MACP: Efficient Model Adaptation for Cooperative Perception

Yunsheng Ma[1]*, Juanwu Lu[1]*, Can Cui[1], Sicheng Zhao[2], Xu Cao[3,5], Wenqian Ye[4,5], Ziran Wang[1]
[1] Purdue University, West Lafayette, IN, USA
[2] Tsinghua University, Beijing, China
[3] University of Illinois Urbana-Champaign, Champaign, IL, USA
[4] University of Virginia, Charlottesville, VA, USA
[5] PediaMed AI, Shenzhen, China
{yunsheng,juanwu,cancui,ziran}@purdue.edu
schzhao@tsinghua.edu.cn, xucao2@illinois.edu, wenqian@virginia.edu

Abstract

Vehicle-to-vehicle (V2V) communications have greatly enhanced the perception capabilities of connected and automated vehicles (CAVs) by enabling information sharing to "see through the occlusions", resulting in significant performance improvements. However, developing and training complex multi-agent perception models from scratch can be expensive and unnecessary when existing single-agent models show remarkable generalization capabilities. In this paper, we propose a new framework termed MACP, which equips a single-agent pre-trained model with cooperation capabilities. We approach this objective by identifying the key challenges of shifting from single-agent to cooperative settings, adapting the model by freezing most of its parameters and adding a few lightweight modules. We demonstrate in our experiments that the proposed framework can effectively utilize cooperative observations and outperform other state-of-the-art approaches in both simulated and real-world cooperative perception benchmarks while requiring substantially fewer tunable parameters with reduced communication costs. Our ource code is available at https://github.com/PurdueDigitalTwin/MACP.

1. Introduction

Automated vehicles have made significant progress in recent years. However, their perception systems fall short in occlusions and long-range perception, hindering higher-level autonomy. Recently, cooperative perception systems through vehicle-to-vehicle (V2V) communications have emerged as a promising solution. This approach shifts the perception from a single-agent perspective into a joint task of the connected and automated vehicles (CAVs) system. It can address occlusion challenges and enhance the overall perception performance of individual vehicles [5]. Through collaborative information sharing, cooperative perception has the potential to revolutionize the automotive industry.

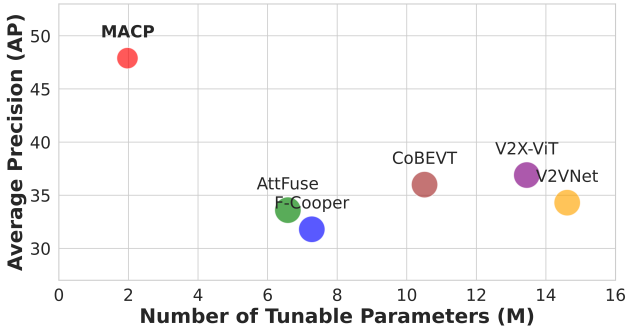


Figure 1. Performance comparison with state-of-the-art (SOTA) methods (F-Cooper [7], V2VNet [38], AttFuse [45], V2X-ViT [44], CoBEVT [41]) on the V2V4Real Dataset [42]: Bubble size corresponds to the transmitted data size required by the algorithms. Our proposed MACP model outperforms the leading SOTA model by achieving a **30%** improvement in Average Precision (AP) at Intersection over Union (IoU) = 70 while requiring only **15%** of the number of tunable parameters and **65%** of the volume of data transmission.

The latest V2V cooperative perception studies explicitly aim to design dedicated perception models for the new setting. These sophisticated perception models have shown impressive performance improvements [30, 41, 44] over their single-agent counterparts. Nevertheless, developing and training these large cooperative perception models can be expensive. Moreover, extensive data labeling for training in the cooperative perception context can be time-consuming and costly, setting back the scalability of these

*Equal Contribution

approaches [42, 54].

An intuitive way to reduce the cost is to take advantage of the generalization potential of existing single-agent perception models and adapt them to cooperative perception. Pre-trained single-agent models have demonstrated excellent transferability [9], raising the possibility of reusing their generalizable representations of the observations. Although promising, the feasibility of this adaptation process remains questionable, as inappropriate fine-tuning using downstream data could compromise the pre-trained model's performance. Therefore, our framework draws inspiration from the parameter-efficient fine-tuning (PEFT) strategy, initially explored in natural language processing [19]. The PEFT strategy aims to retain the pre-trained model's strength by only fine-tuning a small number of additional parameters while freezing the majority of the model.

This paper introduces a novel framework termed "efficient Model Adaption for Cooperative Perception," or MACP. Specifically, we bridge the gap between single-agent and cooperative perception by addressing domain shifts and communication bottlenecks. We design the Convolution Adapter (ConAda) for the feature encoder and communication channel and add Scale and Shift the Features (SSF) operations [25] in the prediction net to mitigate domain shifts. Our experiments show that our framework significantly outperforms previous state-of-the-art (SOTA), with substantially fewer tunable parameters. Moreover, the proposed MACP framework inherently supports compressed data transmission and can effectively utilize shared data. In summary, we make the following contributions:

- We identify the gap between the efficacy of single-agent perception models and the requirements of cooperative perception.
- We propose a novel framework to empower adapting single-agent perception models to the cooperative perception, which is simple to implement and cost-effective to train.
- The proposed model significantly outperforms previous SOTA methods on simulation and real-world cooperative perception benchmarks. Especially, a 30% improvement is achieved in Average Precision (AP) at Intersection over Union (IoU) = 70 on the V2V4Real dataset [42], with only 15% of the number of tunable parameters and 65% of data transmission size.

2. Related Work

2.1. 3D Object Detection

Accurate object perception is crucial for ensuring the safety of autonomous driving systems. SOTA 3D object detection models commonly use sparse convolutions to extract point cloud features [46]. These features are then combined with either anchor-based [10, 46] or center-based [48, 56] strategies for making predictions.

VoxelNet [55] encodes voxel features using PointNet [29], and then applies a region proposal network and a prediction head. SECOND [46] improves performance by integrating efficient sparse convolutions with an anchor-based prediction head. CenterPoint, inspired by CenterNet [13], converts the sparse output from a backbone network into a feature map and then predicts object center locations through heatmap generation. This center-based prediction strategy has been adopted by various models [1, 26, 56] to enhance the performance of 3D detection frameworks. However, single-agent 3D object detection models suffer from occlusions and long-range prediction. We tackle these through V2V cooperative perception.

2.2. Cooperative Perception

The fundamental idea of cooperative perception in CAVs is to enhance their field of view by sharing observations from surrounding vehicles or roadside infrastructures [5]. There are three categories of cooperation approaches based on their data-sharing strategies: (1) **Early Fusion** [8]: CAVs transmit raw sensor data, and the ego vehicle makes predictions based on the aggregated raw data, which incurs the highest data transfer cost; (2) **Late Fusion** [31, 33, 40, 49, 52]: Late fusion involves sharing and combining predictions (*e.g.*, 3D bounding boxes) from CAVs, reducing the data transfer load. However, the performance is dependent on the prediction precision of other CAVs and is sensitive to spatial and temporal misalignment introduced by issues like localization error and transmission delay; (3) **Mid Fusion** [23, 28, 30, 38, 39, 41, 44]: Mid-fusion involves collaborators extracting intermediate features, encoding them and then compressing and broadcasting them to other vehicles. This strategy offers higher flexibility and has the potential to balance performance, robustness, and communication costs, and is therefore gaining more attention. For instance, researchers have proposed vision transformers with attention modules specifically designed for cooperative perception in V2X-ViT [44] and CoBEVT [41]. Qiao et al. have also proposed adaptive feature fusion models with trainable neural networks in AdaFusion [30]. Note that all these models are specifically designed for the cooperative setting and are trained from scratch, which can be costly. This paper distinguishes itself by exploring efficient adaptations of pre-trained single-agent perception models to address these concerns.

2.3. Parameter-Efficient Fine-Tuning

Implementing a pre-trained model for a different task often requires a fine-tuning procedure based on the new dataset. Essentially, the goal is to preserve the knowledge

parameterized by the pre-trained model and adjust it to fit the application context. Previous studies have approached this objective from different perspectives, including multi-task learning with fixed upstream layers [3, 6, 50, 51], sequential life-long learning by adding new parameters [34], and preserving knowledge from the old task [21, 24, 53].

Recent studies have shown that fine-tuning can be achieved by updating or appending a relatively small number of parameters. Fewer tunable parameters generally are more energy efficient [11] and enable fast iterative prototyping and transferring. The Adapter [2, 19, 32, 47] and its applications are among the earliest parameter-efficient fine-tuning (PEFT) methods. Adapter [19] is inserted into transformer [37] layers, and it first projects and transforms features into a low-dimensional space and then projects them back to the source domain. Subsequent studies have extended this idea by using low-rank updates [20] or sparse parameter selection [16, 36]. SSF [25] is another PEFT method designed to scale and shift the features extracted by a pre-trained model. The idea behind SSF is to address the distribution mismatch between the upstream and downstream tasks. These SOTA PEFT methods have demonstrated the ability to significantly reduce the training cost of models while matching the performance of fully fine-tuning all the parameters. However, adapting existing single-agent perception models for cooperative perception contexts remains underexplored, a gap that this paper aims to address.

3. Methodology

3.1. Problem Statement

This paper focuses on cooperative 3D object detection through V2V communication. Given a traffic condition with N agents, each agent i has an observed point cloud $x_i \in \mathbb{R}^{N_i \times d}$ consisting of N_i points in 3D Euclidean space. We represent each point by a d-dimensional vector with attributes such as coordinates and reflected intensity. We denote the set of all point clouds by $\mathbf{X} = \{x_i\}_{i=1,\dots,N}$. The objective is to solve for an optimal model f^* capable of detecting and delineating bounding boxes about surrounding objects and assigning appropriate labels. To simplify our notations, we represent each bounding box and its class label by a d'-dimensional vector $y_j \in \mathbb{R}^{d'}$. Without losing generality, an object detection model f is a mapping from point cloud space to the joint space of bounding boxes and their labels $f : \mathcal{X} \to \mathcal{Y}$ and the trained model ideally describes the probability of observing bounding box set $\mathbf{y}$ conditioned on observed point cloud set $\mathbf{x}$, given by

$$p(\mathbf{y}|\mathbf{x}; f) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})}. \tag{1}$$

To adapt a pre-trained single-agent perception model for the cooperative perception setting, we need to deal with distribution shifts in observed point clouds. Specifically, if we denote the marginal probability of observing a point cloud set in single-agent perception by $p_{\mathcal{S}}(\mathbf{x})$ and the probability of observing the exact set in a cooperative perception by $p_{\mathcal{C}}(\mathbf{x})$, the two probabilities can be different due to additional point clouds shared by V2V communication, that is, $p_{\mathcal{S}}(\mathbf{x}) \neq p_{\mathcal{C}}(\mathbf{x})$. The joint distribution of point clouds and bounding boxes given by the pre-trained model deviates from the ground-truth joint distribution under the cooperative setting:

$$\hat{p}_{\mathcal{C}}(\mathbf{x}, \mathbf{y}; f) = \frac{p_{\mathcal{S}}(\mathbf{x}, \mathbf{y})}{p_{\mathcal{S}}(\mathbf{x})} \cdot p_{\mathcal{C}}(\mathbf{x}) \neq p_{\mathcal{C}}(\mathbf{x}, \mathbf{y}). \tag{2}$$

Existing literature [22, 35] refers to this phenomenon as domain shifts and has shown that it directly leads to performance degradation. In this paper, we propose to introduce a collection of light-weight fine-tuning module g and transform $p(\mathbf{y}|\mathbf{x}; f)$ such that $p(\mathbf{y}|\mathbf{x}; f \cdot g) = g\left[\frac{p_{\mathcal{S}}(\mathbf{x},\mathbf{y})}{p_{\mathcal{S}}(\mathbf{x})}\right]$ and

$$g^* = \underset{g \in \mathcal{G}}{\operatorname{argmin}} \mathcal{L}\left(p_{\mathcal{C}}(\mathbf{x}, \mathbf{y}), \hat{p}_{\mathcal{C}}(\mathbf{x}, \mathbf{y}; f \cdot g)\right), \tag{3}$$

where $\mathcal{L}$ is a loss function measuring the distance between two distributions.

Meanwhile, cooperative perception introduces new challenges in terms of computation and communication. The model must handle compressed data with minimal performance loss to ensure stable data transmission and responsive decision-making in high-traffic environments. This paper presents the MACP framework with specific design and implementation of PEFT modules, following guidelines to account for domain shifts and communication bottlenecks while maximizing performance with minimal trainable parameters. The following sections introduce two PEFT modules, ConAda and SSF Operators, followed by elaborations on how we address the guidelines in their designs and implementations.

3.2. MACP Overview

As illustrated in Fig. 2, the proposed MACP is a decentralized framework where each vehicle encodes point cloud features locally using a Feature Encoder network with ConAda modules. After local feature encoding, the vehicles communicate with each other via a compression-decompression channel driven by another ConAda module. We provide further information about the ConAda module in Sec. 3.3. The collected features are fused with the feature from the local Feature Encoder, and then passed through a Prediction Net with SSF modules (refer to Sec. 3.4) to generate bounding box predictions. Note that in our settings, all vehicle models share the same parameters.

3.3. Convolution Adapter

Mainstream 3D object detection models commonly rely on convolutional layers to capture local feature correlations

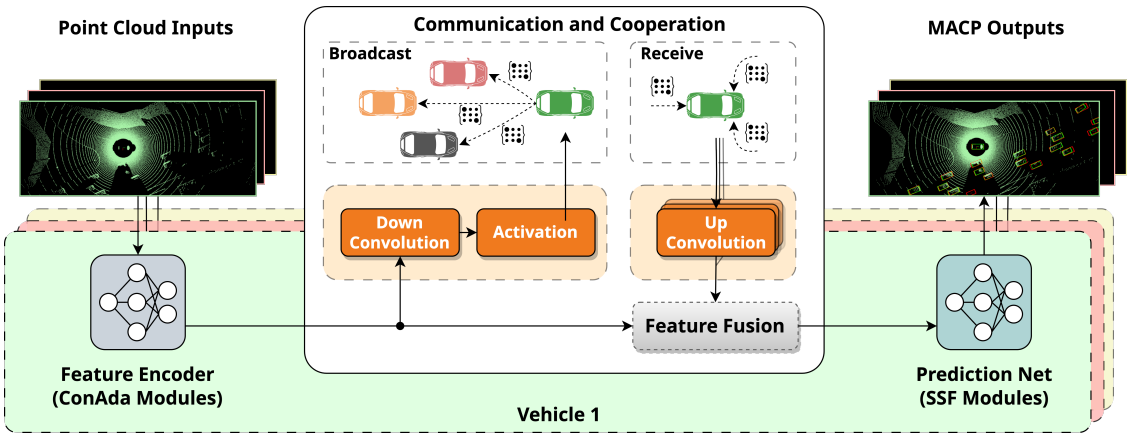


Figure 2. **Overview of the MACP Framework.** The down convolution and the activation layers compress the output from the feature encoder to enable efficient communication with surrounding vehicles. Meanwhile, each vehicle receives features from its surroundings and decompresses the information using the up convolution layer in a stack of ConAda modules. A feature fusion module fuses the decompressed data with the encoded feature from the upstream feature encoder and passes it to the downstream prediction network.

effectively. To adapt the pre-trained single-agent parameters to the cooperative setting, we propose the Convolution Adapter (ConAda) module. As shown in Fig. 3, ConAda consists of a down convolution layer that first projects the source feature map to a lower dimension. Following this projection, a non-linear activation and a subsequent up convolution layer transform and remap the lower-dimension feature back to the source space. Given an input feature map $\mathbf{I}$, the output feature map of ConAda is calculated as follows

$$\mathbf{O} = \text{Conv}(\text{Activation}(\text{Conv}(\mathbf{I}, \mathbf{K}^{\text{down}})), \mathbf{K}^{\text{up}}), \quad (4)$$

where $\mathbf{K}^{\text{down}} \in \mathbb{R}^{D \times D'}$ and $\mathbf{K}^{\text{up}} \in \mathbb{R}^{D' \times D}$ are the down-projection and up-projection kernels respectively, and their dimensions satisfy $D' < D$. Activation$(\cdot)$ is the non-linear activation function, and Conv$(\cdot, \cdot)$ is the convolution operation with both kernel size and strides equal to 1.

Note that while visual data like images are inherently dense, 3D point clouds derived from LiDAR are intrinsically sparse. Applying dense convolutional operations on such data is computationally inefficient [14]. As a result, 3D object detectors commonly use sparse convolutions to extract point cloud features (see Sec. 2.1 for detailed descriptions). To this end, our proposed ConAda module implementation inherently rests on sparse convolution operations. Specifically, we use the Submanifold sparse convolution [15] within the ConAda module. Specifically, we consider a sparse voxelized input $X^{\text{input}} = \{(v_1, f_1), (v_2, f_2), \ldots, (v_N, f_N)\}$, where v_i is the voxel's coordinates and $f_i \in C$ is the associated feature vector. Given the kernel $\mathbf{K} \in \mathbb{R}^{C \times C'}$, for an *occupied* voxel v_i with feature f_i, the output feature vector after the convolution is

$$X^{\text{output}}_{i,c'} = \sum_{k=1}^{C} \mathbf{K}_{k,c'} \times f_{i,k}, \quad (5)$$

where $X^{\text{output}}_{i,c'}$ is the c'-th element of the output feature vector for voxel v_i.

ConAda modules are key components for the Feature Encoder. A Feature Encoder network is a cascade of convolution blocks where the output from the convolution layer passes a ConAda module and adds back to itself with a residual connection [17]. We only train the ConAda parameters during training and freeze the pre-trained parameters in the convolution layer and other layers following the ConAda module.

Meanwhile, ConAda also acts as the communication channel between vehicles. During communications, the down convolution and the activation layer in the ConAda module help compress and encrypt the encoded feature for broadcasting, while the up convolution layer serves to decompress the received signal for feature fusion.

3.4. SSF Operator for Fused Feature

The output from the point encoder module is a latent feature map representing a mixture of point clouds observed by ego and surrounding vehicles. We implement the SSF [25] operator in the consecutive neural network layers to account for the domain shift. Suppose the output feature map from the convolution layer is given by $X^{\text{output}} \in \mathbb{R}^{H' \times W' \times C'}$, we update the feature map by using a scaling factor $\gamma \in \mathbb{R}^{C'}$ and a shifting factor $\beta \in \mathbb{R}^{C'}$, given as

$$X^{\text{output}}_{i,j} = \gamma \odot X^{\text{output}}_{i,j} + \beta, \quad (6)$$

where $\odot$ is the Hadamard product. The SSF operator only scales and shifts the feature without altering their positional identities.

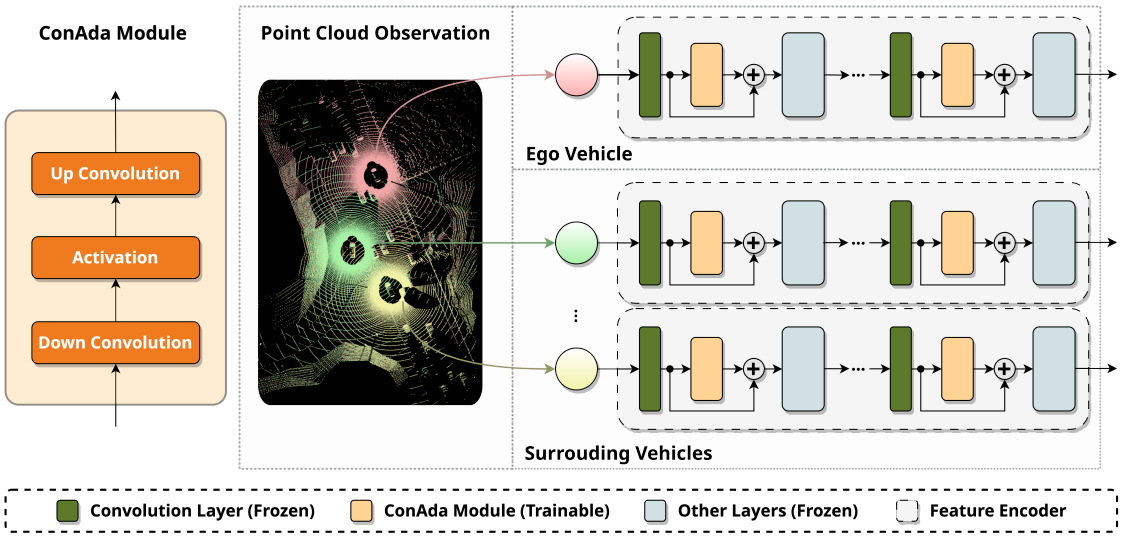


Figure 3. **Illustration of the Distributed Feature Encoder with ConAda module.** Ego and surrounding vehicles each encode their observations with a feature encoder consisting of a cascade of blocks. All feature encoders share the same parameters. In each block, a ConAda module (yellow) processes the output feature map from the pre-trained convolution layer, adds it back to the convolution output through residual connection, and passes it to the consecutive other layers.

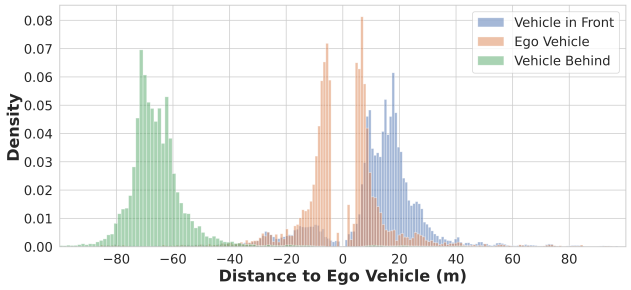


Figure 4. **Example of spatial distribution shifts of point clouds.** On the x-axis, the distance's sign is determined by the angle between the ego vehicle's direction and the line from it to the point. The result indicates a significant difference in the data distribution from the surrounding vehicles from that of the ego vehicle.

3.5. Insights

As mentioned in Sec. 3.1, our MACP framework and the two PEFT modules account for domain shifts while considering computation and communication constraints. We focus on covariate shifts of point cloud features for domain shifts and identify two primary sources: spatial distribution shifts and feature space shifts.

Spatial distribution originates from point clouds collected from different vehicles concentrating around their respective locations and differing in scales due to varying sensors, as shown in Fig. 4. A simple concatenation of these point clouds leads to a shift in the spatial distribution of the data from a bi-modal to a multi-modal mixture distribution. However, since the pre-trained model was trained on the bi-modal or even an uni-modal point cloud distribution, the parameters may not recognize the importance of out-of-distribution point cloud features. Our MACP addresses this issue through the distributed framework. In this way, we enforce each vehicle to encode point cloud features from their local perspectives, which aligns more with the single-agent perception cases and can help eliminate impacts from the out-of-distribution problem.

Feature space shift is manifested as each latent feature map element may contain more or less information compared to the single-agent perception. This requires extra operators to scale and shift the feature maps to align with the latent space from single-agent perception before the direct use of pre-trained parameters. ConAda modules in the MACP framework approach this by projecting and applying non-linear transformation in a low-dimensional space, while the SSF modules directly apply scales and shifts to the input feature maps.

Finally, the ConAda-based communication channel mitigates the communication bottlenecks by allowing flexible compression of the signal to transmit. We will investigate and show in our experiments how this compression-decompression process affects the performance.

4. Experiments and Results

4.1. Experimental Settings

Datasets We conducted comprehensive experiments on two widely used cooperative perception benchmarks: the V2V4Real [42] and the OPV2V [43, 45] datasets. Both

Method	Param (M)		AP@IoU=50/70 (↑)				AM (↓)
	Total	Trainable	Overall	0-30m	30-50m	50-100m	(MB)
No Fusion	6.58	6.58	39.8/22.0	69.2/42.6	29.3/14.4	4.8/1.6	0
Late Fusion	6.58	6.58	55.0/26.7	73.5/36.8	43.7/22.2	36.2/17.3	0.003
Early Fusion	6.58	6.58	59.7/32.1	76.1/46.3	42.5/20.8	47.6/21.1	0.96
F-Cooper [7]	7.27	7.27	60.7/31.8	80.8/46.9	45.6/23.6	32.8/13.4	0.20
V2VNet [38]	14.61	14.61	64.5/34.3	80.6/51.4	52.6/26.6	42.6/14.6	0.20
AttFuse [45]	6.58	6.58	64.7/33.6	79.8/44.1	53.1/29.3	43.6/19.3	0.20
V2X-ViT [44]	13.45	13.45	64.9/36.9	82.0/55.3	51.7/26.6	43.2/16.2	0.20
CoBEVT [41]	10.51	10.51	66.5/36.0	82.3/51.1	52.1/28.2	**49.1**/19.5	0.20
MACP **(Ours)**	8.94	1.97	**67.6/47.9**	**83.7/62.1**	**58.4/38.5**	34.6/**23.1**	0.13

Table 1. **Performance Comparison on the V2V4Real Dataset**. The best-performing method is highlighted in **bold**, while the second-best method is indicated by an underline. ↓: Lower values are better. ↑: Higher values are better.

Method	Param (M)		AP@IoU=50/70 (↑)	
	Total	Trainable	Default Towns	Culver City
No Fusion	6.58	6.58	67.9/60.2	55.7/47.1
Late Fusion	6.58	6.58	85.8/78.1	79.9/66.8
Early Fusion	6.58	6.58	89.1/80.0	82.9/69.6
F-Cooper [7]	7.27	7.27	88.7/79.0	84.6/72.8
V2VNet [38]	14.61	14.61	89.7/82.2	86.0/73.4
AttFuse [45]	6.58	6.58	90.8/81.5	85.4/73.5
V2X-ViT [44]	13.45	13.45	89.1/82.6	87.3/73.7
CoBEVT [41]	10.51	10.51	91.4/86.1	85.9/77.2
AdaFusion [30]	7.27	7.27	91.6/85.6	88.1/79.0
MACP **(Ours)**	8.98	2.00	**93.7/90.3**	**91.4/80.7**

Table 2. **Performance Comparison on the OPV2V Dataset.**

datasets support the 3D object detection task where different types of vehicles are considered as the same category, and the detection targets are vehicles. Each CAV comes with its individual LiDAR point cloud and corresponding ground-truth 3D bounding boxes.

OPV2V is a simulation-based dataset that includes two subsets: Default Towns (DT) and Culver City (CC). The DT subset contains data from eight default towns provided by CARLA [12]. Each frame contains an average of about 3 CAVs, with a minimum of 2 and a maximum of 7. It has an official split for training, validation, and testing with 6.7K, 2K, and 2.7K frames, respectively. The CC scenes form a separate test set of 550 frames, evaluating the model's generalization capability in unseen scenes. *V2V4Real* is a real-world dataset collected by two vehicles driving simultaneously in Columbus, Ohio, USA. It is officially split into the train, validation, and test sets with 14K, 2K, and 4K frames, respectively.

Evaluation For fair comparisons, we use the same settings as in previous studies. Specifically, for the OPV2V dataset, we set the evaluation range in the x and y directions to (-140m, 140m) and (-40m, 40m), respectively, relative to the ego vehicle. For the V2V4Real dataset, the evaluation range in the x and y directions is set to (-100m, 100m) and (-40m, 40m), respectively, with reference to the ego vehicle. We evaluate the detection performance using Average Precision (AP) at Intersection-over-Union (IoU) 0.5 and 0.7 as the metric. Following [42], we use the Average MegaByte (AM) metric to quantify the volume of transmitted data of algorithms on the V2V4Real dataset. All models are evaluated under the *Sync* setting, where data transmission is considered instantaneous [42, 49].

Implementation We use the *LiDAR-only BEVFusion* [26] as the single-agent perception model, which was pre-trained on the nuScenes [4] dataset. The ConAda modules utilize GELU [18] as the activation function. Optimization is performed using AdamW [27] with a weight decay of 10^{-2}. For more details, please refer to the Appendix.

4.2. Comparison with the State of the art

In this section, we compare the proposed MACP method with previous studies, including baseline models such as no fusion, late fusion, early fusion, and several SOTA mid-fusion models (see Sec. 2.2).

4.2.1 Results on V2V4Real

Tab. 1 presents performance comparisons on V2V4Real. Our method achieves superior performance by tuning only 1.97M parameters, which is much less than previous models, thanks to the knowledge successfully transferred from the pre-trained single-agent model. The data transmission size is 0.13 AM, which is 35% lower compared to other mid-fusion methods and 87% lower compared to the early fusion baseline. This size is implemented with a compression factor of 256 (see Eq. (7)). Despite the high compression factor, our method still achieves an overall AP score of 47.9 at IoU=70, which is a 30% improvement over the runner-up method V2X-ViT [44]. These results suggest that efficient finetuning can successfully handle cooperation.

Method	Param (M)		AP@IoU=50/70 (↑)			
	Total	Trainable	Overall	0-30m	30-50m	50-100m
Full Fine-Tune	8.92	8.92	68.3/**51.9**	**85.0/65.7**	**59.1/41.2**	33.0/**26.9**
Train Fusion & Head Only	8.92	1.94	65.5/42.1	83.1/57.4	51.9/30.0	33.5/19.9
Adapter Only [19]	9.17	2.19	63.8/37.5	81.0/50.8	49.1/26.5	33.7/17.6
SSF Only [25]	8.92	1.95	64.7/44.2	82.7/60.2	50.8/30.9	32.6/21.0
ConAda Only	8.97	2.00	67.5/49.0	84.1/62.2	57.0/38.6	34.1/25.6
MACP	8.98	2.00	**69.4**/49.6	83.2/63.1	58.6/40.0	**38.5**/26.5

Table 3. **Effectiveness of proposed components.**

In addition, we observed that MACP performs exceptionally well in high-precision predictions (IoU=70). It outperforms other methods across all object distance ranges: 0-30m, 30-50m, and 50-100m, with margins of 12%, 31%, and 9% over other leading SOTA methods. When considering predictions with lower precision (IoU=50), our method continues to outperform its counterparts in the 0-30m and 30-50m ranges and the overall average precision (AP) metric. However, it is worth noting that our model's AP for objects in the 50-100m range is 30% lower compared to the best-performing method CoBEVT [41]. This discrepancy is due to biases in the pre-trained model and will be analyzed in detail in Sec. 4.3.

4.2.2 Results on OPV2V

As shown in Tab. 2, our method with 2M tunable parameters outperforms all prior methods. Specifically, in the Default Towns where the models are trained and tested, our approach shows a performance increase of 2.3% and 4.8% in terms of AP at IoU=50/70, respectively, compared to other methods. It is worth noting that the pre-trained model is trained on real-world data while the new domain is of simulated data. This suggests that our MACP method is able to bridge the gap not only between the single-agent and cooperative settings but also between real-world and simulation domains. Furthermore, in the Culver City scenes containing previously unseen environments during training, our model continues to outperform the leading SOTA method AdaFusion [30], demonstrating improvements of 3.7% and 2.1% in terms of AP at IoU=50/70, respectively. These results show our model's great generalization ability.

4.3. Module Effectiveness

Since maximizing performance with minimal trainable parameters is one of our primary goals, this experiment aims to understand each proposed module's role in the overall performance and compare their parameter efficiency. We set up two baseline models: a fully fine-tuned BEVFusion model and another that only trains the fusion and bounding-box prediction heads. The second model is essentially the proposed framework but without any of the proposed PEFT

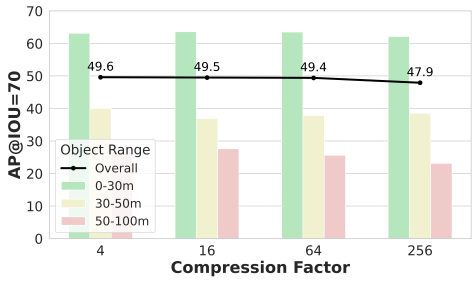


Figure 5. **Effect of compression factors.** The visualization shows how the ConAda compression impacts the overall AP (black line) and AP of bounding boxes in different ranges (color bars).

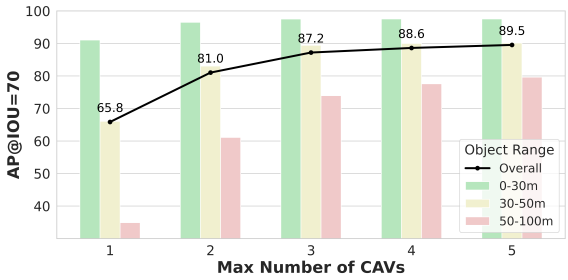
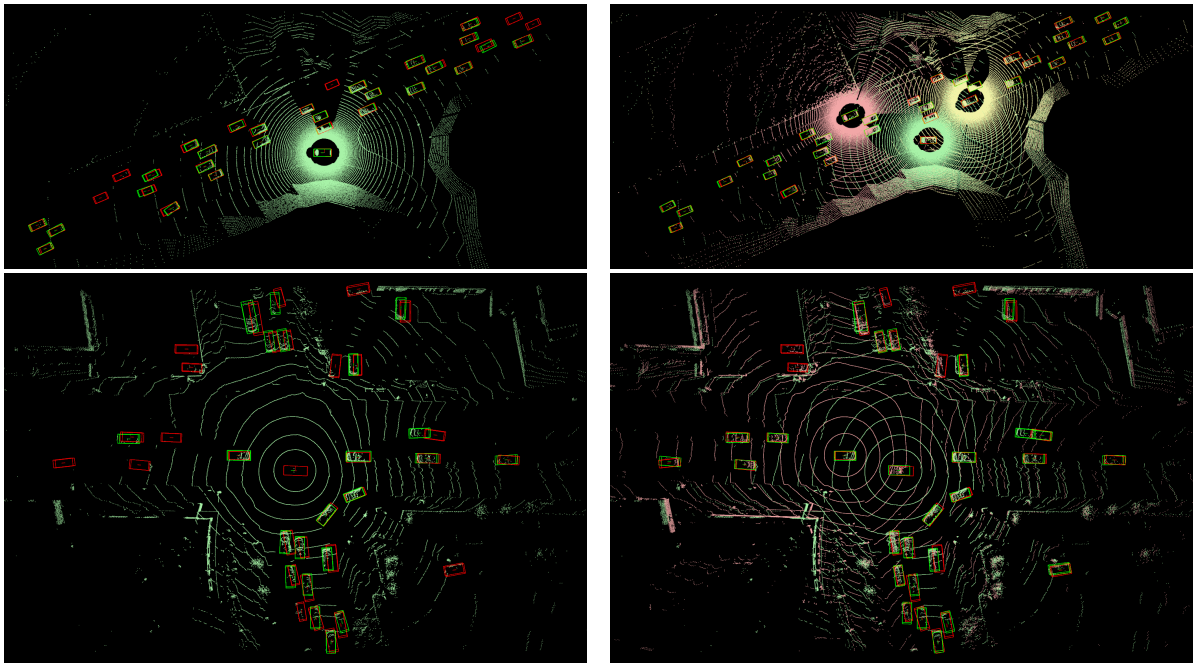


Figure 6. **Effect of max number of CAVs.** The upward trend of AP shows that the additional information from cooperative perception is being used effectively.

modules. We compare the proposed MACP framework with three of its variants. The Adapter Only variant does not use the ConAda module in the feature encoder and replaces the SSF module in the Prediction Net with Adapter [19] modules. The SSF Only variant removes the ConAda modules from the feature encoder, and the ConAda Only variant removes the SSF modules from the Prediction Net of the proposed framework, respectively. All models are set with a compression factor of 4. Tab. 3 details the total and tunable parameters and the overall and range-specific bounding box prediction accuracy.

We observe that the proposed MACP achieves comparable performance to the fully fine-tuned BEVFusion model, but with only around one-fifth of its total number of trainable parameters, indicating promising parameter efficiency. The ConAda Only variant outperforms the other two, sug-



(a) BEVFusion [26] (Single-Agent Perception)

(b) MACP (Cooperative Perception)

Figure 7. Comparison between BEVFusion (Single-Agent Perception) and our MACP (Cooperative Perception) on the OPV2V (Top) and V2V4Real (Bottom) datasets. The point cloud of the ego vehicle is shown in light green, while the point clouds of other vehicles are represented in different colors. The 3D bounding boxes in red and green represent the ground-truth and predicted objects, respectively. Our MACP outperforms BEVFusion in detecting occluded or distant objects.

gesting that our original ConAda structure proposal plays a more significant role in the final performance. Ultimately, MACP benefits from both ConAda and SSF modules to achieve the best results.

Furthermore, it is worth noting that the fully fine-tuned model performs poorly when predicting distant objects (i.e., those in the range of 50-100 meters). This confirms that the poor long-distance prediction precision is due to the pre-trained parameters mentioned in Sec. 4.2.2. In fact, our proposed MACP reduces the negative impact of the pre-trained bias and outperforms the fully fine-tuned model in long-distance predictions.

4.4. Compression Robustness

To test whether our MACP model can handle compression data with minimum performance loss in real-world use cases, we examine the performance sensitivity against the compression factor on the V2V4Real Dataset. The compression factor is used in the cooperation ConAda

$$\text{Compression Factor} = C_{\text{in}}/C_{\text{out}}, \tag{7}$$

where C_{in} and C_{out} are the number of input and output feature map channels of the down convolution. As shown in Fig. 5, the proposed MACP manages to maintain its performance and shows promising robustness against different compression rates.

4.5. Cooperation Effectiveness

While the MACP framework has shown promising performance, there is concern that it may not be able to utilize the extra information from V2V communication effectively. To test this, we conduct an experiment using OPV2V to examine whether increasing the number of CAVs can improve the model's performance. As shown in Fig. 6, the results indicate a clear positive correlation between prediction precision and the maximum number of CAV observations, demonstrating that our model can effectively make use of the extra information collected from the surrounding vehicles. Fig. 7 also provides visualization results supporting this conclusion.

5. Conclusion

In this work, we proposed a novel framework to adapt single-agent models for cooperative perception efficiently. We addressed key challenges including domain shifts, computation, and communication constraints in real-world V2V applications. We have achieved superior performance in simulation-based and real-world cooperative perception benchmarks with high parameter efficiency and lower communication costs. Our limitation is the assumption of ideal communication and localization. We'll cover transmission delay and other real-world factors in the future.

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. TransFusion: Robust LiDAR-Camera Fusion for 3D Object Detection With Transformers. In *CVPR*, pages 1090–1099, 2022. 2

[2] Ankur Bapna and Orhan Firat. Simple, Scalable Adaptation for Neural Machine Translation. In *EMNLP*, pages 1538–1548, 2019. 3

[3] Hakan Bilen and Andrea Vedaldi. Integrated perception with recurrent multi-task neural networks. In *NeurIPS*, volume 29, 2016. 3

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *CVPR*, pages 11621–11631, 2020. 6

[5] Antoine Caillot, Safa Ouerghi, Pascal Vasseur, Remi Boutteau, and Yohan Dupuis. Survey on Cooperative Perception in an Automotive Context. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14204–14223, 2022. 1, 2

[6] Rich Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, July 1997. 3

[7] Qi Chen, Xu Ma, Sihai Tang, Jingda Guo, Qing Yang, and Song Fu. F-cooper: feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds. *ACM/IEEE Symposium on Edge Computing*, pages 88–100, 2019. 1, 6

[8] Qi Chen, Sihai Tang, Qing Yang, and Song Fu. Cooper: Cooperative Perception for Connected Autonomous Vehicles Based on 3D Point Clouds. *IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 514–524, 2019. 2

[9] Can Cui, Yunsheng Ma, Juanwu Lu, and Ziran Wang. Radar Enlighten the Dark: Enhancing Low-Visibility Perception for Automated Vehicles with Camera-Radar Fusion. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2023. 2

[10] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection. In *AAAI*, volume 35, pages 1201–1209, 2021. 2

[11] Radosvet Desislavov, Fernando Martínez-Plumed, and José Hernández-Orallo. Trends in AI inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. *Sustainable Computing: Informatics and Systems*, 38:100857, Apr. 2023. 3

[12] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *CoRL*, volume 78, pages 1–16, 2017. 6

[13] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. CenterNet: Keypoint Triplets for Object Detection. In *ICCV*, pages 6569–6578, 2019. 2

[14] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D Semantic Segmentation With Submanifold Sparse Convolutional Networks. In *CVPR*, pages 9224–9232, 2018. 4

[15] Benjamin Graham and Laurens van der Maaten. Submanifold Sparse Convolutional Networks, June 2017. arXiv:1706.01307 [cs.NE]. 4

[16] Demi Guo, Alexander Rush, and Yoon Kim. Parameter-Efficient Transfer Learning with Diff Pruning. In *ACL*, pages 4884–4896, 2021. 3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 770–778, 2016. 4

[18] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2016. arXiv:1606.08415 [cs.LG]. 6

[19] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. In *ICML*, volume 97, June 2019. 2, 3, 7

[20] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*, 2022. 3

[21] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, Mar. 2017. 3

[22] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In *ICML*, pages 5637–5664, 2021. 3

[23] Yiming Li, Shunli Ren, Pengxiang Wu, Siheng Chen, Chen Feng, and Wenjun Zhang. Learning Distilled Collaboration Graph for Multi-Agent Perception. In *NeurIPS*, volume 34, pages 29541–29552, 2021. 2

[24] Zhizhong Li, Derek Hoiem, Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Learning without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, Dec. 2018. 3

[25] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & Shifting Your Features: A New Baseline for Efficient Model Tuning. In *NeurIPS*, volume 35, pages 109–123, Dec. 2022. 2, 3, 4, 7

[26] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. In *ICRA*, 2023. 2, 6, 8

[27] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019. 6

[28] Ehsan Emad Marvasti, Arash Raftari, Amir Emad Marvasti, Yaser P. Fallah, Rui Guo, and Hongsheng Lu. Cooperative

LIDAR Object Detection via Feature Sharing in Deep Networks. In *VTC*, pages 1–7, 2020. 2

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NeurIPS*, volume 30, 2017. 2

[30] Donghao Qiao and Farhana Zulkernine. Adaptive Feature Fusion for Cooperative Perception Using LiDAR Point Clouds. In *WACV*, pages 1186–1195, 2023. 1, 2, 6, 7

[31] Andreas Rauch, Felix Klanner, Ralph Rasshofer, and Klaus Dietmayer. Car2X-based perception in a high-level fusion architecture for cooperative perception systems. In *IEEE IV*, pages 270–275, 2012. 2

[32] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *NeurIPS*, volume 30, 2017. 3

[33] Matthias Rockl, Thomas Strang, and Matthias Kranz. V2V Communications in Automotive Multi-Sensor Multi-Target Tracking. In *VTC*, pages 1–5, 2008. 2

[34] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks, Dec. 2018. arXiv:1606.04671 [cs]. 3

[35] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate Shift Adaptation by Importance Weighted Cross Validation. *The Journal of Machine Learning Research*, 8:985–1005, 2007. 3

[36] Yi-Lin Sung, Varun Nair, and Colin A Raffel. Training Neural Networks with Fixed Sparse Masks. In *NeurIPS*, volume 34, pages 24193–24205, 2021. 3

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, 2017. 3

[38] Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. V2VNet: Vehicle-to-Vehicle Communication for Joint Perception and Prediction. In *ECCV*, pages 605–621, 2020. 1, 2, 6

[39] Hao Xiang, Runsheng Xu, and Jiaqi Ma. HM-ViT: Heteromodal Vehicle-to-Vehicle Cooperative perception with vision transformer, Apr. 2023. arXiv:2304.10628 [cs]. 2

[40] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation. In *CVPR*, pages 244–253, 2018. 2

[41] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. CoBEVT: Cooperative Bird's Eye View Semantic Segmentation with Sparse Transformers. In *CoRL*, 2022. 1, 2, 6, 7

[42] Runsheng Xu, Xin Xia, Jinlong Li, Hanzhao Li, Shuo Zhang, Zhengzhong Tu, Zonglin Meng, Hao Xiang, Xiaoyu Dong, Rui Song, Hongkai Yu, Bolei Zhou, and Jiaqi Ma. V2V4Real: A Real-world Large-scale Dataset for Vehicle-to-Vehicle Cooperative Perception. In *CVPR*, 2023. 1, 2, 5, 6

[43] Runsheng Xu, Hao Xiang, Xu Han, Xin Xia, Zonglin Meng, Chia-Ju Chen, Camila Correa-Jullian, and Jiaqi Ma. The OpenCDA Open-Source Ecosystem for Cooperative Driving Automation Research. *IEEE Transactions on Intelligent Vehicles*, 8(4):2698–2711, Apr. 2023. 5

[44] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2X-ViT: Vehicle-to-Everything Cooperative Perception with Vision Transformer. In *ECCV*, pages 107–124, Aug. 2022. 1, 2, 6

[45] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. OPV2V: An Open Benchmark Dataset and Fusion Pipeline for Perception with Vehicle-to-Vehicle Communication. In *ICRA*, pages 2583–2589, 2022. 1, 5, 6

[46] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 18(10):3337, Oct. 2018. 2

[47] Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. AIM: Adapting Image Models for Efficient Video Action Recognition. In *ICLR*, 2023. 3

[48] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-Based 3D Object Detection and Tracking. In *CVPR*, pages 11784–11793, 2021. 2

[49] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, and Zaiqing Nie. DAIR-V2X: A Large-Scale Dataset for Vehicle-Infrastructure Cooperative 3D Object Detection. In *CVPR*, pages 21361–21370, 2022. 2, 6

[50] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust Visual Tracking via Structured Multi-Task Sparse Learning. *IJCV*, 101(2):367–383, 2013. 3

[51] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial Landmark Detection by Deep Multi-task Learning. In *ECCV*, pages 94–108, 2014. 3

[52] Zijian Zhang, Shuai Wang, Yuncong Hong, Liangkai Zhou, and Qi Hao. Distributed Dynamic Map Fusion via Federated Learning for Intelligent Networked Vehicles. In *ICRA*, pages 953–959, 2021. 2

[53] Sicheng Zhao, Yunsheng Ma, Yang Gu, Jufeng Yang, Tengfei Xing, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. An End-to-End Visual-Audio Attention Network for Emotion Recognition in User-Generated Videos. In *AAAI*, volume 34, pages 303–311, 2020. 3

[54] Sicheng Zhao, Yezhen Wang, Bo Li, Bichen Wu, Yang Gao, Pengfei Xu, Trevor Darrell, and Kurt Keutzer. ePointDA: An End-to-End Simulation-to-Real Domain Adaptation Framework for LiDAR Point Cloud Segmentation. In *AAAI*, volume 35, pages 3500–3509, 2021. 2

[55] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *CVPR*, pages 4490–4499, 2018. 2

[56] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. CenterFormer: Center-Based Transformer for 3D Object Detection. In *ECCV*, pages 496–513, 2022. 2