]$ is used to adjust the importance of loss terms. As we consider both global MVMI and local MVMI, $\mathcal{L}_{\texttt{MI}}$ can be feather decomposed into $\mathcal{L}_{\texttt{MI}} = \lambda(\beta_{\text{G}}\mathcal{L}_{\texttt{GMI}} + \beta_{\text{L}}\mathcal{L}_{\texttt{LMI}})$, where λ is the weight to re-scale the MI loss. Based on the objective of CMiMNet in (5), we can define $\mathcal{L}_{\texttt{GMI}}$ and $\mathcal{L}_{\texttt{LMI}}$ as $\mathcal{L}_{\texttt{GMI}} = -\frac{1}{N}\sum_{j=1}^N \hat{\mathcal{I}}_{\text{G}}(F_{j\to i}, F'_i; \theta_{\text{G}})$ and $\mathcal{L}_{\texttt{LMI}} = -\frac{1}{N}\sum_{j=1}^N \hat{\mathcal{I}}_{\text{L}}(F_{j\to i}, F'_i; \theta_{\text{L}})$. By this definition, $\mathcal{L}_{\texttt{MI}}$ actually corresponds to $\mathcal{L}_{\texttt{CTRS}}$. The pseudocode for training CMiMC is presented in Algorithm 1 in Appendix A.3.

Experiments and Discussions

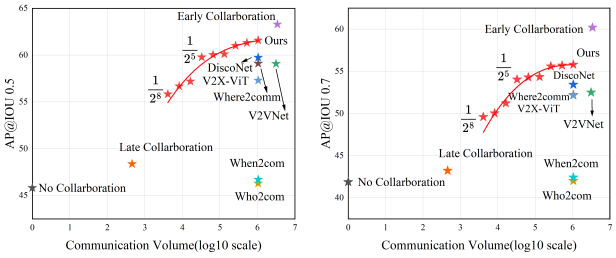
Experimental settings. We evaluate CMiMC on the V2X-Sim 1.0 dataset (Li et al. 2021). V2X-Sim 1.0 is a multi-agent 3D object detection dataset created through the joint simulation of CARLA-SUMO (Dosovitskiy et al. 2017). It consists of 10^4 frames from 100 scenes, and each scene contains 2-5 vehicles. We cropped the original point cloud to a region defined by $[-32, 32] \times [-32, 32] \times [-3, 2]$m (meters) and discretized it into a BEV map of size $(256, 256, 13)$. Each voxel has dimensions of 0.25m in length and width and 0.4m in height. The architectures of associated encoders/decoders are given in Appendix A.1. The batch size is set to 4. The initial learning rate for CMiMNet is 10^{-4}, and that for other components is 10^{-3}, both adopting a multi-step learning rate decay strategy. All experiments were conducted using PyTorch on two Nvidia Ampere A40 GPUs.

Benchmarks. The experiments use 10 benchmarks: No Collaboration (i.e., single-agent perception), Early Collaboration, Late Collaboration, Who2com (Liu et al. 2020b), When2com (Liu et al. 2020a), V2VNet (Wang et al. 2020), Where2comm (Hu et al. 2022), DiscoNet (Li et al. 2021), V2X-ViT (Xu et al. 2022) and a variant of DiscoNet w/o knowledge distillation (KD).

Performance comparison with benchmarks. Table 1 compares APs achieved by CMiMC and other benchmarks.

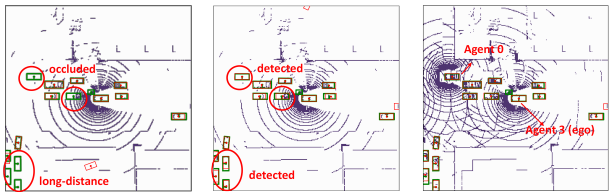
Method	Fusion Mode	Average Precision (AP) IoU = 0.5	IoU = 0.7	Noisy Case (AP@0.5) std = 0.2m	std = 0.4m	Communication Volume (KB)
No Collaboration	✗	45.82	41.89	45.82	45.82	0.00×10^{0}
Late collaboration	Late	48.40	43.23	-	-	4.30×10^{-1}
Who2com (ICRA 2020)	Interm.	46.31	42.02	46.19	46.03	1.05×10^{3}
When2com (CVPR 2020)	Interm.	46.71	42.42	46.64	46.18	1.05×10^{3}
V2VNet (ECCV 2020)	Interm.	59.08	52.50	57.69	54.49	2.99×10^{3}
DiscoNet w/o KD (NeurIPS 2021)	Interm.	58.91	53.33	-	-	1.05×10^{3}
DiscoNet (NeurIPS 2021)	Interm.	59.74	53.43	57.71	57.38	1.05×10^{3}
V2X-ViT (ECCV 2022)	Interm.	57.30	52.16	56.64	53.92	1.05×10^{3}
Where2comm (NeurIPS 2022)	Interm.	59.10	52.20	57.92	57.52	1.05×10^{3}
CMiMC	**Interm.**	**61.58**	**55.80**	**58.80**	**58.02**	1.05×10^{3}
CMiMC ($1/32$ compression ratio)	Interm.	59.78	54.04	58.94	57.52	$\mathbf{3.28 \times 10^{1}}$
Early Collaboration	Early	63.29	60.20	-	-	3.25×10^{3}

Table 1: Performance comparison on V2X-Sim 1.0.



(a) Trade-off in AP@ IoU 0.5 (b) trade-off in AP@IoU 0.7

Figure 4: Perfomance-bandwidth trade-off of CMiMC.



(a) No Collab. (b) CMiMC (ours) (c) Early Collab.

Figure 5: No Collaboration v.s. CMiMC v.s. Early Collaboration. Green/Red boxes are ground-truth/predicted boxes.

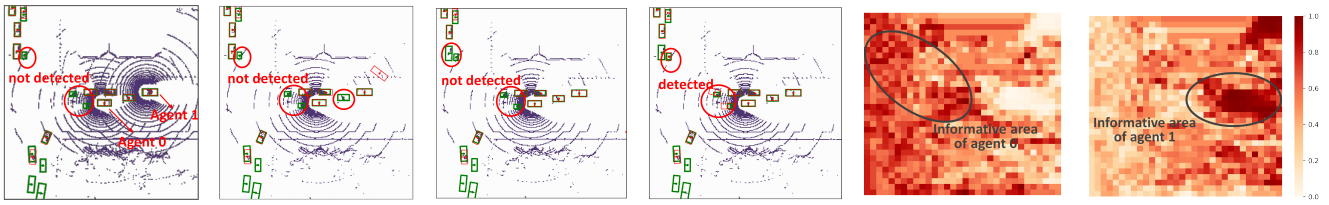
We see that Early Collab. and Late Collab. outperform No Collab., with AP improvements of 38.13%\5.63% and 43.71%\3.20% at IoU $= 0.5\backslash 0.7$. All Intermediate Collab. methods outperform No Collab., and CMiMC achieves the highest AP 61.58% (at IoU $= 0.5$), which is close to Early Collab. CMiMC improves the SOTA performance of DiscoNet by 3.08%\4.44% at IoU 0.5\0.7. DiscoNet applies knowledge distillation (KD), which uses a Early Collab. teacher to guide the training of collaboration encoder. Therefore, the performance of DiscoNet depends on the quality of Early Collab. This becomes a bottleneck for DiscoNet. Comparing DiscoNet w/o KD and DiscoNet, we see that using KD only improves AP by 1.41%\0.19% at IoU 0.5\0.7. By contrast, CMiMC uses contrastive learning to train the collaboration encoder in an unsupervised manner, thereby eliminating the above bottleneck. Compared to DiscoNet w/o KD, CMiMC provides AP improvement by 4.53%\4.63%, surpassing that of KD in DiscoNet.

Robustness to localization noise. We also consider the case of localization noise where the random noise (with a mean of 0m and a standard deviation of $0\backslash 0.2\backslash 0.4$m) is added to the pose information of agents. Based on the results in Table 1, we see a general trend that the performance degrades with the increase in localization noise. However, the proposed CMiMC still outperforms other benchmarks in all examined cases of localization noise.

Perfomance-bandwidth trade-off. Table 1 also shows the communication volume of CMiMC and other benchmarks. No Collab. uses single-agent perception and hence does not incur communication overhead. While Early Collab. achieves the highest AP, it also requires the largest communication volume (3.13×10^{3}KB) due to transmitting raw data of LiDAR point clouds. Late Collab. has a low communication volume but also has low AP. Intermediate collab. methods strike a balance between performance and bandwidth usage. Notably, our CMiMC provides higher AP than other Intermediate Collab. methods with comparable or even lower communication volume.

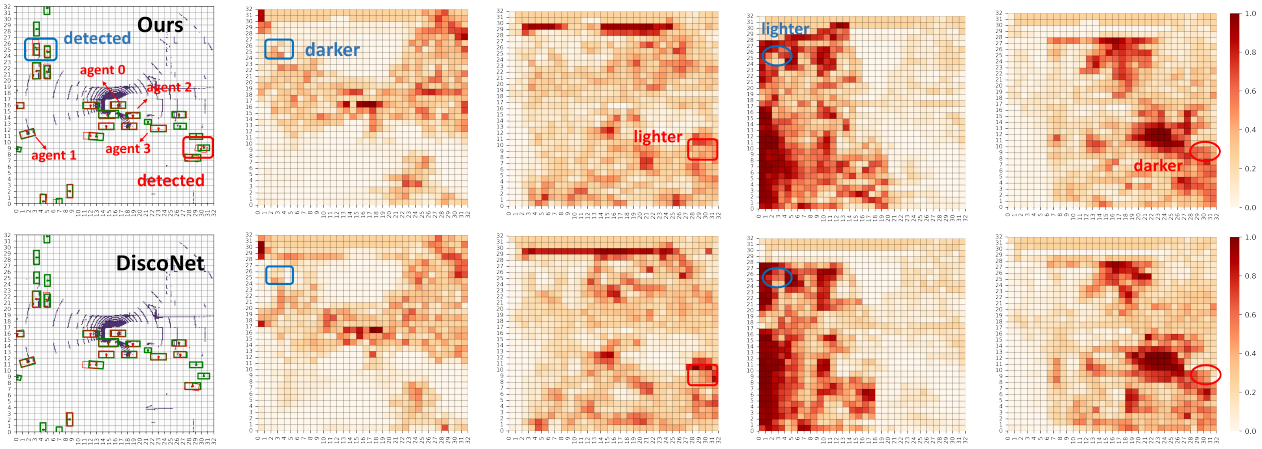
Fig. 4 further shows the performance-bandwidth trade-off of CMiMC. We employ compression techniques (details of compression technique are given in Appendix A.4) to reduce the size of intermediate features. We use 8 compress ratios $1/2^{n}, n = 1, 2, \ldots, 8$ and show the corresponding AP achieved by CMiMC. It can be observed that with compression ratio $1/32$, CMiMC still achieves a higher AP than the SOTA performance of DiscoNet. Particularly, even with the maximum compression ratio, i.e., $1/256$, CMiMC still outperforms When2com, Who2com, and Late Collab.

Visualization of detection results. Fig. 5 shows detection results generated by CMiMC, No Collab., and Early Collab. We see in Fig. 5a that No Collab. cannot perceive objects in



(a) Early Collab. (b) When2com (c) DiscoNet (d) CMiMC (ours) (e) $W_{0\to0}$ (CMiMC) (f) $W_{1\to0}$ (CMiMC)

Figure 6: Detection results of Intermediate Collab. methods (at agent 0). The feature weight matrices are visualized as heatmaps, where darker colors indicate higher weight in the corresponding regions and lighter colors indicate lower weight.



(a) Detection results (b) $W_{0\to0}$ (c) $W_{1\to0}$ (d) $W_{2\to0}$ (e) $W_{3\to0}$

Figure 7: Detection results and feature weight matrix of CMiMC and DiscoNet (at agent 0).

occluded and long-range areas. Fig. 5b shows that CMiMC can overcome these issues by leveraging Intermediate Collab. Further comparing it with Fig. 5c, we see that CMiMC provides perception results similar to Early Collab.

Attention mechanisms. Fig. 6 shows results of methods that involve attention mechanisms (i.e., CMiMC, When2com, and DiscoNet). We see that CMiMC detects more objects than When2com and DiscoNet. The performance inferiority of When2com is due to the utilization of scalar attention which cannot accurately capture the regional characteristics in views. DiscoNet also utilizes voxel-level attention weights, but its performance is bottlenecked by the Early Collab. teacher. As shown in Fig. 6a and 6c, objects not detected by Early Collab. are often missed by DiscoNet as well. In contrast, CMiMC can learn better attention weights (Fig. 6e, 6f), which better capture the importance of regional feature values at a voxel-level resolution.

Visualization of feature weight matrices. Fig. 7 compares the feature weight matrices and perception results of CMiMC and DiscoNet. The results indicate that CMiMC can identify critical regions more efficiently compared to DiscoNet. For example, for the two vehicles in the top left corner (detected successfully by CMiMC but missed by DiscoNet), CMiMC gives higher weight for the view of agent 0 as it captures discriminative information about these vehicles and decreases weight for the view of agent 2 as such information is not contained there.

Supplementary results. We provide supplementary experimental results in Appendix B, including hyper-parameter selection, time complexity, loss curves, and evaluation on more public datasets (DAIR-V2X (Yu et al. 2022), V2X-Sim 2.0 (Li et al. 2022), V2XSet (Xu et al. 2022) and V2V4Real (Xu et al. 2023b)).

Conclusion

This paper proposes an intermediate collaboration method called CMiMC, which presents a novel way of constructing good collaborative views through the maximization of mutual information. The core of CMiMC is CMiMNet which estimates and maximizes the multi-view mutual information between a collaborative view and multiple individual views. CMiMC enables the collaboration encoder to efficiently identify beneficial regional information in individual views and aggregate them properly into the collaborative view. Comprehensive experimental results demonstrate that CMiMC achieves an excellent trade-off between perception performance and communication bandwidth, and outperforms SOTA strategies. CMiMC is a generic framework and can be applied to various application scenarios, e.g., autonomous driving, robotics, and surveillance.

Acknowledgments

The work of Wanfang Su and Lixing Chen was partially supported by the National Natural Science Foundation of China (NSFC) under Grant 62202293 and 62372297. The work of Yang Bai was partially supported by NSFC under Grant 62303306. The work of Zhe Qu was partially supported by NSFC under Grant 62302525.

References

Arnold, E.; Dianati, M.; de Temple, R.; and Fallah, S. 2020. Cooperative perception for 3D object detection in driving scenarios using infrastructure sensors. *IEEE Transactions on Intelligent Transportation Systems*, 23(3): 1852–1864.

Bachman, P.; Hjelm, R. D.; and Buchwalter, W. 2019. Learning representations by maximizing mutual information across views. *Advances in Neural Information Processing Systems*, 32.

Belghazi, M. I.; Baratin, A.; Rajeshwar, S.; Ozair, S.; Bengio, Y.; Courville, A.; and Hjelm, D. 2018. Mutual information neural estimation. In *International Conference on Machine Learning*, 531–540. PMLR.

Chen, Q.; Tang, S.; Yang, Q.; and Fu, S. 2019. Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds. In *International Conference on Distributed Computing Systems*, 514–524. IEEE.

Chen, X.; Ma, H.; Wan, J.; Li, B.; and Xia, T. 2017. Multi-view 3d object detection network for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1907–1915.

Cui, J.; Qiu, H.; Chen, D.; Stone, P.; and Zhu, Y. 2022. Coopernaut: End-to-end driving with cooperative perception for networked vehicles. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17252–17262.

Darbellay, G. A.; and Vajda, I. 1999. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory*, 45(4): 1315–1321.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An open urban driving simulator. In *Annual Conference on Robot Learning*, 1–16. PMLR.

Hido, S.; Tsuboi, Y.; Kashima, H.; Sugiyama, M.; and Kanamori, T. 2011. Statistical outlier detection using direct density ratio estimation. *Knowledge and information systems*, 26: 309–336.

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2019. Learning deep representations by mutual information estimation and maximization. In *International Conference for Learning Representations*.

Hu, Y.; Fang, S.; Lei, Z.; Zhong, Y.; and Chen, S. 2022. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *Advances in Neural Information Processing Systems*, 35: 4874–4886.

Li, Y.; Ma, D.; An, Z.; Wang, Z.; Zhong, Y.; Chen, S.; and Feng, C. 2022. V2X-Sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving. *IEEE Robotics and Automation Letters*, 7(4): 10914–10921.

Li, Y.; Ren, S.; Wu, P.; Chen, S.; Feng, C.; and Zhang, W. 2021. Learning distilled collaboration graph for multi-agent perception. *Advances in Neural Information Processing Systems*, 34: 29541–29552.

Linsker, R. 1988. Self-organization in a perceptual network. *Computer*, 21(3): 105–117.

Liu, Y.-C.; Tian, J.; Glaser, N.; and Kira, Z. 2020a. When2com: Multi-agent perception via communication graph grouping. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4106–4115.

Liu, Y.-C.; Tian, J.; Ma, C.-Y.; Glaser, N.; Kuo, C.-W.; and Kira, Z. 2020b. Who2com: Collaborative perception via learnable handshake communication. In *IEEE International Conference on Robotics and Automation*, 6876–6883. IEEE.

Luo, G.; Zhang, H.; Yuan, Q.; and Li, J. 2022. Complementarity-enhanced and redundancy-minimized collaboration network for multi-agent perception. In *ACM International Conference on Multimedia*, 3578–3586.

Miller, A.; Rim, K.; Chopra, P.; Kelkar, P.; and Likhachev, M. 2020. Cooperative perception and localization for cooperative driving. In *IEEE International Conference on Robotics and Automation*, 1256–1262. IEEE.

Nowozin, S.; Cseke, B.; and Tomioka, R. 2016. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in Neural Information Processing Systems*, 29.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Poole, B.; Ozair, S.; Van Den Oord, A.; Alemi, A.; and Tucker, G. 2019. On variational bounds of mutual information. In *International Conference on Machine Learning*, 5171–5180. PMLR.

Sanghi, A. 2020. Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning. In *European Conference on Computer Vision*, 626–642. Springer.

Silverman, B. W. 1981. Using kernel density estimates to investigate multimodality. *Journal of the Royal Statistical Society: Series B (Methodological)*, 43(1): 97–99.

Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive multiview coding. In *European Conference on Computer Vision*, 776–794. Springer.

Wang, B.; Zhang, L.; Wang, Z.; Zhao, Y.; and Zhou, T. 2023. Core: Cooperative reconstruction for multi-agent perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8710–8720.

Wang, T.-H.; Manivasagam, S.; Liang, M.; Yang, B.; Zeng, W.; and Urtasun, R. 2020. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *European Conference on Computer Vision*, 605–621. Springer.

Xu, R.; Li, J.; Dong, X.; Yu, H.; and Ma, J. 2023a. Bridging the domain gap for multi-agent perception. In *International Conference on Robotics and Automation (ICRA)*, 6035–6042. IEEE.

Xu, R.; Xia, X.; Li, J.; Li, H.; Zhang, S.; Tu, Z.; Meng, Z.; Xiang, H.; Dong, X.; Song, R.; et al. 2023b. V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13712–13722.

Xu, R.; Xiang, H.; Tu, Z.; Xia, X.; Yang, M.-H.; and Ma, J. 2022. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *European Conference on Computer Vision*, 107–124. Springer.

Yu, H.; Luo, Y.; Shu, M.; Huo, Y.; Yang, Z.; Shi, Y.; Guo, Z.; Li, H.; Hu, X.; Yuan, J.; et al. 2022. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21361–21370.