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

Bridging the Domain Gap for Multi-Agent Perception

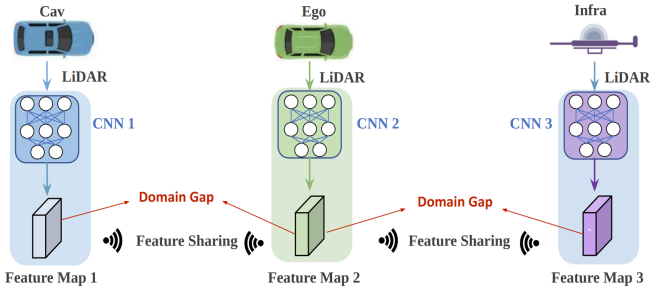
Runsheng Xu[1], Jinlong Li[2], Xiaoyu Dong[3], Hongkai Yu[2], Jiaqi Ma[1*]

***Abstract*— Existing multi-agent perception algorithms usually select to share deep neural features extracted from raw sensing data between agents, achieving a trade-off between accuracy and communication bandwidth limit. However, these methods assume all agents have identical neural networks, which might not be practical in the real world. The transmitted features can have a large domain gap when the models differ, leading to a dramatic performance drop in multi-agent perception. In this paper, we propose the first lightweight framework to bridge such domain gaps for multi-agent perception, which can be a plug-in module for most of the existing systems while maintaining confidentiality. Our framework consists of a learnable feature resizer to align features in multiple dimensions and a sparse cross-domain transformer for domain adaption. Extensive experiments on the public multi-agent perception dataset V2XSet have demonstrated that our method can effectively bridge the gap for features from different domains and outperform other baseline methods significantly by at least 8% for point-cloud-based 3D object detection.**

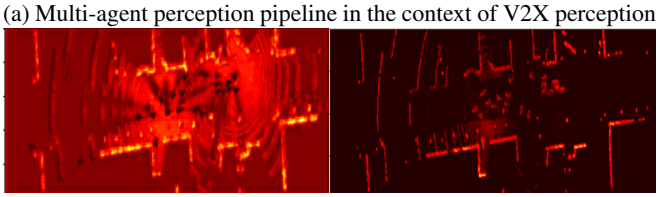
I. INTRODUCTION

Recent studies have demonstrated that by leveraging Vehicle-to-Everything (V2X) communication technology to share visual information, the multi-agent perception system can significantly improve the performance of the single-agent system by seeing through occlusions and perceiving longer range [1], [2], [3], [4], [5], [6], [7], [8]. Instead of sharing raw sensing data or detected outputs, state-of-the-art methods usually share the intermediate neural features computed from the sensor data, as they can achieve the best trade-off between accuracy and bandwidth requirements [9], [1]. Furthermore, transmitted intermediate features are more robust to the GPS noise and communication delay [3], [10], [8]. Despite the advancements in intermediate fusion strategy, previous methods conduct experiments under a strong assumption that all agents are equipped with identical neural networks to extract neural features. This overlooks a critical fact: deploying the same model for all agents is unrealistic, especially for connected autonomous driving [11], [12]. For example, as shown in (a) from Fig. 1, the detection models on connected automated vehicles (CAV) and infrastructure products of distinct companies are usually dissimilar. Even for the same company, diverse detection models may exist due to the different on-vehicle software versions. When the shared features come from different backbones, a noticeable domain gap exists, which can easily diminish the benefits of collaborations.

[1]University of California, Los Angeles, UCLA Mobility Lab. {rxx3386, jiaqima}@ucla.edu
[2]Cleveland State University, Cleveland Vision & AI Lab. j.li56@vikes.csuohio.edu, h.yu19@csuohio.edu
[3]Northwestern University.
*Corresponding Author



(a) Multi-agent perception pipeline in the context of V2X perception



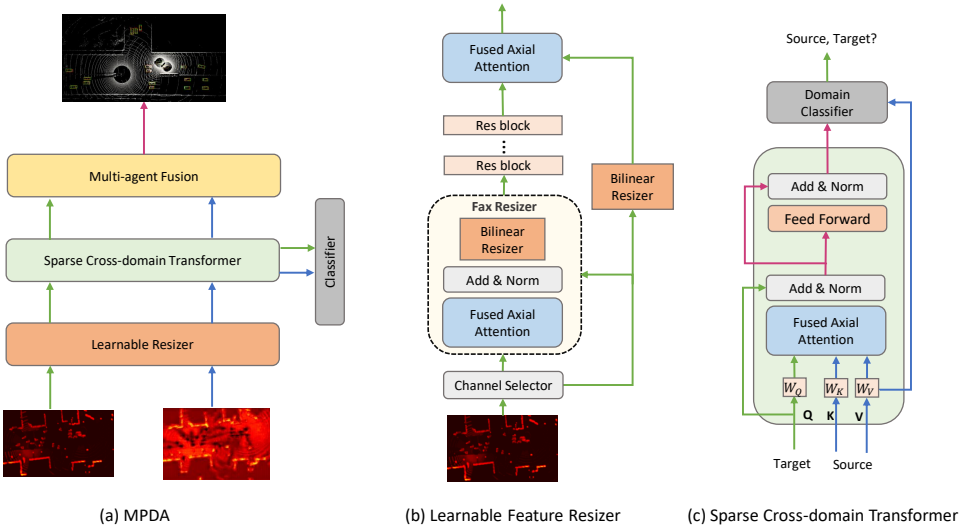
(b) PointPillar feature map (c) VoxelNet feature map

Fig. 1: **Illustration of domain gap of different feature maps for multi-agent perception.**. Here we use V2X cooperative perception in autonomous driving as an example. (a) Ego vehicle receives the shared feature maps from other CAV and infrastructure with different CNN models, which causes domain gaps. (b) Visualization of feature map from ego, which is extracted from PointPillar [13]. (c) Feature map from CAV, which is extracted from VoxelNet [14]. Brighter pixels represent higher feature values.

In this paper, we dive into this unsolved and practical problem in multi-agent perception, especially for autonomous driving. We first carefully investigate the domain gap of different feature maps and then propose our framework based on the analysis. Fig. 1 shows intermediate feature representations obtained from two distinct point cloud based 3D object detection backbones, PointPillar [13] and VoxelNet [14], in the same scenario. We apply the same techniques as [15] to make the visualization informative by summing up all channels' absolute value together. In general, we can observe the features are dissimilar in three aspects:

- **Spatial resolution.** Because of the different voxelization parameters, LiDAR cropping range, and downsampling layers, the spatial resolutions are different.
- **Channel number.** The channel dimensions are distinct due to the difference in convolution layers' settings.
- **Patterns.** As Fig. 1 shows, PointPillar and VoxelNet have the opposite patterns: The object positions have relatively low values on the feature map for PointPillar but high values for VoxelNet.

To address the three dominant distinctions, we present



(a) MPDA (b) Learnable Feature Resizer (c) Sparse Cross-domain Transformer

Fig. 2: **The overview and core components of our framework.** Our MPDA first aligns feature dimensions through a learnable feature resizer and then unifies the pattern through the sparse cross-domain transformer.

the first **M**ulti-agent **P**erception **D**omain **A**daption framework, dubbed as MPDA, to bridge the domain gap. Fig. 2 depicts the overall architecture. Specifically, two components, namely Learnable Resizer and Sparse Cross-Domain Transformer, are proposed. As multiple factors could cause different spatial resolutions, we argue that using rudimentary resizing algorithms such as bilinear and nearest interpolation may cause severe misalignment. Therefore, we propose to resize the received intermediate features in a learnable way and optimize with the multi-agent fusion algorithms jointly to improve the detection performance. Moreover, aligning the channel dimension by simply dropping channels can potentially lead to losing important information; thus, our resizer also includes a learnable channel selector to alleviate such loss. To diminish the pattern disparity, the sparse cross-domain transformer will efficiently reason the received and ego features locally and globally and generates domain-invariant representations by adversarially fooling a domain classifier. Finally, the state-of-the-art multi-agent fusion algorithm V2X-ViT [3] is utilized to fuse information across multiple agents. Since the framework does not require any key information from other models (e.g., model type, parameters), it can maintain confidentiality. We conduct extensive experiments on the public dataset V2XSet [3], and the result demonstrates that our framework can increase the accuracy of V2X-ViT by at least 8% under various realistic settings. Overall, our contributions are summarized as follows:

- We pioneer the domain gap identification (spatial resolution, channel number, pattern) in multi-agent perception and propose a new Multi-agent Perception Domain Adaption (MPDA) framework, which is the **first work to bridge the domain gap for multi-agent perception**.
- We present a novel Learnable Resizer to better align spatial and channel features from other agents in an adaptive way.
- We propose a sparse cross-domain transformer that can efficiently unify the feature patterns from various agents.
- The proposed MPDA framework can be easily combined with other multi-agent fusion algorithms and does not require confidential model information from other agents. Extensive experiments on the public dataset V2XSet demonstrate that our method achieves the best performance with real-time performance.

II. Related Work

Multi-Agent Perception: Despite the great progress in autonomous driving in the past years, there still exist many challenges that single-vehicle systems can not get over [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]. Especially, the single-agent perception systems suffer severely from occlusions and limitations in sensor range [26]. Multi-agent perception was born to alleviate such pains. As a pioneering work, V2VNet [9] first proposes the intermediate fusion approach, in which all agents should broadcast the features extracted from the raw point cloud to achieve the trade-off between bandwidth requirement and accuracy. Following this design ethos, OVP2V [1] uses a single-head self-attention module to fuse the received intermediate features. DiscoNet [5] utilizes a graph neural network and knowledge distillation to aggregate the shared representations. Recently, V2X-ViT [3] first proposed to use a vision transformer for multi-agent perception and achieves robust performance under GPS error and communication delay. [27] captures both aleatoric and epistemic uncertainties with one inference pass and tailors a moving block bootstrap algorithm with direct modeling of the multivariant Gaussian distribution of each corner of the bounding box. The method can be used with different collaborative object detectors and helps to improve safety-critical systems such as CAVs. Despite the prominent performance achieved by these methods, none of them consider the realistic domain gap issue caused by the

model discrepancy. We aim to fill such a gap in this paper.

Transformers in Vision: Since Dosovitskiy *et al.* [28] successfully adapt the Transformer architecture [29] into the computer vision area by regarding image patches as visual words, Vision Transformers (ViT) have gained increasing attention [30], [31], [32]. For example, [33] shows that by applying a 3D attention mechanism, the object detection performance can outperform traditional convolutional neural networks. Despite obtaining great benefits from global interactions, the full attention in ViT [29], [28] usually requires large computation resources. To avoid such costs, recent methods [34], [35], [36] have explored different sparse attention mechanisms such as local and sparsely global schemes. In this work, we adapt an efficient 3D attention called Fused Axial Attention (FAX) in CoBEVT [2] to our domain adaption framework as it already shows excellent efficiency on multi-agent fusion.

Domain Adaptation: Due to the time consumption of data annotation and the domain gap between different domains, domain adaptation is utilized to solve these problems by adapting the model trained on a labeled source domain to address an unlabeled target domain. Recent works on domain adaptation mainly address different computer vision tasks [37], [38], [39], [40], [41], [42], [43], [44], [45], [46]. In domain adaptation, to minimize the domain shift between different domains, feature distribution can be aligned in common levels: domain level [47], [48], [49] and category level [42], [39], [50], [51], [52]. Domain level alignment generally involves minimizing some measure of distance between the source and target feature distributions like maximum mean discrepancy [47]. [53] proposes a novel prototype-based shared-dummy classifier (PSDC) model to address the challenges of open-set domain adaptation, including distinguishing between unknown target instances and shared classes and aligning shared class prototypes, which outperforms existing methods on several datasets. [54] proposes a novel class-Balanced Multicentric Dynamic prototype (BMD) strategy for the Source-free Domain Adaptation (SFDA) task to adapt pre-trained source models to the target domain without accessing the well-labeled source data. The proposed BMD strategy avoids the gradual dominance of easy-transfer classes on prototype generation, introduces a novel inter-class balanced sampling strategy, and incorporates dynamic network update information during model adaptation. While category level alignment aligned each category distribution between source and target domain using an adversarial manner between the feature extractor and domain classifiers. A fine-grained alignment leads to more accurate distribution alignment in the same label space. Xu et al. [42] adopted Transformers for category-level domain adaptation to show great potential in image classification. In this paper, similar to [42], a sparse cross-domain Transformer is proposed to unify the feature patterns from different agents.

Learnable Resizer: [55] first comes up the concept of learnable resizer for image classifications. Instead of using rudimentary interpolation, they employ a convolution neural network to resize the RGB images for classification and jointly train with vision models. Our learnable feature resizer is inspired by this work but differs in three major aspects: 1) We investigate an unexplored practical application scenario for a learnable resizer – domain adaption for multi-agent perception. 2) Our resizing target is the LiDAR feature, which is more sparse than images. Therefore, instead of using a pure convolution neural network, we integrate our resizer with a sparse transformer. 3) Besides resizing the spatial dimension, we also embed a simple but effective algorithm to resize the channel dimension to the required number.

III. Methodology

In this paper, we consider a realistic scenario for multi-agent perception, where each agent in the collaboration may be equipped with a separate model and transmit visual features with domain discrepancy. We mainly focus on the cooperative perception task of LiDAR-based 3D object detection for autonomous driving, where the agents are connected to autonomous vehicles and intelligent roadside infrastructure, but our framework is generally-applicable to other multi-agent perception applications as long as they broadcast neural features for collaborations. Since we focus on the problem of domain gaps in this work, we assume the relative poses between agents are accurate and no communication delay exists.

Fig. 2(a) shows the overall architecture of our MPDA, which consists of 1) a learnable feature resizer, 2) a sparse cross-domain transformer, 3) a domain classifier, and 4) multi-agent feature fusion. In this section, we will describe the details of each module.

A. Learnable Feature Resizer

We regard the feature maps computed locally on ego vehicle as source domain features $F_S \in \mathbb{R}^{1\times H_S\times W_S\times C_S}$ and received features from other agents as target domain features $F_T \in \mathbb{R}^{N\times H_T\times W_T\times C_T}$, where N is the number of other collaborators/agents, H is the height, W is the width, C is the channel number, and $H_S \neq H_T, W_S \neq W_T, C_S \neq C_T$. The goal of our feature resizer Φ is to align the dimensions of the source domain feature with the target domain in a learnable way:

$$F_T^{'} = \Phi(F_T),\ \text{s.t. } F_T^{'} \in \mathbb{R}^{N\times H_S\times W_S\times C_S}. \tag{1}$$

We jointly train Φ with multi-agent detection models so it can intelligently learn the optimal approach to resize the features, which is fundamentally different from the naive resizing method such as bilinear interpolation. The architecture of our learnable feature resizer is designed as Fig 2(b) shows, which includes four major components: channel aligner, FAX resizer, skip connection, and res-block.

Channel Aligner: We use a simple 1×1 convolution layer to align the channel dimension, whose input channel number is $C_{in} = 2C_S$ and outputs C_S channels. When $C_T > C_{in}$, we randomly drop $C_{in} - C_T$ channels and apply the 1×1 convolution layer to obtain a new feature. We repeat this process on F_T for n times to get features with

$n \times H_T \times W_T \times C_S$ dimensions and average them along the first dimension. In this way, we ameliorate the loss of information due to channel dropping. When $C_T < C_{in}$, we perform padding with randomly selected channels from F_T to meet the required input channel number for the 1×1 convolution.

FAX Resizer: To search for the optimal resizing solution, the neural network is supposed to have a large receptive field to gain the global information and pay attention to details to capture the critical object information. Since LiDAR features are usually sparse due to empty voxels, applying large-kernel convolution to get global information may diffuse the meaningless information to the important area. Therefore, we apply the fused axial (FAX) attention block [2] before bilinear resizing to fetch better feature representations. FAX sparsely employs local window and grid attention to efficiently capture global and local interactions. More importantly, it can discard empty voxels through a dynamic attention mechanism to eliminate their potential negative effects. After FAX, a bilinear resizer is implemented to reshape the feature map to the same spatial dimension as the source feature map. Compared to simple bilinear interpolation, our FAX resizer can adjust the input features first to avoid misalignment and distortion issues during resizing.

Skip connection: We also employ the bilinear feature resizing method in the skip connection to make learning easier.

Res-Block: We implement standard residual blocks [56] r times after resizing the feature maps to further refine them.

B. Sparse Cross-Domain Transformer

After retrieving the resized feature $F_T^{'}$, we need to convert its pattern to be indistinguishable from the domain classifier to obtain the domain-invariant features. To reach this goal, we need to effectively reason the correlations between $F_T^{'}$ and F_S both locally and globally. Therefore, we propose the sparse cross-domain transformer, which enjoys the benefits of dynamic and global attention brought by the transformer architecture while avoiding expensive computation. Fig. 2(c) shows the details of our proposed architecture. We first apply different convolution layers W_Q, W_K, W_V on $F_T^{'}$ and F_S to obtain query, key, and value, respectively. Then the query from the target domain and key/value from the source domain will be fed into the FAX block, capturing sparsely local and global spatial interactions across target and source domain features. Finally, a standard feed-forward neural network (FFN) is implemented to refine the interacted feature further. The whole process can be formulated as below:

$$Q = W_Q(F_T^{'}), \quad K = W_K(F_S), \quad V = W_V(F_S), \tag{2}$$

$$\hat{F_T^{'}} = Q + LN(FAX(Q, K, V)), \tag{3}$$

$$F_T^{''} = \hat{F_T^{'}} + LN(FFN(\hat{F_T^{'}})), \tag{4}$$

where LN is layer normalization, Q is the query, K is the key, and V is the value. Afterward, we pair $F_T^{''}$ and F_S together and send them the domain classifier and multi-agent fusion module.

C. Domain Classifier

We use the H-divergence [57] to measure the divergence between $F_T^{''}$ and F_S. Let us denote X as a feature map that may come from the source or target domain and $h : X \to \{0, 1\}$ a domain classifier, which tries to predict source domain sample X_S as 0 and target domain sample X_T as 1. In our paper, the domain classifier comprises two convolution layers. Suppose H is the hypothesis space for the domain classifier and G is the combination of our learnable resizer and sparse cross-domain transformer, then G needs to be optimized towards the following objective:

$$\max_{G} \min_{h \in H} \left(\mathbf{E}_S(h(X)) + \mathbf{E}_T(h(X))\right) \tag{5}$$

where $\mathbf{E}_S(h(X))$ and $\mathbf{E}_T(h(X))$ are the domain classification error over the source domain and target domain respectively and X is produced by G. This optimization can be achieved in an adversarial training manner by a gradient reverse layer (GRL) [58].

D. Multi-Agent Fusion

Our MPDA framework is very flexible and can integrate most of the multi-agent fusion algorithms. In this work, we select a state-of-the-art model, V2X-ViT [3], as our multi-agent fusion algorithm. V2X-ViT employs a heterogeneous multi-agent self-attention block and a multi-scale windowed attention block sequentially to intelligently fuse the different agents' features. To achieve the best performance, besides learning to fool the domain classifier, G also targets to directly optimize the detection performance. Let us denote M as the multi-agent fusion algorithm, then the second training objective for G is:

$$\min_{G,M} \left(\mathbf{E}_D(V)\right), \quad V = M(F_S, F_T^{''}), \tag{6}$$

where $\mathbf{E}_D(V)$ is the 3D detection error and V is the fused feature with shape of $1 \times H_S \times W_S \times C_S$.

E. Loss

For 3D object detection, we use the smooth L1 loss for bounding box regression and focal loss [59] for classification. For the domain classifier, we utilize cross-entropy loss to learn domain-invariant features. The final loss is the combination of detection and domain adaptation loss:

$$L = \alpha L_{det} + \beta L_{domain}, \tag{7}$$

where α and β are the balance coefficients within range $[0, 1]$.

IV. Experiments

A. Dataset

We conduct experiments on the public large-scale V2X perception dataset V2XSet [3]. V2XSet is collected together by the high-fidelity simulator CARLA [60] and cooperative driving automation frame [61]. It provides LiDAR data from different autonomous vehicles and roadside intelligent infrastructure at the same timestamp and scenario. In total, V2XSet has 11,447 frames and can be split into 6,694/1,920/2,833 frames for training/validation/testing respectively.

B. Experiments Setup

Evaluation metrics. We evaluate the performance of our proposed framework by the final 3D detection accuracy. Similar to previous works in this area [1], [3], we set the evaluation range as $x \in [-140, 140]$ meters, $y \in [-40, 40]$ meters and measure the accuracy with Average Precisions (AP) at Intersection-over-Union (IoU) threshold of 0.7.

Evaluation protocols During training, we randomly select one agent as the ego agent. During testing, we choose a fixed one as the ego for each scenario. We estimate our model under three distinct settings:

1) *Normal scenario:* In this scenario, all ego agents and other agents use PointPillar [13] with identical parameter as the detection backbone, named p_0. **Among all experiments, the ego vehicle will always have p_0 as the backbone.**
2) *Hetero scenario 1:* During training, ego agents use PointPillar p_0, whereas other agents employ p_1, which also belongs to the PointPillar family but with heterogeneous configurations including voxelization resolutions and the number of convolution layers. To assess the generalization probability of the proposed MPDA, another trained PointPillar model p_2 will be used for testing, which has different parameters from any training model.
3) *Hetero scenario 2:* We assume even the model types are heterogeneous in this scenario. All ego vehicles are still trained based on p_0, and other agents are based on the different detection model SECOND [62] s_0. During the testing stage, we use another trained SECOND model s_1 with distinct parameters with s_0.

To ensure all the backbones are trained properly, we first assume that all agents are equipped with the same backbone and combine it with the V2X-ViT model [3] to perform 3D detection. As shown in Table I, all backbones have achieved reasonable accuracy. We also demonstrate partial parameters of various backbones in Tabel II, and there are noticeable differences between them in terms of voxel resolution, LiDAR cropping range, and the number of convolutional layers.

TABLE I: **Detection backbone models' performance on the testing set without domain gap.** We assume all agents have the same detection model in this experiment.

	p_0	p_1	p_2	s_0	s_1
AP@0.7	71.2	68.3	70.1	74.5	77.0

TABLE II: **Parameters of different detection backbone.**

Backbone	Voxel Resolution	Half Lidar Cropping Range (x&y)	# of 2D&3D CNN Layers
p_0	0.4, 0.4, 4	140.8 & 38.4	19 & 0
p_1	0.8, 0.6, 4	140.8 & 38.4	16 & 0
p_2	0.6, 0.6, 4	153.6 & 38.4	17 & 0
s_0	0.2, 0.2, 0.2	140.8 & 41.6	12 & 12
s_1	0.1, 0.1, 0.1	140.8 & 41.6	13 & 13

Compared methods: We consider *No Fusion* as the baseline, which does not involve any collaboration in the system. To demonstrate the significant effect of the domain gap, we first directly use the pre-trained model provided by [3] and simply apply bilinear interpolation with the channel dropping technique to align the dimensions. We then let the pre-trained model finetune on *Hetero1* and *Hetero2* scenarios to make the comparison fair since our framework will see features from different domains in training. To show the effectiveness of the two critical components in our framework, we first only add the learnable resizer. Then we add the sparse cross-domain transformer as well to be our complete framework, MPDA. We will also compare with *Late Fusion* method, which directly transmits the detected 3d bounding box along with the confidence score and merges all the overlapped predictions according to the sorted confidence scores. Though *Late Fusion* does not have the domain gap issue like intermediate fusion, it still suffers from the confidence score discrepancy issue, e.g., different models can have diverse confidence estimation biases.

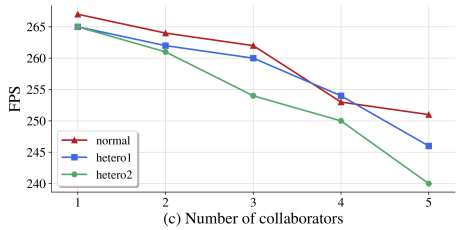


Fig. 3: **Inference speed of MPDA under different settings.**

Implementation details: For the multi-agent fusion method, we follow the same hyperparameters for V2X-ViT as its original implementation in [3]. For all backbones training, we use Adam [63] as the optimizer, decay the learning rate by 0.1 for every 10 epochs with an initial learning rate of 0.001. The coefficient of detection loss L_{det} is set to 1.0 and that of domain classification loss L_{domain} is set to 0.1.

TABLE III: **3D detection performance in Normal scenario (w/o domain gap) and Hetero scenarios (w/ domain gap).** We show the Average Precision (AP) at IoU=0.7. DC stands for domain classifier. * notes that we do not use the domain classifier when training on the normal scenario.

Method	Normal	Hetero 1	Hetero 2
No Fusion	40.2	40.2	40.2
Late Fusion	60.2	51.7	52.8
V2X-ViT	71.2	<u>26.7</u>	<u>34.5</u>
V2X-ViT (finetuned)	71.2	48.6	64.8
V2X-ViT + Resizer	72.3	54.8	72.1
V2X-ViT + MPDA (w/o DC)	73.4	56.3	72.5
V2X-ViT + MPDA	**73.4***	**57.6**	**73.3**

C. Quantitaive Evaluation

Major performance analysis: Table III depicts the performance comparison of various methods on *Normal*, *Hetro1*, and *Hetero2* settings, respectively. Under the *Normal* scenario, all methods exceed the baseline *No Fusion* by a

large margin. Nevertheless, the results are different when the models deployed on the agents are heterogeneous. The pre-trained V2X-ViT drops to 26.7% and 34.5% on *Hetero1* and *Hetero2* respectively, which is even much lower than the single agent perception system. **This dramatic performance drop indicates highly negative impacts by the domain gap.** After directly finetune on *Hetero1* and *Hetero2*, V2X-ViT's performance increases though still not satisfying and lower than *Late Fusion* in *Hetero1*. On the contrary, our MPDA has achieved 57.6%, and 73.3% on the two heterogeneous settings, which performs favorably against other methods and significantly outperforms *No Fusion*'s baseline. Note that the performance on *Hetero1* is relatively lower for all methods. A potential reason for it is p_2's voxel resolution, and the LiDAR cropping range is quite distinct from p_0 and p_1, which makes the adaption challenging. With the deployment of our framework, the accuracy of V2X-ViT increases by 9% and 8.5% on *Hetero1* and *Hetero2* respectively. We also found that our MPDA can enhance the performance on *Normal* setting as well by 2.2%, which attributes the capability of our resizer and sparse cross-domain transformer to help generate more robust feature representations.

Main component analysis: As Table III describes, all of our designed components in MPDA have contributed to more accurate detections. Adding the learnable resizer improves the detection performance by 6.2% and 7.3% under two heterogeneous settings. The sparse cross-domain transformer combined with the domain classifier can further increase the AP by 2.8% and 1.2%.

Inference time: Real-time performance is critical for real-world deployment. Thus, here we calculate the inference speed of our proposed MPDA framework under different scenarios concerning various collaborator numbers. As Fig. 3 shows, our MPDA can always achieve more than 200FPS under different settings, indicating our design's efficiency.

D. Qualitative Evaluation

Domain adaption visualization: Similar to [15], we sum up all the absolute values of all channels to visualize the feature maps to investigate their patterns. As Fig. 4 shows, without any domain adaption, there are noticeable gaps between the ego agent's and other collaborators' features. After applying our MPDA, the converted features become more similar to the ego's, which visually proves the effectiveness of MPDA.

3D detection visualization: We visually compare different methods in the same scenario under two heterogeneous settings and show the result in Fig. 5. Obviously, without seeing any features from different backbones, the pre-trained V2X-ViT model provided by the authors from [3] has many missing detections. After directly finetuning under *Hetero1* and *Hetero2* settings, the results get improved, but there still exist noticeable missing detections, false positives, and large displacement. On the contrary, our MPDA has a more robust performance, detecting most of the objects and predicting accurate bounding box positions.

V. CONCLUSIONS

This paper is the first work that investigates the domain gap issue in multi-agent perception. Based on the analysis, we propose the first multi-agent perception domain adaption framework, which mainly contains a learnable feature resizer and a sparse cross-domain transformer. Extensive experiments on the V2XSet dataset prove that our framework can effectively bridge the domain gap. In the future, we will combine robust generative representation learning techniques such as Diffusion [64] and conduct real-world field experiments on this practical issue.

REFERENCES

[1] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, "Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2583–2589.

[2] R. Xu, Z. Tu, H. Xiang, W. Shao, B. Zhou, and J. Ma, "Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers," *arXiv preprint arXiv:2207.02202*, 2022.

[3] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, "V2x-vit: Vehicle-to-everything cooperative perception with vision transformer," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[4] M. Hua, G. Chen, B. Zhang, and Y. Huang, "A hierarchical energy efficiency optimization control strategy for distributed drive electric vehicles," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 233, no. 3, pp. 605–621, 2019.

[5] Y. Li, S. Ren, P. Wu, S. Chen, C. Feng, and W. Zhang, "Learning distilled collaboration graph for multi-agent perception," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 541–29 552, 2021.

[6] G. Chen, X. Zhao, Z. Gao, and M. Hua, "Dynamic drifting control for general path tracking of autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, 2023.

[7] Y. Yuan, H. Cheng, and M. Sester, "Keypoints-based deep feature fusion for cooperative vehicle detection of autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3054–3061, 2022.

[8] Z. Lei, S. Ren, Y. Hu, W. Zhang, and S. Chen, "Latency-aware collaborative perception," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[9] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction," in *ECCV*. Springer, 2020, pp. 605–621.

[10] W. Liu, X. Xia, L. Xiong, Y. Lu, L. Gao, and Z. Yu, "Automated vehicle sideslip angle estimation considering signal measurement characteristic," *IEEE Sensors Journal*, vol. 21, no. 19, pp. 21 675–21 687, 2021.

[11] R. Song, L. Zhou, V. Lakshminarasimhan, A. Festag, and A. Knoll, "Federated learning framework coping with hierarchical heterogeneity in cooperative its," *arXiv preprint arXiv:2204.00215*, 2022.

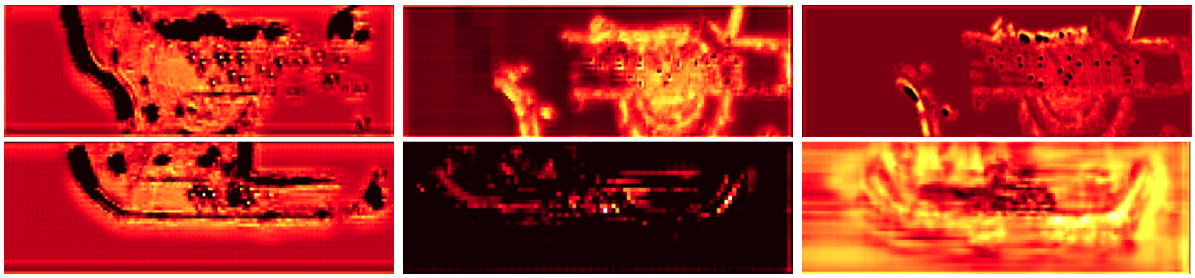
[12] R. Song, A. Hegde, N. Senel, A. Knoll, and A. Festag, "Edge-aided sensor data sharing in vehicular communication networks," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, 2022, pp. 1–7.

[13] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[14] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *CVPR*, 2018, pp. 4490–4499.

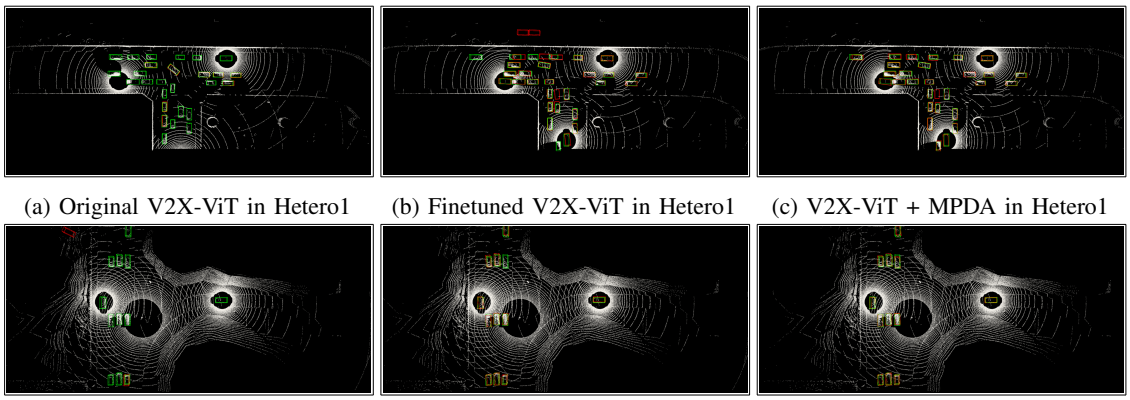
[15] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," *arXiv preprint arXiv:1612.03928*, 2016.

[16] J. Thunberg, G. Sidorenko, K. Sjöberg, and A. Vinel, "Efficiently bounding the probabilities of vehicle collision at intelligent intersections," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 2, pp. 47–59, 2021.



(a) ego's feature (b) collaborator's feature before MPDA (c) collaborator's feature after MPDA

Fig. 4: **Visalzation of intermediate features before and after domain adaption.** From left to right: (a) ego's feature, (b) collaborator's feature before domain adaption, (c) collaborator's feature after domain adaption. Row 1 is the *Hetero1* scenario where ego and others both use PointPillar, but the parameters differ. Row 2 is the *Hetero2* scenario where ego uses PointPillar, and others use SECOND. It is obvious that after domain adaption, others' intermediate features have more similar patterns as ego's.



(a) Original V2X-ViT in Hetero1 (b) Finetuned V2X-ViT in Hetero1 (c) V2X-ViT + MPDA in Hetero1

(d) Original V2X-ViT in Hetero2 (e) Finetuned V2X-ViT in Hetero2 (f) V2X-ViT + MPDA in Hetero2

Fig. 5: **3D detection visualization.** Green and red 3D bounding boxes represent the ground truth and prediction respectively. With our MPDA, the detection results are clearly more accurate.

[17] H. Xie, Y. Wang, X. Su, S. Wang, and L. Wang, "Safe driving model based on v2v vehicle communication," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 449–457, 2022.

[18] Q. Han, Y.-S. Zhou, Y.-X. Tang, X.-G. Tuo, and P. He, "Event-triggered finite-time sliding mode control for leader-following second-order nonlinear multi-agent systems," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 570–579, 2022.

[19] S. E. Shladover, "Opportunities and challenges in cooperative road vehicle automation," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 2, pp. 216–224, 2021.

[20] R. Valiente, B. Toghi, R. Pedarsani, and Y. P. Fallah, "Robustness and adaptability of reinforcement learning-based cooperative autonomous driving in mixed-autonomy traffic," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 397–410, 2022.

[21] S. C. Calvert and G. Mecacci, "A conceptual control system description of cooperative and automated driving in mixed urban traffic with meaningful human control for design and evaluation," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 1, pp. 147–158, 2020.

[22] R. A. Shet and S. Yao, "Cooperative driving in mixed traffic: An infrastructure-assisted approach," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 2, pp. 429–447, 2021.

[23] A. Ferrara, G. P. Incremona, E. Birliba, and P. Goatin, "Multi-scale model based hierarchical control of freeway traffic via platoons of connected and automated vehicles," *IEEE Open Journal of Intelligent Transportation Systems*, 2022.

[24] Y. H. Khalil and H. T. Mouftah, "Licanet: Further enhancement of joint perception and motion prediction based on multi-modal fusion," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 222–235, 2022.

[25] S. Kitajima, H. Chouchane, J. Antona-Makoshi, N. Uchida, and J. Tajima, "A nationwide impact assessment of automated driving systems on traffic safety using multiagent traffic simulations," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 302–312, 2022.

[26] H. Hu, Z. Liu, S. Chitlangia, A. Agnihotri, and D. Zhao, "Investigating the impact of multi-lidar placement on object detection for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2550–2559.

[27] S. Su, Y. Li, S. He, S. Han, C. Feng, C. Ding, and F. Miao, "Uncertainty quantification of collaborative detection for self-driving," *arXiv preprint arXiv:2209.08162*, 2022.

[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[30] Z. Zhao, Z. Wu, Y. Zhuang, B. Li, and J. Jia, "Tracking objects as pixel-wise distributions," *arXiv preprint arXiv:2207.05518*, 2022.

[31] Z. Zhao and J. Jia, "End-to-end view synthesis via nerf attention," *arXiv preprint arXiv:2207.14741*, 2022.

[32] Z. Zhao, K. Samel, B. Chen, and l. song, "Proto: Program-guided transformer for program-guided tasks," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 17 021–17 036. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/8d34201a5b85900908db6cae92723617-Paper.pdf

[33] W. Liu, K. Quijano, and M. Crawford, "Yolov5-tassel: Detecting tassels in rgb uav imagery with improved yolov5 based on transfer learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2022.

[34] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021.

[35] Z. Fan, Z. Song, H. Liu, Z. Lu, J. He, and X. Du, "Svt-net: Super light-weight sparse voxel transformer for large scale place recognition." AAAI, 2022.

[36] Z. Tu, H. Talebi, H. Zhang, P. Milanfar, A. Bovik, and Y. Li, "Maxvit: Multi-axis vision transformer," *arXiv preprint arXiv:2204.01697*, 2022.

[37] S. Song, H. Yu, Z. Miao, J. Fang, K. Zheng, C. Ma, and S. Wang, "Multi-spectral salient object detection by adversarial domain adaptation," in *AAAI Conference on Artificial Intelligence (AAAI)*, vol. 34, no. 07, 2020, pp. 12 023–12 030.

[38] W. Shao, S. Zhao, Z. Zhang, S. Wang, M. S. Rahaman, A. Song, and F. D. Salim, "Fadacs: A few-shot adversarial domain adaptation architecture for context-aware parking availability sensing," in *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2021, pp. 1–10.

[39] Z. Du, J. Li, H. Su, L. Zhu, and K. Lu, "Cross-domain gradient discrepancy minimization for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3937–3946.

[40] J. Li, Z. Xu, L. Fu, X. Zhou, and H. Yu, "Domain adaptation from daytime to nighttime: A situation-sensitive vehicle detection and traffic flow parameter estimation framework," *Transportation Research Part C: Emerging Technologies*, vol. 124, p. 102946, 2021.

[41] L. Fu, H. Yu, F. Juefei-Xu, J. Li, Q. Guo, and S. Wang, "Let there be light: Improved traffic surveillance via detail preserving night-to-day transfer," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[42] T. Xu, W. Chen, W. Pichao, F. Wang, H. Li, and R. Jin, "Cdtrans: Cross-domain transformer for unsupervised domain adaptation," in *International Conference on Learning Representations*, 2021.

[43] X. Yao, S. Zhao, P. Xu, and J. Yang, "Multi-source domain adaptation for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3273–3282.

[44] Z. Fan, Y. He, Z. Wang, K. Wu, H. Liu, and J. He, "Reconstruction-aware prior distillation for semi-supervised point cloud completion," *arXiv preprint arXiv:2204.09186*, 2022.

[45] Z. Fan, Y. Zhu, Y. He, Q. Sun, H. Liu, and J. He, "Deep learning on monocular object pose detection and tracking: A comprehensive overview," *ACM Computing Surveys (CSUR)*, 2021.

[46] L. Yang, L. Li, Z. Zhang, X. Zhou, E. Zhou, and Y. Liu, "Dpgn: Distribution propagation graph network for few-shot learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 390–13 399.

[47] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*. PMLR, 2015, pp. 97–105.

[48] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European conference on computer vision*. Springer, 2016, pp. 443–450.

[49] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 472–480.

[50] Y. Zhang, H. Tang, K. Jia, and M. Tan, "Domain-symmetric networks for adversarial domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5031–5040.

[51] H. Hu, Z. Qiao, M. Cheng, Z. Liu, and H. Wang, "Dasgil: Domain adaptation for semantic and geometric-aware image-based localization," *IEEE Transactions on Image Processing*, vol. 30, pp. 1342–1353, 2020.

[52] L. Yang and S. Hong, "Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion," in *International Conference on Machine Learning*. PMLR, 2022, pp. 25 038–25 054.

[53] Z. Liu, G. Chen, Z. Li, Y. Kang, S. Qu, and C. Jiang, "Psdc: A prototype-based shared-dummy classifier model for open-set domain adaptation," *IEEE Transactions on Cybernetics*, pp. 1–14, 2022.

[54] S. Qu, G. Chen, J. Zhang, Z. Li, W. He, and D. Tao, "Bmd: A general class-balanced multicentric dynamic prototype strategy for source-free domain adaptation," in *European conference on computer vision*, 2022.

[55] H. Talebi and P. Milanfar, "Learning to resize images for computer vision tasks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 497–506.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[57] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3339–3348.

[58] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[59] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[60] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[61] R. Xu, Y. Guo, X. Han, X. Xia, H. Xiang, and J. Ma, "Opencda: An open cooperative driving automation framework integrated with co-simulation," in *2021 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2021.

[62] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[63] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[64] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, Y. Shao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," *arXiv preprint arXiv:2209.00796*, 2022.