STAMP: SCALABLE TASK- AND MODEL-AGNOSTIC COLLABORATIVE PERCEPTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Perception is a crucial component of autonomous driving systems. However, single-agent setups often face limitations due to sensor constraints, especially under challenging conditions like severe occlusion, adverse weather, and long-range object detection. Multi-agent collaborative perception (CP) offers a promising solution that enables communication and information sharing between connected vehicles. Yet, the heterogeneity among agents—in terms of sensors, models, and tasks—significantly hinders effective and efficient cross-agent collaboration. To address these challenges, we propose STAMP, a scalable task- and model-agnostic collaborative perception framework tailored for heterogeneous agents. STAMP utilizes lightweight adapter-reverter pairs to transform Bird's Eye View (BEV) features between agent-specific domains and a shared protocol domain, facilitating efficient feature sharing and fusion while minimizing computational overhead. Moreover, our approach enhances scalability, preserves model security, and accommodates a diverse range of agents. Extensive experiments on both simulated (OPV2V) and real-world (V2V4Real) datasets demonstrate that STAMP achieves comparable or superior accuracy to state-of-the-art models with significantly reduced computational costs. As the first-of-its-kind task- and model-agnostic collaborative perception framework, STAMP aims to advance research in scalable and secure mobility systems, bringing us closer to Level 5 autonomy. Our project page is at https://jocular-manatee-91cad0.netlify.app and the code is available at https://anonymous.4open.science/r/STAMP.

1 INTRODUCTION

Multi-agent collaborative perception (CP) (Bai et al., 2022b; Han et al., 2023; Liu et al., 2023) has emerged as a promising solution for autonomous systems by leveraging communication among multiple connected and automated agents. It enables agents—such as vehicles, infrastructure, or even pedestrians—to share sensory and perceptual information, providing a more comprehensive view of the surrounding environment to enhance overall perception capabilities. Despite its potential, CP faces significant challenges, particularly when dealing with heterogeneous agents that defer in input modalities, model parameters, architectures, or learning objectives. For instance, Xu et al. (2023b) observed that features from heterogeneous agents vary in spatial resolution, channel number, and feature patterns. This domain gap hinders effective and efficient CP, particularly when employing fusion-based approaches.

To facilitate collaborative perception among heterogeneous agents—often referred to as heterogeneous collaborative perception—one might consider using early or late fusion. However, early fusion requires high communication bandwidth, making it impractical for real-time applications. Late fusion often results in suboptimal accuracy, and it is not viable across models with different downstream tasks. Alternative methods attempt to achieve heterogeneous intermediate fusion by either incorporating adapters (Xu et al., 2023b) or sharing parts of the models (Lu et al., 2024). While these approaches can bridge the domain gap, they are limited in scalability or security, rendering them inefficient or unsafe for practical deployment. Additionally, recent studies have highlighted increased security vulnerabilities in CP systems compared to single-agent frameworks (Hu et al., 2024; Tu et al., 2021; Li et al., 2023b). Notably, Li et al. (2023b) found that black-box attacks are nearly ineffective without the knowledge of other agents' models, emphasizing the importance of task- and model-agnostic approaches in enhancing system-level security against adversarial threats.

To address these challenges, we propose **STAMP**, a Scalable Task- And Model-agnostic collaborative Perception framework. Our approach employs lightweight adapter-reverter pairs to transform the Bird's Eye View (BEV) features of each heterogeneous agent into a unified protocol BEV feature domain. These protocol BEV features are then broadcasted to other agents and subsequently mapped back to their corresponding local domains, enabling collaboration within each agent's source domain. We refer to this process as **collaborative feature alignment (CFA)**. Our proposed pipeline offers several key advantages. Firstly, it enables existing heterogeneous agents to collaborate with minimal additional disk memory ($\sim$1MB) and computational overhead, making it scalable for a large number of heterogeneous agents. Secondly, the alignment process is designed to be task- and model-agnostic, allowing our framework to integrate with various models and tasks without retraining the model or the need to share models among agents, enhancing both *flexibility* and *security*.

We conducted comprehensive experiments to evaluate the performance of our collaborative perception framework. Using the simulated OPV2V dataset (Xu et al., 2022b) and the real-world V2V4Real dataset (Xu et al., 2023d), we demonstrated that our STAMP pipeline achieves comparable or superior accuracy with a significantly lower training resource growth rate as the number of heterogeneous agents increases. Our method requires, on average, only 2.36 GPU hours (**7.2x** saving) of training time per additional agent, compared to 17.07 GPU hours per additional agent for existing heterogeneous collaborative pipelines. We also demonstrated our pipeline's unique ability to perform *task- and model-agnostic* collaboration, a capability *not supported in existing methods*. This achievement establishes a new benchmark for heterogeneous CP in autonomous driving, showcasing clear performance improvements in scenarios where other methods are unable to operate.

2 Related Works

2.1 Multi-agent Collaborative Perception

Multi-agent CP has emerged as a promising solution to overcome the inherent limitations of single-agent perception systems, particularly in addressing occlusions and extending perception range (Hu et al., 2022a).

Information-sharing schemes. There are three main information-sharing schemes in multi-agent CP systems: early fusion, late fusion, and intermediate fusion. ❶ Early fusion (Gao et al., 2018; Chen et al., 2019; Arnold et al., 2020) involves the direct sharing of raw sensor data, such as LiDAR point clouds or camera images, between agents. This method maximizes information transfer but requires high bandwidth for transmitting. ❷ Late fusion (Melotti et al., 2020; Fu et al., 2020; Zeng et al., 2020; Shi et al., 2022; Glaser & Kira, 2023; Xu et al., 2023a; Su et al., 2023; 2024), involves sharing only final prediction results, such as object detection bounding boxes or occupancy predictions. This approach significantly reduces communication bandwidth overhead, making it more feasible for implementation in real-world systems. However, late fusion often results in suboptimal accuracy due to the loss of intermediate information that could be valuable for collaborative decision-making. ❸ Intermediate fusion (Wang et al., 2020; Liu et al., 2020b;a; Guo et al., 2021; Li et al., 2021; Hu et al., 2022b; Bai et al., 2022a; Cui et al., 2022; Xu et al., 2022a; 2023c; Qiao & Zulkernine, 2023; Li et al., 2023a; Wang et al., 2023; Yu et al., 2023; Yang et al., 2024) has emerged as a promising middle ground, involving the sharing of mid-level information, typically in the form of Bird's Eye View (BEV) features. This approach strikes a balance between communication bandwidth efficiency and information richness. Intermediate fusion allows for more flexibility in collaborative processing while maintaining a reasonable data transfer load. However, intermediate fusion faces significant challenges in addressing domain gap issues for heterogeneous agents.

Collaborative perception datasets. Several significant collaborative perception datasets have emerged recently (Yazgan et al., 2024). The simulated OPV2V (Xu et al., 2022b) and V2X-Sim dataset (Li et al., 2022) each contain approximately $10k$ multi-agent scenes, featuring RGB images and LiDAR point clouds with 3D object detection, tracking, and segmentation annotations. Two real-world DAIR-V2X Yu et al. (2022) and V2V4Real (Xu et al., 2023d) datasets provide $39k$ and $20k$ dual-agent samples, respectively, with object detection annotations only. For our framework evaluation, we selected two complementary datasets: the OPV2V dataset for its diverse downstream tasks, and the V2V4Real dataset to validate performance in real-world scenarios.

2.2 Heterogeneous Collaborative Perception

In a CP system, the heterogeneity of agents can manifest as three different types: heterogeneous modalities, heterogeneous model architectures or parameters, and heterogeneous downstream tasks. ❶ **Heterogeneous modalities.** Each model is expected to take input data of different modalities (e.g., RGB images or LiDAR point clouds), requiring different encoders to process the data. Xiang et al. (2023) propose a hetero-modal vision transformer to fuse heterogeneous BEV features, but this requires end-to-end model training, which is impractical for existing heterogeneous agents. Xu et al. (2023b) introduce multi-agent perception domain adaptation (MPDA), which aligns feature maps between heterogeneous agent pairs. While effective for collaboration, this method's polynomial complexity limits its scalability as the number of heterogeneous models increases. Lu et al. (2024) introduce a backward alignment training strategy, creating heterogeneous models by fixing a base network's decoder and training only the encoders. While this enables collaboration between existing heterogeneous agents, it incurs high computational costs, especially for models with large encoders. ❷ **Heterogeneous model architectures or parameters.** Model architectures or parameters may differ across agents, resulting in feature map in different domains, rendering existing heterogeneous intermediate fusion methods (Xiang et al., 2023; Lu et al., 2024) inapplicable. However, late fusion methods (Xu et al., 2023b) remain viable as the model output for all models is in the same domain. ❸ **Heterogeneous downstream tasks.** The learning objectives are different across agents, which results in model outputs in different domains. Li et al. (2023c) propose task-agnostic CP by training models with multi-robot scene completion objectives. Despite the effectiveness of task-agnostic collaboration, their method does not support heterogeneous modality inputs and model architectures.

3 Methodology

3.1 Preliminaries: Intermediate Collaborative Perception

A CP system typically comprises multiple (N) agents, each equipped with its own CP model. This work mainly focuses on intermediate fusion, so we consider all CP models to be trained using an intermediate fusion strategy. The architecture of these models generally consists of an encoder E_i, a compressor ϕ_i, a decompressor ψ_i, a collaborative fusion layer U_i, and a decoder D_i, where $i \in \{1, 2, \ldots, N\}$ represents the agent index.

The CP process unfolds as follows: Upon receiving input data I_i, the encoder E_i of agent i transforms this data into a Bird's Eye View (BEV) feature representation F_i. To save transmission bandwidth, each agent uses the compressor ϕ_i to compress F_i to $\tilde{F}_i$ before broadcasting them to other agents within a predefined collaborative distance δ. Here, δ denotes the maximum range for inter-agent collaboration. Each agent collects the BEV features from other agents and uses their own decompressor ψ_i to decompress $\tilde{F}_k$ to F_k, where agent i and agent k are within the distance δ for collaboration. Then, the collaborative fusion layer U_i collects and integrates the BEV features from all cooperating agents, producing a consolidated BEV feature F'. Finally, the decoder D_i processes this fused feature F' to generate the final model output O_i. This process can be formally described as follows for each agent $i \in \{1, 2, \ldots, N\}$:

$$\text{Encoding:} \quad F_i = E_i(I_i) \tag{1}$$
$$\text{Compression:} \quad \tilde{F}_i = \phi_i(F_i) \tag{2}$$
$$\text{Decompression:} \quad F_j = \psi_i(\tilde{F}_j), \quad \forall j \in \mathcal{N}(i,j) \leq \delta \tag{3}$$
$$\text{Fusion:} \quad F'_i = U_i(\{F_j \mid \mathcal{N}(i,j) \leq \delta\}) \tag{4}$$
$$\text{Decoding:} \quad O_i = D_i(F'_i) \tag{5}$$

where $\mathcal{N}(i,k)$ refers to the Euclidean distance between the agent i and agent k.

3.2 Framework Overview

Our proposed framework, STAMP, enables collaboration among existing heterogeneous agents without sharing model details or downstream task information. We replace the compression (Equation 2) and decompression (Equation 3) with adaptation and reversion steps. Specifically, for each agent i, we introduce a local adapter ϕ_i and a local reverter ψ_i. The adaptation process is defined as follows:

$$\text{Adaptation:} \quad F_{iP} = \phi_i(F_i), \quad \forall i \in \{1, 2, \ldots, N\} \tag{6}$$

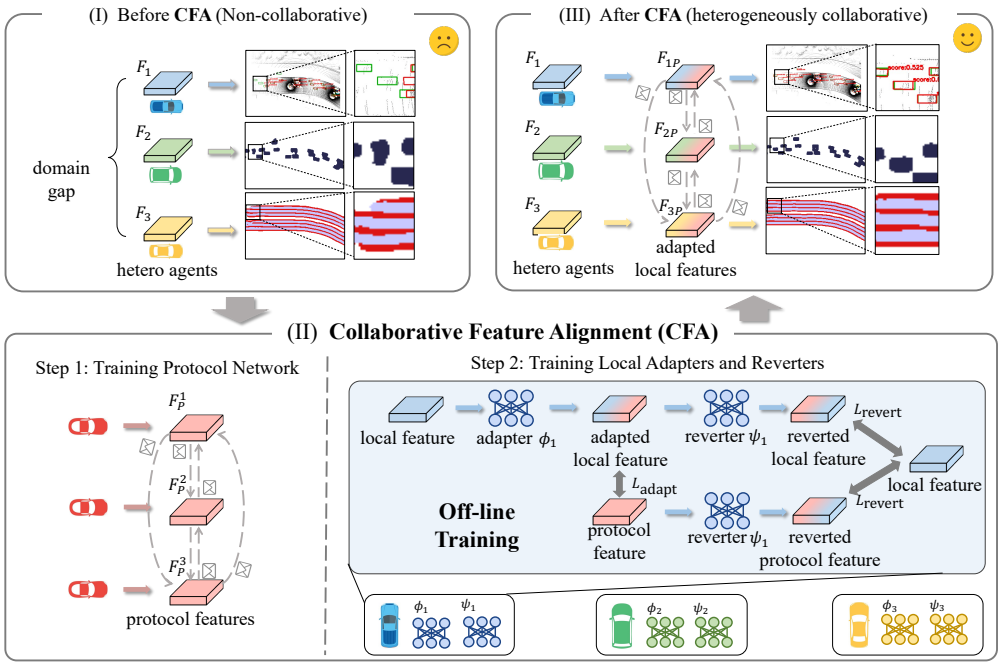

Figure 1: [We have provided a streamlined figure to enhance the clarity of our framework's architectural design.] Initially, agents are non-collaborative (I), resulting in degraded performance. Collaborative Feature Alignment (CFA) enables collaboration among heterogeneous agents through a two-step process (II): training a protocol network and training local adapters and reverters. The protocol network facilitates communication between Agent 1, Agent 2, and Agent 3, each with heterogeneous models and features. Gradient-colored feature maps represent features adapted or reverted between domains. After CFA implementation, agents become collaborative (III) with improved performance.

Here, the adapter ϕ_i maps the local BEV feature F_i to a unified BEV feature representation, which we term the protocol feature, denoted as F_P. The resulting adapted feature is denoted as F_{iP}.

Following adaptation, the features from all heterogeneous agents i are broadcast to other agents j within the collaborative distance δ. Each receiving agent j ($j \neq i$) then uses its local reverter ψ_j to map the received features back to its own local feature representation. The resulting reverted features are denoted as F_{ij}. This reversion process is formulated as:

$$\text{Reversion:} \quad F_{ij} = \begin{cases} \psi_j(F_{iP}), & \text{if } i \neq j \\ F_i, & \text{if } i = j \end{cases} \quad \forall i, j \in \{1, 2, \ldots, N\} \tag{7}$$

Note that F_i is already in the local feature presentation, so stay intact. This adaptation and reversion process seamlessly integrates into the standard heterogeneous CP pipeline, forming the core of our STAMP framework.

The STAMP framework supports agents with different modalities, model architectures, and downstream tasks, while maintaining a collaboration process that is entirely agnostic to those characteristics of other agents. To the best of our knowledge, STAMP is the first framework that simultaneously addresses all three aspects of agent heterogeneity. Furthermore, the adapters and reverters can be implemented in a highly lightweight manner, ensuring high scalability across a large number of heterogeneous agents. A comprehensive comparative summary STAMP and other heterogeneous CP frameworks is presented in Table 1.

3.3 Collaborative Feature Alignment

We propose the **Collaborative Feature Alignment (CFA)** module to train a unified BEV feature representation and a local adapter-reverter pair. As illustrated in Figure 1 (I), before CFA, heterogeneous agents perform multiple tasks individually without the ability to collaborate, resulting

Table 1: Comparison of heterogeneity support and scalability across existing heterogeneous collaboration frameworks. "†" indicates that while the current codebase does not support the specified heterogeneity, we believe the proposed method could accommodate it with minor modifications.

Frameworks	Modality	Model Architecture	Downstream Task	Scalability
Calibrator (Xu et al., 2023a)	✓	✓		high
MPDA (Xu et al., 2023b)	✓	†	†	low
HEAL (Lu et al., 2024)	✓			medium
Scene Completion (Li et al., 2023c)	†		✓	high
STAMP (**ours**)	✓	✓	✓	high

in suboptimal performance. After CFA (III), these agents can effectively collaborate, leading to a significant performance boost.

❶ **Training protocol network.** The first step is to learn a unified BEV embedding space by training a protocol network. This process follows the standard training process of a collaborative perception model, as described in Equations (1) to (5). We denote the protocol encoder, fusion model, and decoder as E_P, U_P, and D_P, respectively. The compressor and decompressor are set as identity functions. The input data, BEV feature, fused BEV feature, and final output of the protocol model are represented by I_P, F_P, F'_P, and O_P, respectively.

❷ **Training local adapters and reverters.** We introduce a notation $\mathcal{X} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^K\}$ to denote a set of **world states**, where K is the total number of world states. A modality transformation function $\mathcal{T} : \mathcal{X} \to I$ transforms a world state to sensor data of a given modality. For instance, in an autonomous driving scenario, $\mathbf{x}$ could represent the world state surrounding the ego vehicle, and $\mathcal{T}$ could be the matrix of six surround-view RGB cameras, resulting in I as the six RGB images captured by these cameras.

Given a local model i, protocol model P, and a set of world states $\mathcal{X} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^K\}$ within the collaborative distance, we define $I_i^k = \mathcal{T}_i(\mathbf{x}^k)$ as the input for local model i and $I_P^k = \mathcal{T}_P(\mathbf{x}^k)$ as the input for protocol model P. Here, $\mathcal{T}_i$ and $\mathcal{T}_P$ denote the sensor modalities of the local model i and protocol model, respectively. By passing these inputs into their corresponding encoders, we get a set of local BEV features $F_i^{1:K}$ and protocol BEV features $F_P^{1:K}$. A domain gap exists between $F_i^{1:K}$ and $F_P^{1:K}$ due to the heterogeneity of sensor modalities $\mathcal{T}_i$ and $\mathcal{T}_P$, as well as the encoders E_i and E_P. To bridge this gap, we introduce a local adapter ϕ_i that maps the local BEV feature F_i^k to the protocol feature representation. The objective function for ϕ_i is:

$$\phi_i = \arg\min_{\phi_i} L_{\phi_i}(F_{iP}^{1:K}, F_P^{1:K}) \quad \text{where} \quad F_{iP}^k = \phi_i(F_i^k) \tag{8}$$

where L_{ϕ_i} represent the feature alignment loss for the adapter ϕ_i. Similarly, we introduce a reverter ψ_i that maps the protocol BEV feature F_P^k to the local feature representation, *i.e.*, $F_{Pi}^k = \psi_i(F_P^k)$. since F_{iP}^k is also in the protocol representation, by including $F_{ii}^k = \psi_i(F_{iP}^k)$, we provide additional supervision for ψ_i. The objective function for the reverter is formulated as:

$$\begin{aligned} \psi_i &= \arg\min_{\psi_i} \left(L_{\psi_i}(F_{Pi}^{1:K}, F_i^{1:K}) + L_{\psi_i}(F_{ii}^{1:K}, F_i^{1:K}) \right) \\ \text{where} \quad & F_{Pi}^k = \psi_i(F_P^k),\ F_{ii}^k = \psi_i(F_{iP}^k) \end{aligned} \tag{9}$$

Here, L_{ψ_i} represents the feature alignment loss for the reverter ψ_i. To achieve our objective functions, we conduct alignment in both the feature space and the decision space.

❸ **Feature space alignment.** For a given local model i, we first align the feature pairs $(F_{iP}^{1:K}, F_P^{1:K})$, $(F_{Pi}^{1:K}, F_i^{1:K})$, and $(F_{ii}^{1:K}, F_i^{1:K})$ for all k using the $L2$-norm. This direct alignment of feature spaces is formulated as:

$$L_{\phi_i}^f = \frac{1}{K} \sum_k \|F_{iP}^k, F_P^k\|_2, \quad L_{\psi_i}^f = \frac{1}{K} \sum_k \left(\|F_{Pi}^k, F_i^k\|_2 + \|F_{ii}^k, F_i^k\|_2 \right) \tag{10}$$

❹ **Decision space alignment.** When $\mathcal{T}_i$ and $\mathcal{T}_P$ represent significantly different sensor modalities (e.g., RGB camera vs. LiDAR), the disparity between their intermediate feature representations can be substantial. In such cases, achieving exact equivalence between F_{iP}^k and F_P^k for all k is neither

feasible nor necessary. Nevertheless, since both F_i^k and F_{iP}^k are derived from the same world state $\mathbf{x}^k$, their corresponding downstream task outputs should align with the same ground truth labels. To enforce this alignment in the decision space, we introduce additional loss terms:

$$\begin{aligned} L_{\phi_i}^d &= \mathcal{L}_P(D_P \circ U_P(F_{iP}^{1:K}),\ \mathrm{GT}_P) \\ L_{\psi_i}^d &= \mathcal{L}_i(D_i \circ U_i(F_{Pi}^{1:K}),\ \mathrm{GT}_i) + \mathcal{L}_i(D_i \circ U_i(F_{ii}^{1:K}),\ \mathrm{GT}_i) \end{aligned} \tag{11}$$

where $\mathcal{L}_P$ and $\mathcal{L}_i$ represent the task-specific loss functions for training the protocol model and local model i, respectively. GT_P and GT_i denote the corresponding ground truth labels for the protocol and local model i. Finally, to balance the importance of adaptation and reversion, as well as feature and decision space alignment, we introduce scaling factors λ_ϕ^f, λ_ψ^f, λ_ϕ^d, and λ_ψ^d. The total loss function for local model i is:

$$L_i = \lambda_\phi^f L_{\phi_i}^f + \lambda_\psi^f L_{\psi_i}^f + \lambda_\phi^d L_{\phi_i}^d + \lambda_\psi^d L_{\psi_i}^d \tag{12}$$

3.4 Adapter and Reverter Architecture

To bridge the domain gap between heterogeneous agents, we propose a flexible architecture for both the adapter ϕ and reverter ψ. This architecture addresses three main sources of domain gap caused by agent heterogeneity, as identified by Xu et al. (2023b): spatial resolution, feature patterns, and channel dimensions. Our design employs simple linear interpolation for spatial resolution alignment, three ConvNeXt layers (Liu et al., 2022) with hidden channel dimension C_{hidden} for feature pattern alignment, and two additional convolutional layers for channel dimension alignment (input: $C_{\text{in}} \rightarrow C_{\text{hidden}}$, output: $C_{\text{hidden}} \rightarrow C_{\text{out}}$). For the model architecture details, please refer to Appendix A.1.

Note that this high-level architecture is flexible and open to various implementations. In Section 4.4, we evaluate alternative approaches for feature pattern alignment, demonstrating our framework's flexibility across different specific implementations.

4 Experiments

Our STAMP framework enables collaboration among agents with heterogeneous modalities, models architectures, and downstream tasks without sharing model or task information. We first compare our framework with existing heterogeneous CP frameworks in Section 4.2. Given that no previous work supports simultaneous task- and model-agnostic heterogeneous collaboration, we concentrate our evaluation on the 3D object detection task. This focus ensures a fair comparison across two key dimensions: object detection average precision, trainable parameters and GPU hours required for training. Next, in Section 4.3, we demonstrate our framework's unique capability in a task- and model-agnostic setting, evaluating its performance using four existing collaborative models with heterogeneous architectures and downstream tasks. Then, we present ablation studies on channel sizes, model architectures, and loss functions in Section 4.4 to further analyze our framework's design choices. Finally, we present some feature and output visualization in Section 4.5.

4.1 Experimental Setup

Our experiments utilize two CP datasets: the simulated OPV2V dataset (Xu et al., 2022b) and the real-world V2V4Real dataset (Xu et al., 2023d). We employ both datasets in Section 4.2 and Section 4.4 for method comparison and ablation studies. The task- and model-agnostic evaluation in Section 4.3 uses only OPV2V due to its multi-task annotations. This combination leverages the scale and diversity of simulated data with the realism of real-world data, ensuring comprehensive model evaluation.

Implementation details. We use different setups for 3D object detection (Section 4.2) and task-agnostic settings (Section 4.3), detailed within each section. Unless using end-to-end training, local and protocol models are trained for 30 epochs using Adam optimizer (Kingma & Ba, 2014). For end-to-end training, we use $\text{Iters}_N = 30N$ epochs, where N is the number of heterogeneous models, to ensure all models receive the same amount of supervision. Local adapters ϕ and reverters ψ are trained for 5 epochs. We set loss scaling factors $\lambda_{\text{adapt}}^f = \lambda_{\text{revert}}^f = \lambda_{\text{adapt}}^d = \lambda_{\text{revert}}^d = 0.5$ empirically. For additional details, please refer to Section 4.2, Section 4.3, and Appendix A.1.

Table 2: Performance comparison using AP@30 and AP@50 metrics on the OPV2V dataset. Agent positions are perturbed with Gaussian noise of standard deviations 0.0, 0.2, and 0.4.

		AP@30 ↑				AP@50 ↑			
σ	Agent Index	**Agent 1**	**Agent 2**	**Agent 3**	**Agent 4**	**Agent 1**	**Agent 2**	**Agent 3**	**Agent 4**
0.0	Late Fusion	**0.902**	0.931	0.935	0.935	0.894	0.908	0.913	0.914
	Calibrator	0.901	0.935	0.939	0.938	**0.896**	0.914	0.916	0.920
	E2E Training	0.899	0.973	0.978	0.986	0.885	0.967	0.977	0.980
	HEAL	0.897	0.975	0.986	0.985	0.889	0.972	0.978	0.983
	STAMP (ours)	**0.902**	**0.981**	**0.987**	**0.989**	0.894	**0.977**	**0.983**	**0.985**
0.2	Late Fusion	**0.900**	0.910	0.902	0.905	0.882	0.783	0.797	0.792
	Calibrator	0.897	0.908	0.898	0.902	**0.885**	0.778	0.800	0.791
	E2E Training	**0.900**	0.967	0.961	0.961	0.879	0.936	0.941	0.934
	HEAL	0.899	0.971	0.965	0.962	0.881	**0.938**	0.940	0.940
	STAMP (ours)	**0.900**	**0.975**	**0.968**	**0.968**	0.882	0.937	**0.948**	**0.946**
0.4	Late Fusion	**0.888**	0.882	0.864	0.870	**0.874**	0.639	0.614	0.617
	Calibrator	0.874	0.889	0.887	0.871	0.867	0.644	0.622	0.628
	E2E Training	0.885	0.952	0.956	0.952	0.863	0.883	0.892	0.899
	HEAL	0.880	0.959	0.961	0.962	0.852	0.913	**0.915**	**0.912**
	STAMP (ours)	**0.888**	**0.961**	**0.963**	**0.966**	**0.874**	**0.915**	**0.915**	0.909

4.2 Heterogeneous Collaborative Perception For 3D Object Detection

Performance comparison. We compare our method with existing heterogeneous CP approaches on the 3D object detection task. We select two late fusion methods (vanilla late fusion and calibrator (Xu et al., 2023a)) and two intermediate fusion methods (end-to-end training and HEAL (Lu et al., 2024)) for comparison. Late fusion methods offer a simple way to mitigate domain gaps in collaborative 3D object detection. Xu et al. (2023a) propose using a calibrator to address residual domain gaps in late fusion, which arise from differences in training data and procedures among heterogeneous models. For intermediate fusion, end-to-end training of all heterogeneous models together allows collaboration during the training stage to bridge domain gaps. Lu et al. (2024) introduce a backward alignment technique, first training a base network, then fixing its decoder while training only the encoders to create heterogeneous models. An architectural comparison between these frameworks and our proposed STAMP framework is illustrated and visualized in Appendix A2.

We prepared 12 heterogeneous local models (six with LiDAR modality and six with RGB camera modality) and one protocol model with LiDAR modality (details in Appendix A.1). Each agent has a visible range of $51.2\text{m} \times 51.2\text{m}$ square units. Considering that most samples of the OPV2V dataset contain no more than four agents, we only select the first four models for evaluation on the OPV2V dataset. Similarly, we select the first two models for the V2V4Real dataset since it has two agents for each sample. All 12 models are used for efficiency comparison.

For the OPV2V dataset, we simulate real-world noise by adding Gaussian noise with standard deviations $\sigma = \{0.0, 0.2, 0.4\}$ to the agents' locations. As shown in Table 2, late fusion methods underperform as the number of agents increases, with performance degrading further at higher noise levels ($\sigma = 0.4$). This is particularly evident when camera agents (agents 3 and 4) are involved, highlighting the late fusion methods' vulnerability to bottleneck agents' incorrect predictions. Our framework demonstrates superior or comparable performance to other heterogeneous fusion methods across all noise levels.

Table 3 compares the average precision on the real-world V2V4Real dataset. Our STAMP pipeline demonstrates superior performance, achieving the highest AP@30 for both agents (0.523 and 0.633) and competitive AP@50 scores. STAMP outperforms Late Fusion methods and matches or exceeds the performance of existing heterogeneous intermediate fusion approaches like HEAL. These results indicate that the CFA module in STAMP is effective not only in simulated environments but also in real-world scenarios.

Table 3: Performance comparison using AP@30 and AP@50 metrics on the V2V4Real dataset.

	AP@30 ↑		AP@50 ↑	
Agent Index	**Agent 1**	**Agent 2**	**Agent 1**	**Agent 2**
Late Fusion	0.523	0.511	0.483	0.471
Calibrator	0.520	0.524	0.484	0.488
E2E Training	0.513	0.612	0.473	0.598
HEAL	0.515	0.628	0.480	**0.595**
STAMP (ours)	**0.523**	**0.633**	**0.483**	0.594

Efficiency comparison. We conducted an efficiency comparison between our approach and existing heterogeneous CP pipelines, focusing on the total number of parameters and training GPU hours. Training GPU hours refers to the time required to complete model training on the OPV2V dataset using an RTA A6000 GPU. To analyze training costs at scale, we report the number of training parameters and the estimated training time. Figure 2 illustrates the changes in the number of training parameters and estimated training GPU hours as the number of heterogeneous agents increases from 1 to 12. End-to-end training and HEAL exhibit a steep increase in both parameters and GPU hours as the number of agents grows. In contrast, although our pipeline shows higher parameters and GPU hours at the one or two number of agents (due to the training of the protocol model), it demonstrates a much slower growth rate because our proposed adapter ϕ and reverter ψ is very light-weighted and only takes 5 epochs to finish training. This highlights the scalability of our pipeline.

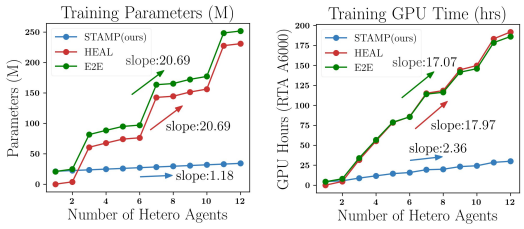

Figure 2: Training efficiency comparison of our framework and existing heterogeneous CP frameworks across a number of heterogeneous agents.

4.3 Model- and Task-agnostic Fusion

In this section, we evaluate our proposed framework's performance in a task-agnostic setting using the OPV2V dataset. To simulate agent heterogeneity, we assign four agents with diverse input sensors, learning objectives, and evaluation metrics, equipping them with various backbones and fusion models. Agent 1 was equipped with a SECOND encoder (Yan et al., 2018) and a window attention fusion module (Xu et al., 2022a). For Agent 2, we implemented an EfficientNet-b0 encoder (Tan, 2019), while Agents 3 and 4 were equipped with PointPillar encoders (Lang et al., 2019). Agents 2, 3, and 4 all utilized the Pyramid Fusion module Lu et al. (2024). Table 4 summarizes these models' key characteristics. We compare our STAMP framework against two baseline scenarios: non-collaborative (single-agent perception without information sharing) and collaborative without feature alignment (performing intermediate fusion despite domain gaps). Table 4 presents the evaluation results on the OPV2V dataset, with added Gaussian noise (standard deviations $\sigma = \{0.0, 0.2, 0.4\}$) to the agents' locations.

Table 4: Heterogeneous CP results in a model- and task-agnostic setting. Tasks include 3D object detection ('Object Det'), static object BEV segmentation ('Static Seg'), and dynamic object BEV segmentation ('Dynamic Seg'). 3D object detection is evaluated using Average Precision at 50% IoU threshold (AP@50), while segmentation tasks use Mean Intersection over Union (MIoU).

	Agent Index	**Agent 1**	**Agent 2**	**Agent 3**	**Agent 4**
Agent Info	Metric	AP@50	AP@50	MIoU	MIoU
	Downstream Task	Object Det	Object Det	Static Seg	Dynamic Seg
	Sensor Modality	Lidar	Camera	Lidar	Lidar
	Backbone	SECOND	EfficientNet-b0	PointPillar	PointPillar
	Feature Resolution	64×64	128×128	128×128	128×128
	Channel Size	256	64	64	64
	Fusion Method	Window Attention	Pyramid Fusion	Pyramid Fusion	Pyramid Fusion
Evaluation (σ : 0.0)	Non-Collab	**0.941**	0.399	0.548	0.675
	Collab w/o. CFA	0.909 (−0.032)	0.399 (+0.000)	0.114 (−0.434)	0.070 (−0.605)
	STAMP (ours)	0.936 (−0.005)	**0.760** (+0.362)	**0.624** (+0.076)	**0.690** (+0.014)
Evaluation (σ : 0.2)	Non-Collab	**0.936**	0.399	0.521	0.658
	Collab w/o. CFA	0.902 (−0.034)	0.399 (+0.000)	0.114 (−0.407)	0.069 (−0.588)
	STAMP (ours)	0.930 (−0.006)	**0.734** (+0.336)	**0.615** (+0.094)	**0.676** (+0.018)
Evaluation (σ : 0.4)	Non-Collab	**0.925**	0.399	0.503	0.630
	Collab w/o. CFA	0.886 (−0.039)	0.400 (+0.001)	0.114 (−0.389)	0.069 (−0.561)
	STAMP (ours)	0.923 (−0.002)	**0.585** (+0.186)	**0.600** (+0.097)	**0.650** (+0.020)

Our method consistently outperforms single-agent segmentation for agents 3 and 4 in the BEV segmentation task. Conversely, collaboration without feature alignment significantly degrades performance compared to the single-agent baseline, underscoring the importance of our adaptation mechanism in aligning heterogeneous features. For agent 2's camera-based 3D object detection, our pipeline achieves substantial gains (e.g., AP@50 improves from 0.399 to 0.760 in noiseless

conditions), while collaboration without feature alignment shows negligible changes. These results demonstrate our pipeline's effectiveness in bridging domain gaps between heterogeneous agents, enabling successful collaboration across diverse models, sensors, and tasks. The consistent improvements, particularly under noisy conditions, highlight our approach's robustness and adaptability.

However, we observe that both collaborative approaches lead to performance degradation for Agent 1 compared to its single-agent baseline, despite our method outperforming collaboration without feature alignment. This unexpected outcome is attributed to Agent 2's limitations, which rely solely on less accurate camera sensors for 3D object detection. This scenario illustrates a bottleneck effect, where a weaker agent constrains the overall system performance, negatively impacting even the strongest agents. This challenge in multi-agent collaboration systems prompts us to introduce the concept of a **Multi-group Collaboration System**. In Appendix A.3, we elaborate on the advantages of such a system and demonstrate how our framework can be easily integrated to potentially mitigate performance discrepancies in heterogeneous agent collaborations.

4.4 Ablation Studies

In this section, we conduct ablation studies on three factors that may affect our pipeline's performance: BEV feature channel size, adapter & reverter architectures, and loss functions for collaborative feature alignment. All experiments are conducted on both the OPV2V and V2V4Real datasets.

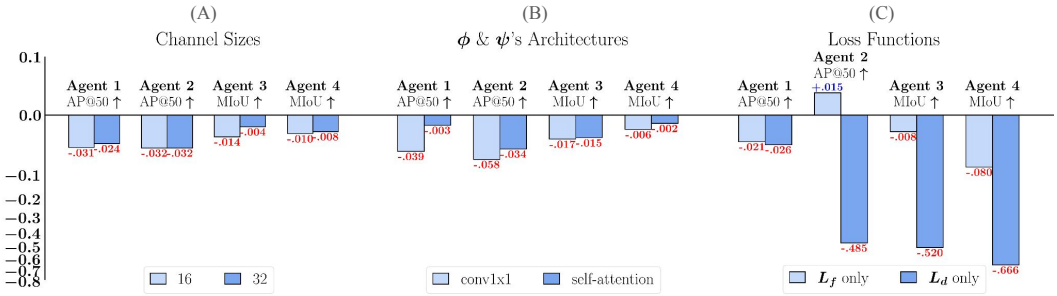

Figure 3: Ablation studies on the OPV2V dataset: (a) Model performance across different BEV feature channel sizes. (b) Performance comparison of various adapter and reverter architectures. (c) Performance results using different combinations of loss function components (L_f and L_d).

BEV feature channel size. Changing the BEV feature channel size is essentially a form of feature compression, which is crucial for controlling communication bandwidth in multi-agent collaboration systems. Our collaborative feature alignment module inherently supports feature compression by adjusting the protocol BEV feature's channel size. We experiment with two channel sizes for the protocol BEV feature, 32 and 16, and compare their performance to our standard implementation with a channel size of 64 (Figures 3 and 4). Surprisingly, reducing the channel size results in only minor performance changes for both datasets, revealing our model's resilience to high BEV feature compression rates.

Figure 4: Ablation studies on the V2V4real set.

Adapter & reverter architecture. We evaluate two alternative architectures for the adapter and reverter—a single 1×1 convolutional layer and three self-attention layers—compared to our standard implementation of three ConvNeXt layers. The results demonstrate that performance is not highly sensitive to the adapter and reverter architecture, showcasing our framework's flexibility across various specific implementations.

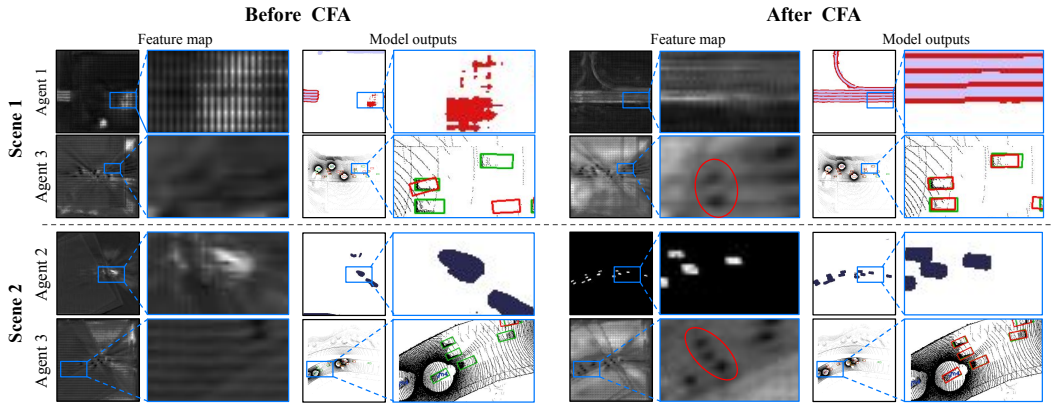

Figure 5: Visualization of feature maps and model outputs before and after Collaborative Feature Alignment (CFA) for two scenes with different agents and tasks. For 3D object detection, green boxes indicate the ground truth labels and red boxes indicate the predictions. CFA enhances feature clarity and information preservation, resulting in improved perception accuracy across heterogeneous agents.

Loss function. The final loss function in our collaborative feature alignment module comprises two components: L_f for feature space alignment and L_d for decision space alignment. We evaluate our method's performance using each loss function individually. Figure 3 shows that on the OPV2V dataset, using only L_d leads to a significant performance drop, while using only L_f results in more fluctuating and generally lower performance. Figure 4 demonstrates that on the V2V4Real dataset, dropping either L_d or L_f results in performance degradation. These findings underscore the necessity of using both loss functions in combination for optimal performance.

4.5 Visualization

Figure 5 illustrates the impact of our CFA method on feature maps and output results across various tasks. We visualize feature maps by averaging each channel of the fused feature map, F_i', to a (W, H) shape and plotting in a grayscale. Without CFA, the fused feature maps appear noisy and lack critical information for downstream tasks, leading to poor output results. In contrast, CFA significantly enhances feature preservation, resulting in clearer feature maps and more accurate outputs across different tasks. This visualization demonstrates CFA's effectiveness in maintaining essential information during the fusion process, which directly translates to improved performance in CP tasks. More comprehensive visualization results are shown on the Appendix A.4.

5 Conclusion

In this paper, we introduce STAMP, a scalable, task- and model-agnostic multi-agent collaborative perception framework. This framework simultaneously addresses three aspects of agent heterogeneity: varieties in modalities, model architectures, and downstream learning tasks. By utilizing lightweight adapter-reverter pairs, STAMP enables efficient collaborative perception while maintaining high security, scalability, and flexibility. Experiments on both the simulated OPV2V dataset and the real-world V2V4Real datasets demonstrate its superior performance and computational efficiency over existing state-of-the-arts. This approach opens new avenues for developing more reliable, efficient, and secure collaborative systems in future autonomous driving applications.

Limitations. Our experiments revealed a bottleneck effect in Collaborative Perception (CP), where the performance of the weakest agent constrains the overall system performance. This finding underscores the necessity for multi-group collaborative systems, where agents communicate only within defined groups. Such systems could mitigate the bottleneck effect by allowing for more selective collaboration. In Appendix A.3, we provide a more detailed discussion of multi-group collaborative systems and the advantages of our framework in this context.

Reproducibility statement. To ensure the reproducibility of our results, we have provided detailed information about our experimental setup, including dataset descriptions, model architectures, and training procedures in the main text and appendices. We encourage researchers to refer to Appendix A.1 for more implementation details. We also release the codebase at https://anonymous.4open.science/r/STAMP.

Ethics statement. Our task- and model-agnostic framework enhances local model security, reducing risks like model stealing (Oliynyk et al., 2023) and adversarial attacks (Tu et al., 2021). While limiting model sharing improves security, as assessed by (Li et al., 2023b), we recognize the need for further security analysis. We advocate for collaboration with experts to rigorously evaluate and strengthen our approach in order to contribute to safer and more trustworthy autonomous driving systems and advance privacy in collaborative perception.

REFERENCES

Eduardo Arnold, Mehrdad Dianati, Robert de Temple, and Saber Fallah. Cooperative perception for 3d object detection in driving scenarios using infrastructure sensors. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):1852–1864, 2020. 2

Zhengwei Bai, Guoyuan Wu, Matthew J Barth, Yongkang Liu, Emrah Akin Sisbot, and Kentaro Oguchi. Pillargrid: Deep learning-based cooperative perception for 3d object detection from onboard-roadside lidar. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1743–1749. IEEE, 2022a. 2

Zhengwei Bai, Guoyuan Wu, Matthew J Barth, Yongkang Liu, Emrah Akin Sisbot, Kentaro Oguchi, and Zhitong Huang. A survey and framework of cooperative perception: From heterogeneous singleton to hierarchical cooperation. *arXiv preprint arXiv:2208.10590*, 2022b. 1

Qi Chen, Sihai Tang, Qing Yang, and Song Fu. Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 514–524. IEEE, 2019. 2

Jiaxun Cui, Hang Qiu, Dian Chen, Peter Stone, and Yuke Zhu. Coopernaut: End-to-end driving with cooperative perception for networked vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17252–17262, 2022. 2

Chen Fu, Chiyu Dong, Christoph Mertz, and John M Dolan. Depth completion via inductive fusion of planar lidar and monocular camera. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10843–10848. IEEE, 2020. 2

Hongbo Gao, Bo Cheng, Jianqiang Wang, Keqiang Li, Jianhui Zhao, and Deyi Li. Object classification using cnn-based fusion of vision and lidar in autonomous vehicle environment. *IEEE Transactions on Industrial Informatics*, 14(9):4224–4231, 2018. 2

Nathaniel Moore Glaser and Zsolt Kira. We need to talk: Identifying and overcoming communication-critical scenarios for self-driving. *arXiv preprint arXiv:2305.04352*, 2023. 2

Jingda Guo, Dominic Carrillo, Sihai Tang, Qi Chen, Qing Yang, Song Fu, Xi Wang, Nannan Wang, and Paparao Palacharla. Coff: Cooperative spatial feature fusion for 3-d object detection on autonomous vehicles. *IEEE Internet of Things Journal*, 8(14):11078–11087, 2021. 2

Yushan Han, Hui Zhang, Huifang Li, Yi Jin, Congyan Lang, and Yidong Li. Collaborative perception in autonomous driving: Methods, datasets, and challenges. *IEEE Intelligent Transportation Systems Magazine*, 2023. 1

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 1

Hanjiang Hu, Zuxin Liu, Sharad Chitlangia, Akhil Agnihotri, and Ding 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*, pp. 2550–2559, 2022a. 2

Senkang Hu, Zhengru Fang, Yiqin Deng, Xianhao Chen, and Yuguang Fang. Collaborative perception for connected and autonomous driving: Challenges, possible solutions and opportunities. *arXiv preprint arXiv:2401.01544*, 2024. 1

Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *Advances in neural information processing systems*, 35:4874–4886, 2022b. 2

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12697–12705, 2019. 8, 1

Jinlong Li, Runsheng Xu, Xinyu Liu, Jin Ma, Zicheng Chi, Jiaqi Ma, and Hongkai Yu. Learning for vehicle-to-vehicle cooperative perception under lossy communication. *IEEE Transactions on Intelligent Vehicles*, 8(4):2650–2660, 2023a. 2

Yiming Li, Shunli Ren, Pengxiang Wu, Siheng Chen, Chen Feng, and Wenjun Zhang. Learning distilled collaboration graph for multi-agent perception. *Advances in Neural Information Processing Systems*, 34:29541–29552, 2021. 2

Yiming Li, Dekun Ma, Ziyan An, Zixun Wang, Yiqi Zhong, Siheng Chen, and Chen Feng. V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving. *IEEE Robotics and Automation Letters*, 7(4):10914–10921, 2022. 2

Yiming Li, Qi Fang, Jiamu Bai, Siheng Chen, Felix Juefei-Xu, and Chen Feng. Among us: Adversarially robust collaborative perception by consensus. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 186–195, 2023b. 1, 11

Yiming Li, Juexiao Zhang, Dekun Ma, Yue Wang, and Chen Feng. Multi-robot scene completion: Towards task-agnostic collaborative perception. In *Conference on Robot Learning*, pp. 2062–2072. PMLR, 2023c. 3, 5

Si Liu, Chen Gao, Yuan Chen, Xingyu Peng, Xianghao Kong, Kun Wang, Runsheng Xu, Wentao Jiang, Hao Xiang, Jiaqi Ma, et al. Towards vehicle-to-everything autonomous driving: A survey on collaborative perception. *arXiv preprint arXiv:2308.16714*, 2023. 1

Yen-Cheng Liu, Junjiao Tian, Nathaniel Glaser, and Zsolt Kira. When2com: Multi-agent perception via communication graph grouping. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 4106–4115, 2020a. 2

Yen-Cheng Liu, Junjiao Tian, Chih-Yao Ma, Nathan Glaser, Chia-Wen Kuo, and Zsolt Kira. Who2com: Collaborative perception via learnable handshake communication. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6876–6883. IEEE, 2020b. 2

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022. 6

Yifan Lu, Yue Hu, Yiqi Zhong, Dequan Wang, Siheng Chen, and Yanfeng Wang. An extensible framework for open heterogeneous collaborative perception. *arXiv preprint arXiv:2401.13964*, 2024. 1, 3, 5, 7, 8, 2

Gledson Melotti, Cristiano Premebida, and Nuno Gonçalves. Multimodal deep-learning for object recognition combining camera and lidar data. In *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 177–182. IEEE, 2020. 2

Daryna Oliynyk, Rudolf Mayer, and Andreas Rauber. I know what you trained last summer: A survey on stealing machine learning models and defences. *ACM Computing Surveys*, 55(14s): 1–41, 2023. 11

Donghao Qiao and Farhana Zulkernine. Adaptive feature fusion for cooperative perception using lidar point clouds. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 1186–1195, 2023. 2

Shuyao Shi, Jiahe Cui, Zhehao Jiang, Zhenyu Yan, Guoliang Xing, Jianwei Niu, and Zhenchao Ouyang. Vips: Real-time perception fusion for infrastructure-assisted autonomous driving. In *Proceedings of the 28th annual international conference on mobile computing and networking*, pp. 133–146, 2022. 2

Sanbao Su, Yiming Li, Sihong He, Songyang Han, Chen Feng, Caiwen Ding, and Fei Miao. Uncertainty quantification of collaborative detection for self-driving. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5588–5594. IEEE, 2023. 2

Sanbao Su, Songyang Han, Yiming Li, Zhili Zhang, Chen Feng, Caiwen Ding, and Fei Miao. Collaborative multi-object tracking with conformal uncertainty propagation. *IEEE Robotics and Automation Letters*, 2024. 2

Mingxing Tan. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 8, 1

James Tu, Tsunhsuan Wang, Jingkang Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. Adversarial attacks on multi-agent communication. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7768–7777, 2021. 1, 11

Binglu Wang, Lei Zhang, Zhaozhong Wang, Yongqiang Zhao, and Tianfei Zhou. Core: Cooperative reconstruction for multi-agent perception. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8676–8686. IEEE Computer Society, 2023. 2

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 *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 605–621. Springer, 2020. 2

Hao Xiang, Runsheng Xu, and Jiaqi Ma. Hm-vit: Hetero-modal vehicle-to-vehicle cooperative perception with vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 284–295, 2023. 3

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 *European conference on computer vision*, pp. 107–124. Springer, 2022a. 2, 8

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 *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2583–2589. IEEE, 2022b. 2, 6

Runsheng Xu, Weizhe Chen, Hao Xiang, Xin Xia, Lantao Liu, and Jiaqi Ma. Model-agnostic multi-agent perception framework. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1471–1478. IEEE, 2023a. 2, 5, 7

Runsheng Xu, Jinlong Li, Xiaoyu Dong, Hongkai Yu, and Jiaqi Ma. Bridging the domain gap for multi-agent perception. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6035–6042. IEEE, 2023b. 1, 3, 5, 6

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 *Conference on Robot Learning*, pp. 989–1000. PMLR, 2023c. 2

Runsheng Xu, Xin Xia, Jinlong Li, Hanzhao Li, Shuo Zhang, Zhengzhong Tu, Zonglin Meng, Hao Xiang, Xiaoyu Dong, Rui Song, et al. V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13712–13722, 2023d. 2, 6

Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 8, 1

Dingkang Yang, Kun Yang, Yuzheng Wang, Jing Liu, Zhi Xu, Rongbin Yin, Peng Zhai, and Lihua Zhang. How2comm: Communication-efficient and collaboration-pragmatic multi-agent perception. *Advances in Neural Information Processing Systems*, 36, 2024. 2

Melih Yazgan, Mythra Varun Akkanapragada, and J Marius Zöllner. Collaborative perception datasets in autonomous driving: A survey. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pp. 2269–2276. IEEE, 2024. 2

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21361–21370, 2022. 2

Haibao Yu, Yingjuan Tang, Enze Xie, Jilei Mao, Jirui Yuan, Ping Luo, and Zaiqing Nie. Vehicle-infrastructure cooperative 3d object detection via feature flow prediction. *arXiv preprint arXiv:2303.10552*, 2023. 2

Wenyuan Zeng, Shenlong Wang, Renjie Liao, Yun Chen, Bin Yang, and Raquel Urtasun. Dsdnet: Deep structured self-driving network. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pp. 156–172. Springer, 2020. 2

Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4490–4499, 2018. 1

A APPENDIX

A.1 IMPLEMENTATION DETAILS

For training all models, we initialize the learning rate at 0.001 and reduce it by a factor of 0.1 at 50% and 83% of the total epochs. We utilize a single NVIDIA RTX A6000 GPU for both model training and inference. Training time for each model varies between 7 to 30 GPU hours, depending on the specific model architecture. For adapters and reverters, we start with a learning rate of 0.01, reducing it by a factor of 0.1 after the first epoch. These components are trained in pairs, requiring 1 to 5 GPU hours depending on the specific encoder and decoder architectures.

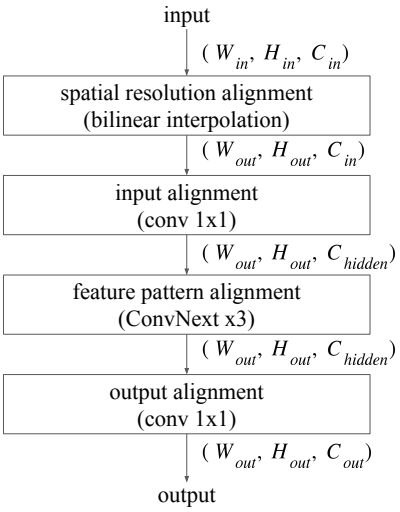

Figure A1: Architecture of adapter and reverter.

Adapter and Reverter's Architecture. We use the same architectures for both adapters and reverters across all CP models, as visualized in Figure A1. The dimension of the broadcasting feature map is set to $(128, 128, 64)$. C_{hidden} is set to be 64. $W_{\text{in}}, H_{\text{in}}, C_{\text{in}}, W_{\text{out}}, H_{\text{out}},$ and C_{out} of adapters and reverters vary according to the feature dimensions of each local model and the broadcasting feature map dimension. For instance, in the task- and model-agnostic setting, Agent 1's feature dimension is $128 \times 128 \times 64$, so we set $(W_{\text{in}}, H_{\text{in}}, C_{\text{in}}) = (W_{\text{out}}, H_{\text{out}}, C_{\text{out}}) = (128, 128, 64)$ for both its adapter and reverter. For Agent 2, with a feature dimension of $64 \times 64 \times 256$, we configure the adapter with $(W_{\text{in}}, H_{\text{in}}, C_{\text{in}}) = (64, 64, 256)$ and $(W_{\text{out}}, H_{\text{out}}, C_{\text{out}}) = (128, 128, 64)$, while the reverter is set with $(W_{\text{in}}, H_{\text{in}}, C_{\text{in}}) = (128, 128, 64)$ and $(W_{\text{out}}, H_{\text{out}}, C_{\text{out}}) = (64, 64, 256)$.

Index	**Agent 1**	**Agent 2**	**Agent 3**	**Agent 4**
Modality	Lidar	Lidar	Camera	Camera
Encoder	PointPillar (Lang et al., 2019)	SECOND (Yan et al., 2018)	EfficientNetB0 (Tan, 2019)	ResNet101 (He et al., 2016)
Encoder Param.(M)	0.87	3.79	56.85	6.88
Index	**Agent 5**	**Agent 6**	**Agent 7**	**Agent 8**
Modality	Camera	Lidar	Camera	Lidar
Encoder	ResNet34 (He et al., 2016)	VoxelNet (Zhou & Tuzel, 2018)	EfficientNetB1 (Tan, 2019)	PointPillar (large) (Lang et al., 2019)
Encoder Param.(M)	6.51	2.13	66.41	1.91
Index	**Agent 9**	**Agent 10**	**Agent 11**	**Agent 12**
Modality	Camera	Lidar	Camera	Lidar
Encoder	ResNet50 (He et al., 2016)	SECOND (large) (Yan et al., 2018)	EfficientNetB2 (Tan, 2019)	VoxelNet (large) (Zhou & Tuzel, 2018)
Encoder Param.(M)	6.88	4.82	71.43	3.18

Table A1: Modality, encoder, and encoder parameters (M) of each heterogeneous model in the 3D object detection setting.

3D object detection setting. Under the experiments on 3D object detection task, we prepared 12 heterogeneous models. Table A1 displays the Modality, Encoder, and Encoder Parameters (M) information of each of the 12 heterogeneous models. For model 7, 9, and 11, we enlarge the encoders by increasing the size of hidden layers. For all heterogeneous models, we choose pyramid fusion layers proposed by Lu et al. (2024) to be the fusion module and three 1×1 convolutional layers for classification, regression, and direction, respectively.

A.2 Architectural Comparison between Existing Frameworks

Figure A2 illustrated various frameworks that address heterogeneous CP. Late fusion simply combines agent outputs through post-processing. Calibrator (Xu et al., 2023a) enhances this approach by using calibrators to address domain gaps between heterogeneous agent outputs. End-to-end training, while effective, lacks scalability due to its requirement of re-training all agents' models. It also compromises security and task flexibility by shared fusion models and decoders. HEAL (Lu et al., 2024) improves upon this by fixing decoders and fusion models, re-training only the encoders, reducing training resources but still facing scalability issues due to the computational cost of encoder retraining as well as the security issue due to the shared fusion models and decoders. Our proposed framework, STAMP, introduces a novel approach using lightweight adapter and reverter pairs to align feature maps for collaboration. The lightweight nature of these components ensures scalability, while the maintenance of local fusion and decoders ensures both security and task agnosticism. This design effectively addresses the limitations of previous methods.

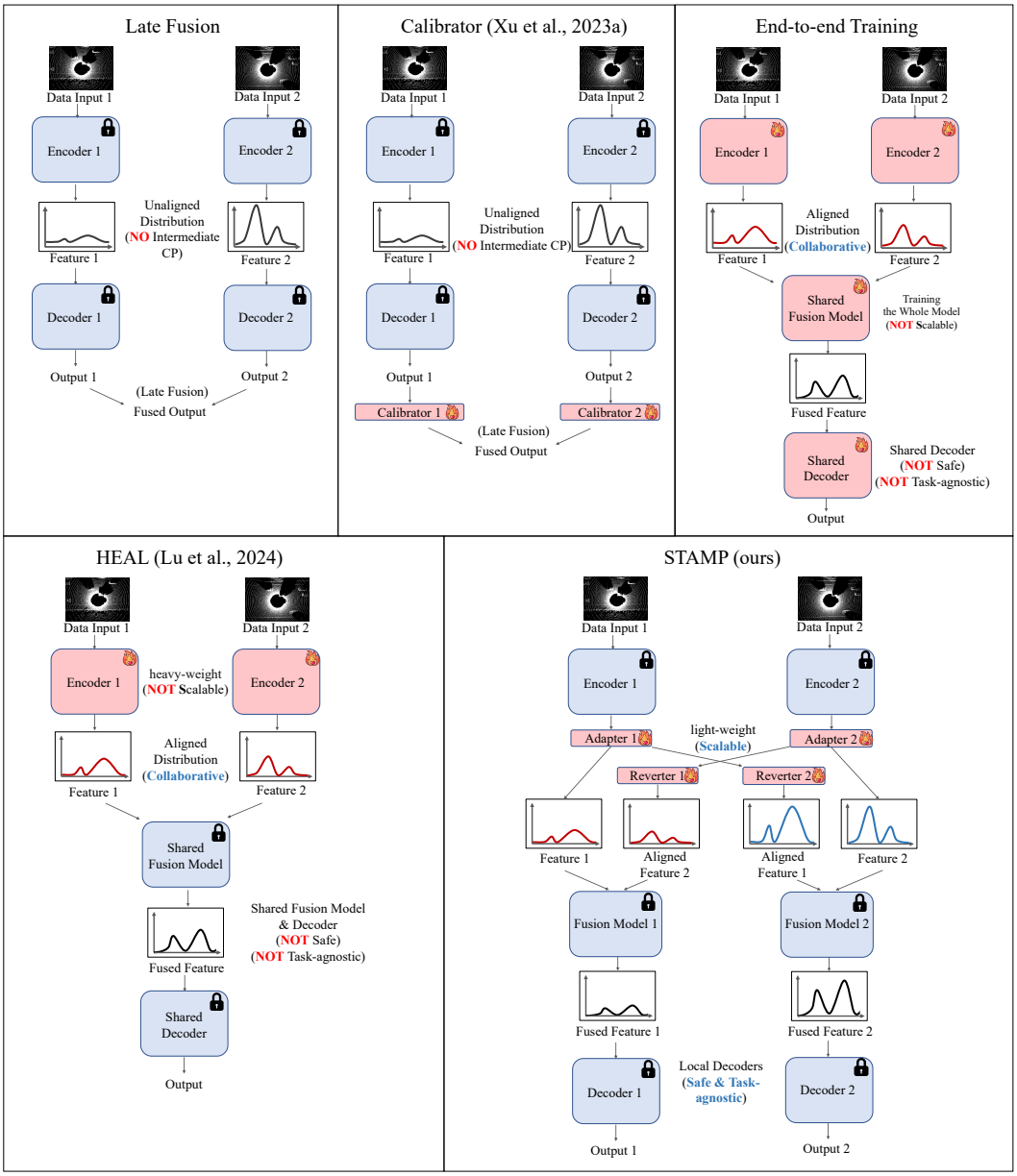

Figure A2: Architectural comparison of collaborative perception frameworks: existing approaches versus our proposed STAMP method. Blue boxes represent models with fixed parameters, while red boxes indicate models whose parameters are trained during the collaboration process.

A.3 MULTI-GROUP AND MULTI-MODEL COLLABORATIONS SYSTEM

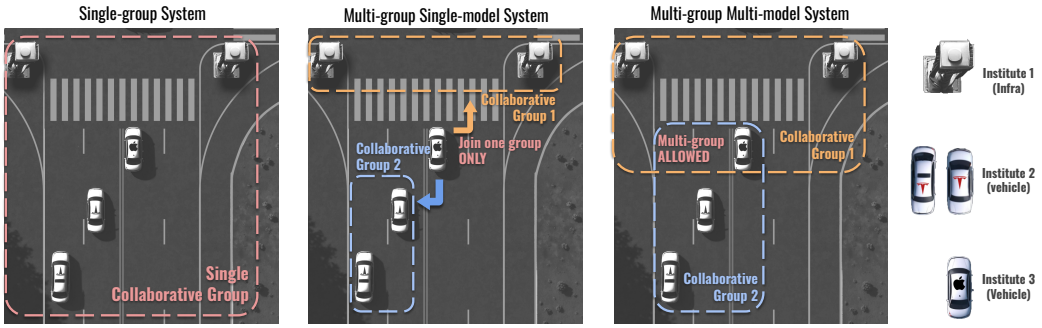

Figure A3: Comparison of collaborative perception systems: (Left) Single-group system where all agents collaborate within one group. (Middle) Multi-group single-model system allowing agents to join only one of multiple collaboration groups. (Right) Multi-group multi-model system enabling agents to participate in multiple collaboration groups simultaneously. The figure illustrates how different system architectures impact agent interactions and group formations in autonomous driving scenarios.

In our experimental findings, we observed a bottleneck effect in CP systems, where the overall system performance is constrained by the capabilities of the weakest agent. This limitation underscores the need for more selective collaboration, leading us to introduce the concept of a **Collaboration Group** - a set of agents that collaborate under specific criteria. These criteria are essential for maintaining the quality and integrity of CP, admitting agents that meet predefined standards while excluding those with inferior models, potential malicious intent, or incompatible alignments. As illustrated in Figure A3, we can distinguish between three collaborative system types:

- Single-group systems, where agents either operate independently or are compelled to collaborate with all others, are susceptible to performance bottlenecks caused by inferior agents and vulnerabilities introduced by malicious attackers.
- Multi-group single-model systems, allowing multiple collaboration groups but restricting agents to a single group because each agent can only equip a single model.
- Multi-group multi-model systems, enabling agents to join multiple groups if they meet the predefined standards.

The multi-group structure offers significant advantages over traditional single-group systems. It enhances agents' potential for diverse collaborations, consequently improving overall performance. This approach mitigates the bottleneck effect by allowing high-performing agents to maintain efficiency within groups of similar capability while potentially assisting less capable agents in other groups. Furthermore, it enhances system flexibility, enabling dynamic group formation based on specific task requirements or environmental conditions.

However, implementing such a multi-group system poses challenges for existing heterogeneous collaborative pipelines. End-to-end training approaches require simultaneous training of all models, conflicting with the concept of distinct collaboration groups. Methods like those proposed by Lu et al. (2024) require separate encoders for each group, becoming impractical as the number of groups increases due to computational and memory constraints.

Our proposed STAMP framework effectively addresses these limitations, offering a scalable solution for multi-group CP. The key innovation lies in its lightweight adapter and reverter pair (approximately 1MB) required for each collaboration group an agent joins. This efficient design enables agents to equip multiple adapter-reverter pairs, facilitating seamless participation in various groups without significant computational overhead. The minimal memory footprint ensures scalability, even as agents join numerous collaboration groups, making STAMP particularly well-suited for multi-group and multi-model collaboration systems.

A.4 More Visualization Results

Figure A4 and A5 illustrate more feature map and result visualizations before and after collaborative feature alignment (CFA). Prior to CFA, agents' feature maps exhibit disparate representations. For instance, in Figure A4, the pre-fusion feature maps of agents 1, 3, and 4 appear entirely black, indicating a significantly lower scale compared to agent 2's feature map. This discrepancy leads to instability in feature fusion. Post-CFA, the features are aligned to the same domain, resulting in more coherent fusion and accurate inference outputs.

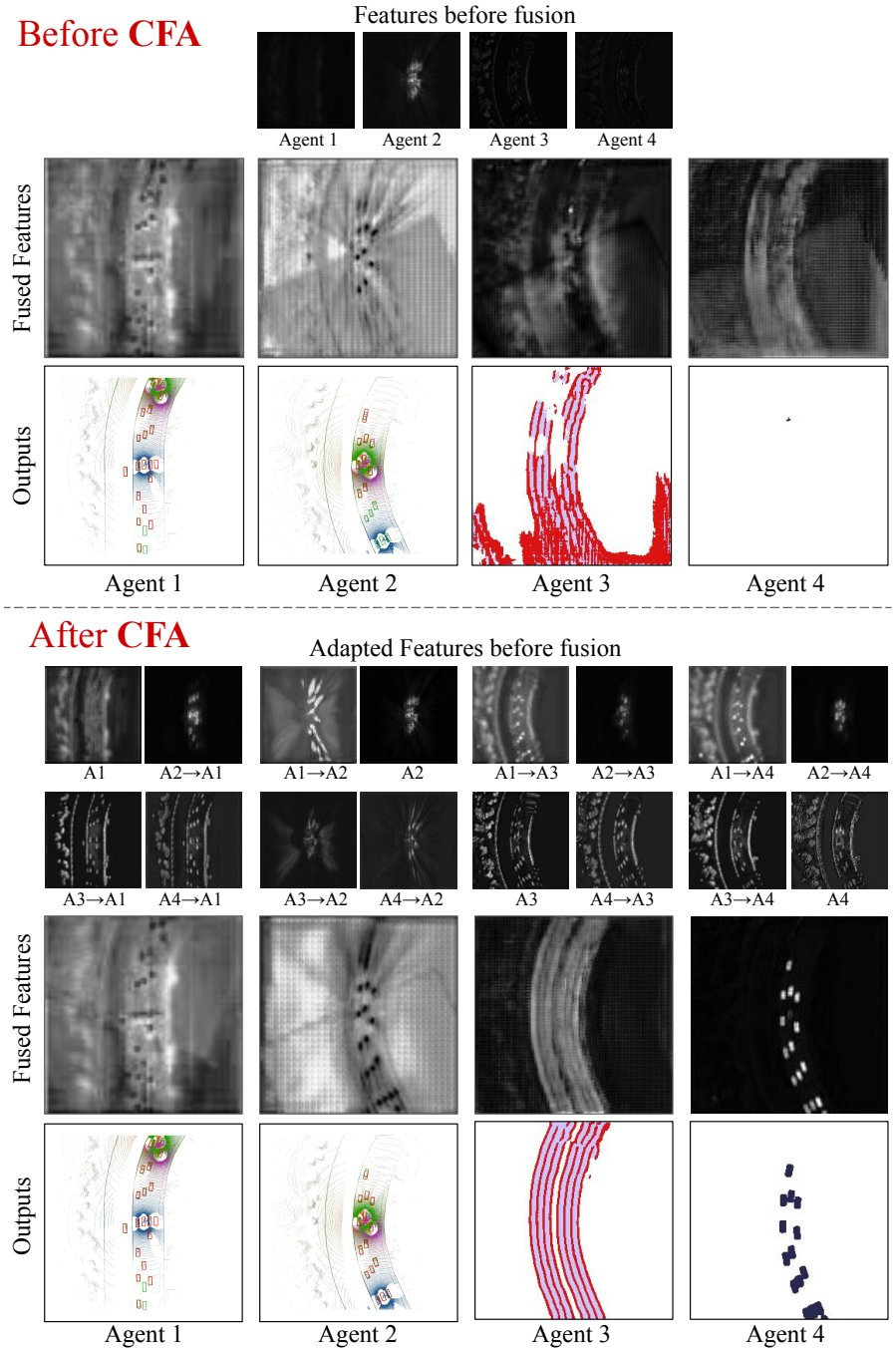

Figure A4: Visualization of feature maps and inference results before and after Collaborative Feature Alignment (CFA) in a three-agent scene. $A_i \rightarrow A_j$ denotes the feature map aligned from agent i's domain to agent j's domain, also represented as F_{ij}.

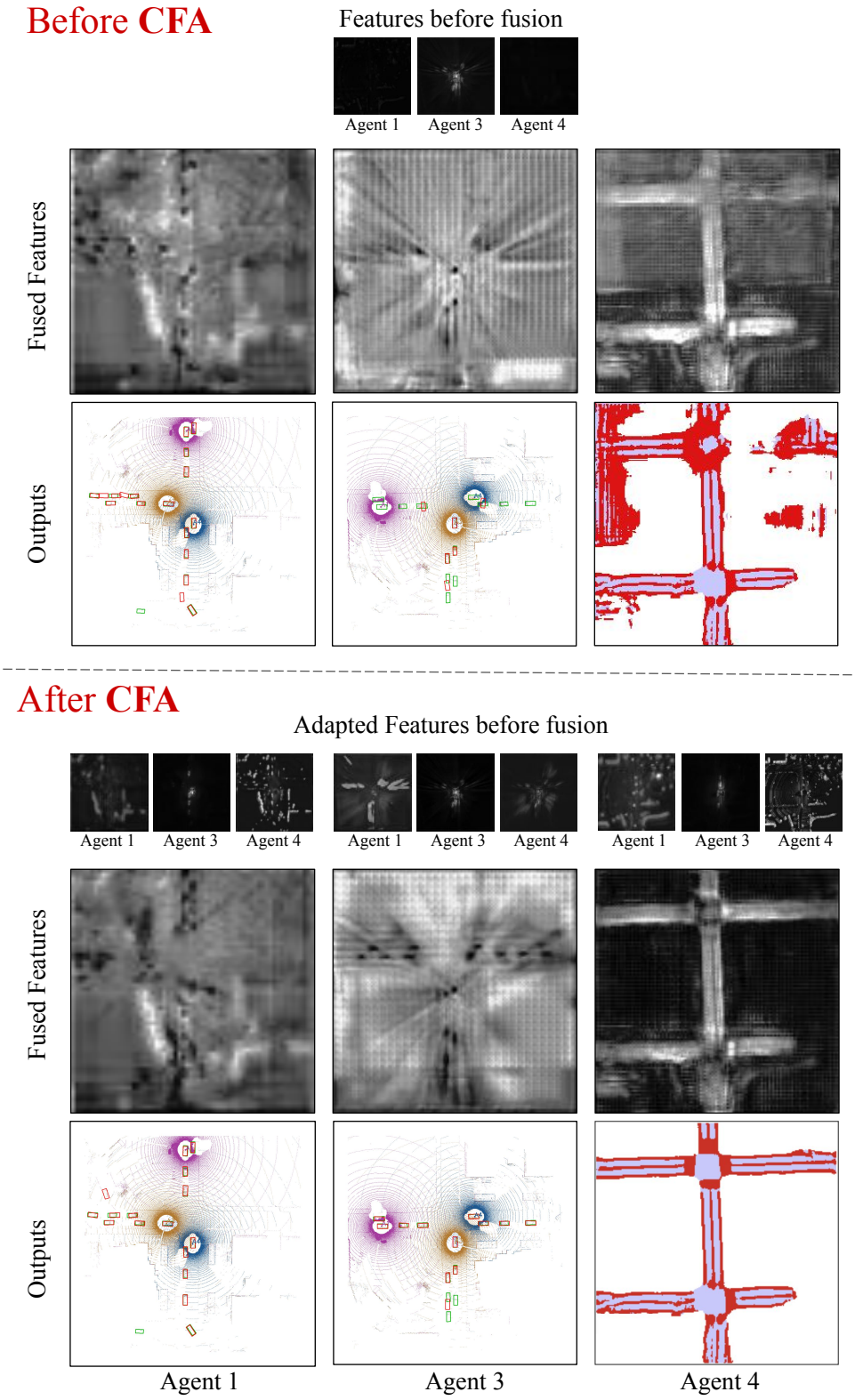

Figure A5: Visualization of feature maps and inference results before and after Collaborative Feature Alignment (CFA) in a four-agent scene. $A_i \rightarrow A_j$ denotes the feature map aligned from agent i's domain to agent j's domain, also represented as F_{ij}.