

DATA: Domain-And-Time Alignment for High-Quality Feature Fusion in Collaborative Perception

Chengchang Tian^{1*}, Jianwei Ma^{1*}, Yan Huang^{1†}, Zhanye Chen^{1†}, Honghao Wei², Hui Zhang¹, Wei Hong¹

¹ Southeast University, Nanjing, China

² Washington State University, Pullman, WA, USA

{chengchang.tian, jianwei.ma, yan.huang, chenzhanye, huizhang, weihong}@seu.edu.cn
honghao.wei@wsu.edu

Abstract

Feature-level fusion shows promise in collaborative perception (CP) through balanced performance and communication bandwidth trade-off. However, its effectiveness critically relies on input feature quality. The acquisition of high-quality features faces domain gaps from hardware diversity and deployment conditions, alongside temporal misalignment from transmission delays. These challenges degrade feature quality with cumulative effects throughout the collaborative network. In this paper, we present the **Domain-And-Time Alignment (DATA)** network, designed to systematically align features while maximizing their semantic representations for fusion. Specifically, we propose a **Consistency-preserving Domain Alignment Module (CDAM)** that reduces domain gaps through proximal-region hierarchical downsampling and observability-constrained discriminator. We further propose a **Progressive Temporal Alignment Module (PTAM)** to handle transmission delays via multi-scale motion modeling and two-stage compensation. Building upon the aligned features, an **Instance-focused Feature Aggregation Module (IFAM)** is developed to enhance semantic representations. Extensive experiments demonstrate that DATA achieves state-of-the-art performance on three typical datasets, maintaining robustness with severe communication delays and pose errors. The code will be released at <https://github.com/ChengchangTian/DATA>.

1. Introduction

Collaborative perception (CP) [2, 6, 19] has emerged as a crucial solution to overcome the inherent limitations of single-agent perception [27, 32], such as limited perception range and occluded areas. By enabling multiple agents

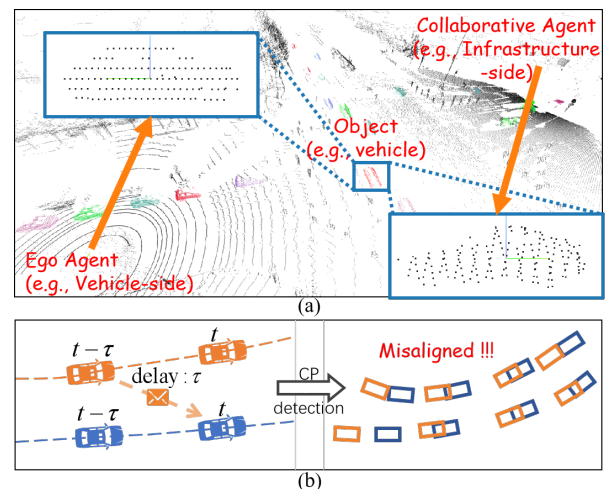


Figure 1. (a) Domain gap. E.g., different structured and scattered foreground patterns of different agents. (b) Temporal Misalignment. E.g., communication latency in the collaboration process and the followed observation misalignment between agents.

to share their own perceptual information, CP facilitates a more comprehensive understanding of surroundings.

For CP, intermediate fusion [10, 17, 20, 21, 26, 31, 35], which operates at the feature level for information sharing and integration, has been extensively studied due to its balanced trade-off between perception performance and communication bandwidth. However, it is difficult to maintain high-quality input features for fusion. In real-world deployments, obtaining high-quality features during the feature acquisition phase is fraught with significant challenges [34]. Hardware heterogeneity [16, 18, 31, 38] (e.g., LiDARs with varying numbers of laser beams and diverse modes for acquiring point clouds) and differences in agent conditions (e.g., sensor mounting heights and angles) result in distinct data distributions among agents. This divergence in data among agents creates a domain gap in CP, as illustrated in Figure 1(a). Furthermore, temporal delays [14, 27, 28, 39] are introduced during communication transmission between agents, as illustrated in Figure 1(b), causing the features of

^{1*}Equal contribution

^{2†}Corresponding authors

the same object to be misaligned. The domain and time misalignment not only significantly degrades the quality and reliability of the acquired features, but also exhibits cumulative effects throughout the collaborative system. Hence, domain and time alignment during feature acquisition is crucial to the precision and robustness of CP.

Various attempts have been made to achieve domain and time alignment during feature acquisition. For domain alignment, DI-V2X [16] achieves domain-invariant representations through a reference domain constructed by randomly mixing instances from different sources. However, this mixing approach compromises physical validity by disrupting occlusion relationships between objects in the reference domain. For temporal alignment, the methods based on global motion flow, like FFNet [39], model the entire scene to construct future frame features. However, their scene-wide optimization is biased by dominant background regions, undermining the modeling of fine-grained foreground motion patterns. In contrast, RoI-based methods, like CoBEVFlow [28], employ localized motion prediction for focused modeling. However, their reliance on region proposals impairs occlusion handling and recall, limiting the understanding of scene-wide motion dynamics.

To improve CP performance, we propose the **Domain-And-Time Alignment (DATA)** network, generating high-quality features for 3D object detection. Specifically, it focuses on learning domain-invariant and time-coherent representation for robust feature acquisition. The DATA network consists of three main modules: (i) A Consistency-preserving Domain Alignment Module (CDAM) to reduce domain gaps in the training stage. The domain gaps, within a single agent and between agents, are minimized through two complementary approaches: achieving the consistency of distance-adaptive point density while preserving physical validity via proximal-region hierarchical downsampling, and mitigating genuine domain gaps through feature-level adversarial learning under consistent observation conditions across shared regions; (ii) A Progressive Temporal Alignment Module (PTAM) to address temporal misalignment. It hierarchically captures motion patterns using multi-scale features and models complex motion through two-stage compensation to achieve scene-wide representation, and the multi-window self-supervised training strategy simultaneously enables effective foreground object motion learning and maintains global scene coherence; (iii) An Instance-focused Feature Aggregation Module (IFAM) to effectively aggregate the aligned features from multiple agents. To validate the effectiveness of DATA, we conduct extensive experiments on three typical CP datasets: DAIR-V2X-C [38], V2XSET [31], and V2XSIM [18]. Comprehensive experimental results demonstrate that our method outperforms existing state-of-the-art methods by 2.36% AP₇₀ on DAIR-V2X-C, and by 1.84% and 2.85% AP₇₀ on V2XSIM and

V2XSET. DATA also exhibits exceptional delay robustness, maintaining 75.58% AP₅₀ under 500ms communication delay, surpassing SOTA methods by 2.61%. The main contributions of this paper can be summarized as follows:

- We propose DATA, which is a new CP framework that primarily addresses challenges of feature acquisition through domain and time alignment, complemented by instance-level feature refinement to maximize the semantic representations of aligned features.
- We design CDAM, PTAM, and IFAM to reduce domain gaps, achieve temporal feature coherence, and sufficiently exploit the semantic representations of aligned features.

2. Related Work

Object Detection in Collaborative Perception. Collaborative Perception is a crucial section in autonomous driving systems. Recently, various studies [3, 5, 15, 36, 37] have explored diverse approaches to improve perception performance. Where2comm [9] selectively transmits perceptually critical features via confidence maps. CodeFilling [11] further compresses features based on codebook-based encoding to reduce transmission cost. HM-ViT [29] proposes heterogeneous 3D graph transformers to handle varying sensor configurations between agents. HEAL [22] introduces a backward alignment training mechanism to construct a unified feature space. MRCNet [8] tackles pose noise, perception noise, and motion blur through a motion-aware robust communication framework.

Domain Alignment in Collaborative Perception. Addressing the domain gap issue [34] is a crucial step in enhancing collaborative perception performance. Recent approaches address domain gaps through different mechanisms. V2X-ViT [31] addresses domain gaps by encoding different combinations of agents through specialized embeddings in its heterogeneous multi-agent self-attention module. MPDA [33] tackles domain gaps through a learnable feature resizer and sparse cross-domain transformer. DI-V2X [16] proposes a distillation framework with domain-mixing instance augmentation and progressive distillation. In this paper, we propose density-aware and region-aware training mechanisms to better bridge the domain gap between various heterogeneous agents.

Time Alignment in Collaborative Perception. Temporal synchronization is also instrumental in determining real-world deployment performance. Initial efforts to address this challenge include V2VNet [27] and V2X-ViT [31], which pioneer neural network approaches using convolutional networks and delay-aware positional encoding for delay compensation. Furthermore, SyncNet [14] extends these approach by incorporating multiple historical frames through a pyramid LSTM architecture. FFNet [39] introduces a flow-based framework to predict future features for aligned fusion. Meanwhile, CoBEVFlow [28] com-

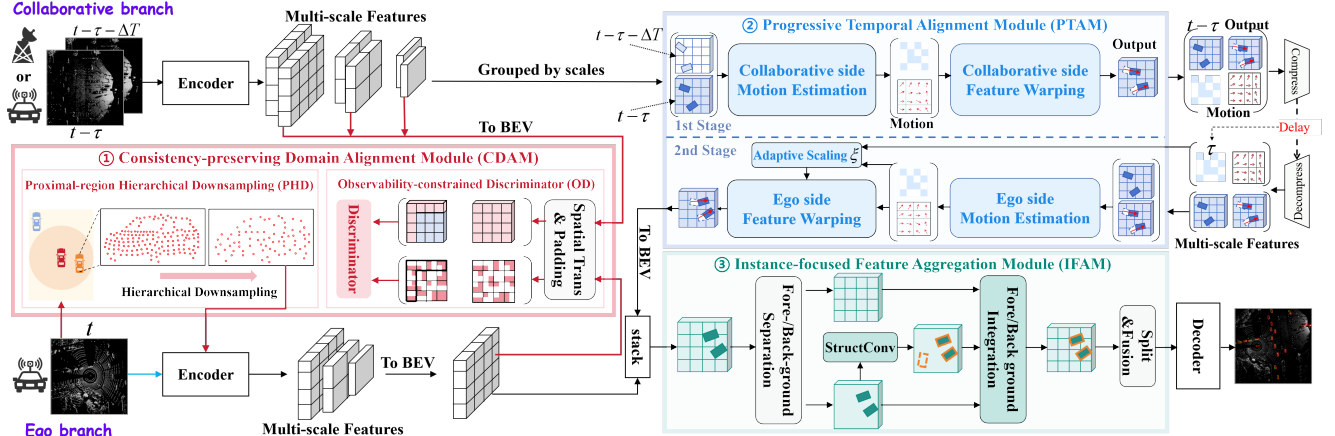


Figure 2. Overview of DATA. Arrows indicate data flow: black arrows represent streams used for both training and inference, red arrows are used only during training, and blue arrow is used only during inference. Notation table in supplementary materials aids understanding.

combines RoI-based matching with transformer-based methods for motion prediction. In this paper, we propose two-stage compensation methods to capture both fine-grained and global motion patterns. In Additional, we introduce a multi-window self-supervised strategy to better learn the motion patterns of each object within local regions.

3. Method

3.1. Problem Formulation and Overall Architecture

Our framework operates in a CP system with N agents, where each agent simultaneously functions as both a data receiver and a data provider. In the whole paper, we define the agents, which are currently receiving and transmitting data, as the ego agent and collaborative agents. And subscripts i and j denote the ego agent and collaborative agents, where $i \neq j$. At the current time t , the ego agent processes its latest data $\mathcal{X}_i(t)$, while integrating data transmitted by the collaborative agents at timestamp $t - \tau$, where τ represents the transmission delay. To compensate for this transmission delay, the collaborative agent processes its two latest frames $\mathcal{X}_j(t - \tau)$ and $\mathcal{X}_j(t - \tau - \Delta T)$ before transmission, providing both perception and motion information.

The overall architecture of DATA is shown in Figure 2.

(i) *CDAM to Align Domain (Only in Training)*: First, the point clouds of ego agent undergoes PHD of CDAM (red part) to generate multi-scale features. In collaborative branch, it processes point clouds to generate multi-scale features for two timestamps. The features of both agents are converted to BEV features, then inputted to the OD of CDAM to facilitate domain alignment between both agents. (ii) *PTAM to Align Time (in Training & Testing)*: The multi-scale features of collaborative agent pass through the first stage of PTAM (blue part). Then the output feature, along with the feature from the latest frame ($t - \tau$) and motion information, are compressed and transmitted to the ego agent. The ego agent decompresses the received data to recover

the multi-scale features and implements the second stage of PTAM to further adjust the features based on the transmission delay τ , achieving complete temporal alignment.

(iii) *IFAM to Fuse Features (in Training & Testing)*: The temporally aligned multi-scale features are converted to BEV features, subsequently fed into IFAM (green part) together with the ego’s BEV features to fuse the complementary information into a comprehensive representation. This fused representation is served as input for the detection head (decoder) to produce the final detection results.

3.2. Consistency-preserving Domain Alignment Module (CDAM)

Domain gaps from hardware heterogeneity and deployment variations manifest at raw data and feature levels. Therefore, we introduce CDAM to address these through PHD for raw data processing and OD for feature-level alignment.

3.2.1. Proximal-region Hierarchical Downsampling (PHD)

Point clouds of a single agent exhibit significant density variations with varying observation distances [34], biasing the model learning toward high-density regions and affecting the feature extraction. To address this issue, PHD is proposed to balance the density distribution of point clouds at the ego agent. PHD mainly consists of three steps.

Step 1: Proximal Object Selection. For the input scene, we define the set of all objects observable to ego agent i as $\mathcal{O}_i = \{o_1, o_2, \dots, o_{N_i^{\text{total}}}\}$, where N_i^{total} is the total number of the observable objects. Objects within the proximal region are identified based on their distance to the ego agent as

$$\mathcal{O}_i^{\text{prox}} = \{o_k \mid d_i(o_k) \leq d_{th}, k = 1, \dots, N_i^{\text{total}}\}, \quad (1)$$

where $d_i(o_k)$ is the distance from the ego agent i to the k -th object and d_{th} is the distance threshold. To maintain the distribution similarity with the original scene after downsampling, we select N_{proc} objects from $\mathcal{O}_i^{\text{prox}}$ for subsequent

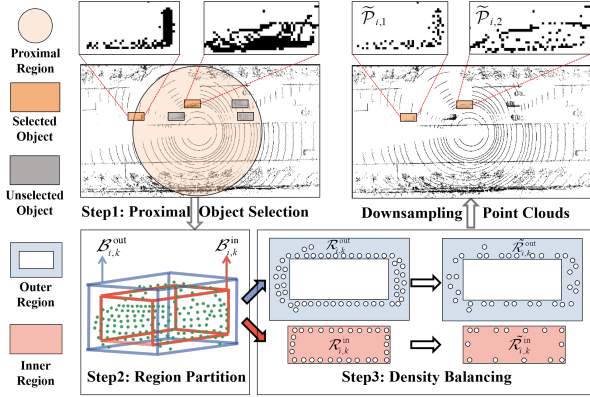


Figure 3. Pipeline of PHD. Point clouds processed by PHD simultaneously deliver contour preservation and density reduction.

processing. If $|\mathcal{O}_i^{\text{prox}}| > N_{\text{max}}$, then $N_{\text{proc}} = N_{\text{max}}$ objects are randomly selected from $\mathcal{O}_i^{\text{prox}}$, otherwise, all objects in $\mathcal{O}_i^{\text{prox}}$ are selected. Herein, $|\cdot|$ denotes the cardinality of a set and N_{max} is the predefined maximum number of objects.

Step 2: Region Partition. For each selected object, its point clouds are partitioned into inner and outer regions by using concentric bounding boxes with different scales. Let $\mathcal{B}_{i,k}^{\text{out}}$ denote the oriented bounding box of the k -th object parameterized by center coordinates (x_k, y_k, z_k) , dimensions (h_k, w_k, l_k) , and orientation θ_k . We also define a scaling factor $\alpha \in (0, 1)$ to adjust the height, width, and length of the bounding box. This creates an inner bounding box with the same center and orientation as $\mathcal{B}_{i,k}^{\text{out}}$, i.e., $\mathcal{B}_{i,k}^{\text{in}} = \{(x_k, y_k, z_k, \alpha h_k, \alpha w_k, \alpha l_k, \theta_k)\}$, where $k = 1, \dots, N_{\text{proc}}$. Then the point sets in the inner and outer regions are

$$\mathcal{R}_{i,k}^{\text{in}} = \{p | p \in \mathcal{B}_{i,k}^{\text{in}}\}, \quad \mathcal{R}_{i,k}^{\text{out}} = \{p | p \in \mathcal{B}_{i,k}^{\text{out}} \setminus \mathcal{B}_{i,k}^{\text{in}}\}. \quad (2)$$

This partition effectively separates the sparse interior points from the dense points on the object contour.

Step 3: Density Balancing. Next, the Farthest Point Sampling (FPS) [23] is applied with different ratios to the partitioned regions. For the inner region, a high downsampling ratio β_{in} is used while a conservative downsampling ratio β_{out} is applied to the outer region with more points left, thus capturing the object contour and preserving its geometric details. The downsampling can be formulated as

$$\tilde{\mathcal{R}}_{i,k}^{\text{in}} = \text{FPS}(\mathcal{R}_{i,k}^{\text{in}}, \beta_{\text{in}}), \quad \tilde{\mathcal{R}}_{i,k}^{\text{out}} = \text{FPS}(\mathcal{R}_{i,k}^{\text{out}}, \beta_{\text{out}}). \quad (3)$$

Finally, the point clouds of the k -th object, combining both inner and outer regions, can be formulated as $\tilde{\mathcal{P}}_{i,k} = \tilde{\mathcal{R}}_{i,k}^{\text{in}} \cup \tilde{\mathcal{R}}_{i,k}^{\text{out}}, k = 1, \dots, N_{\text{proc}}$. Through this hierarchical downsampling, PHD enhances density-consistent representations for varying distances and preserves critical information of the original data, particularly occlusion relationships and geometric structures.

3.2.2. Observability-constrained Discriminator (OD)

In CP, the domain gaps between agents arise from both sensor-intrinsic properties and observation characteristics.

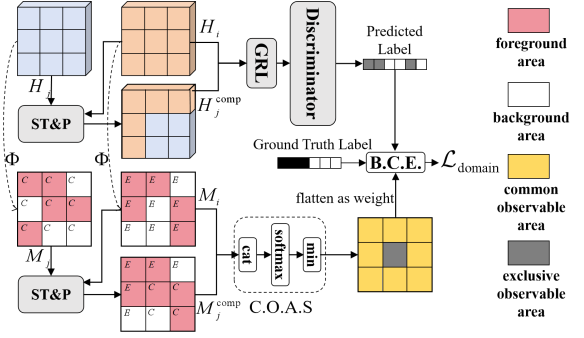


Figure 4. Pipeline of OD. ST&P denotes Spatial Transformation and Padding and C.O.A.S. refers to Common Observable Area Selection. Genuine domain gaps are located by common observable areas, enabling intrinsic domain-invariant feature learning.

To align these domain gaps, it is crucial to identify the regions where both agents maintain valid observations. To address this issue, we propose an OD module to explicitly incorporate observability into the domain alignment process.

First, we define the observations of ego agent i as the ego domain, while the observations of $N - 1$ collaborative agents are the collaborative domain. For domain discrimination, one collaborative agent is randomly selected for domain alignment. This enables diverse agent combinations that facilitate domain-invariant feature learning.

Then, through a foreground estimator $\Phi(\cdot)$, the OD uses Bird's Eye View (BEV) features H_i and H_j of ego and collaborative agents to generate observability maps $M_i = \Phi(F_i)$ and $M_j = \Phi(F_j)$, where $M_i, M_j \in \mathbb{R}^{1 \times H \times W}$, for indicating the observability at each spatial location.

Subsequently, we project the collaborative features onto the coordinates of ego agent through spatial transformation, creating $H_{j \rightarrow i}$ and $M_{j \rightarrow i}$. However, this may potentially produce void regions. Let \mathcal{V} denote the set of valid grids in the transformed feature map and the voids are filled as

$$H_j^{\text{comp}} = \mathcal{I}_{\mathcal{V}} \cdot H_{j \rightarrow i} + (1 - \mathcal{I}_{\mathcal{V}}) \cdot H_i, \quad (4)$$

$$M_j^{\text{comp}} = \mathcal{I}_{\mathcal{V}} \cdot M_{j \rightarrow i} + (1 - \mathcal{I}_{\mathcal{V}}) \cdot M_i, \quad (5)$$

where H_j^{comp} and M_j^{comp} represent the complemented feature and observability map, respectively, and the indicator function $\mathcal{I}_{\mathcal{V}}$ equals to 1 for points in \mathcal{V} and 0 for the others. To ensure the domain alignment focus on the regions with shared observability of both agents, an observability weighting map $W \in \mathbb{R}^{1 \times H \times W}$ is computed as $W = \min(\text{softmax}([M_i, M_j^{\text{comp}}]))$, where $[\cdot]$ denotes concatenation and all operations are performed along the first dimension. Finally, the domain alignment objective is

$$\max_{\theta} \min_{\mu} \mathcal{L}_{\text{domain}} = \frac{1}{\sum_{sp \in \mathcal{S}} W_{\text{flat}}^{sp}} \sum_{sp \in \mathcal{S}} W_{\text{flat}}^{sp} \cdot \mathcal{L}_{\text{BCE}}(D_{\mu}^{sp}(\Psi_{\theta}), Z^{sp}), \quad (6)$$

where \mathcal{S} denotes the set of all spatial positions with sp denoting one position, W_{flat} denotes the flattened observability weighting map, Ψ_{θ} is the feature extractor (point clouds as

input and BEV features as output), D_μ denotes the discriminator, \mathcal{L}_{BCE} is the binary cross-entropy loss, and Z is the domain label (0 for ego agent and 1 for collaborative agent).

To jointly optimize this min-max problem, a gradient reversal layer (GRL) [4] is inserted before the discriminator. The GRL leaves the input unchanged in the forward pass and applies a negative scaling factor $\gamma = -0.1$ during backpropagation. This allows the end-to-end adversarial training, where the discriminator learns to distinguish domains in the regions with shared observability and the feature extractor learns to generate domain-invariant features that match physical constraints of multi-agent perception.

3.3. Progressive Temporal Alignment Module (PTAM)

3.3.1. Kinematic Perspective of Temporal Alignment

Temporal asynchrony causes feature misalignment between agents, presenting a fundamental challenge in multi-agent CP. To compensate the temporal misalignment of collaborative features at the historical time $t - \Delta t$, we leverage a kinematic perspective [41, 42]. The temporal evolution of any-scale features at collaborative agent is formulated as

$$F_j(t, \mathbf{x} + \mathbf{v}(t - \Delta t, \mathbf{x})\Delta t) = F_j(t - \Delta t, \mathbf{x}), \quad (7)$$

where Δt is a time interval (within the range of typical transmission delay in CP [1, 13, 14]), $F_j(t - \Delta t, \mathbf{x})$ denotes the features at position \mathbf{x} of time $t - \Delta t$, and $\mathbf{v}(t - \Delta t, \mathbf{x})$ denotes the velocity field describing the motion of features at time $t - \Delta t$. This captures that features of one position at time t can be obtained by tracing back along the velocity field to the corresponding positions of time $t - \Delta t$.

Following the kinematic formulation, the core mechanism of PTAM realizes feature temporal evolution through three essential components: motion field $\Delta p \in \mathbb{R}^{2 \times H \times W}$ representing \mathbf{v} , temporal scaling factor ξ corresponding to Δt , and displacement compensation operation via bilinear sampling. In the following, we omit \mathbf{x} for simplicity.

3.3.2. Two-Step Implementation for Both Agents

Based on this core mechanism, the progressive temporal alignment strategy of PTAM consists of a two-stage prediction executed serially. Each stage corresponds to one agent.

(i) First Stage. The collaborative agent captures historical motion patterns by utilizing its latest two frames of features $F_j(t - \tau - \Delta T)$ and $F_j(t - \tau)$ to predict an intermediate feature F_j^{inter} .

(ii) Second Stage. The ego agent performs adaptive temporal alignment based on received motion information and transmission delay characteristics. Due to the potential communication error between two agents, the received features are defined as $\hat{F}_j(t - \tau)$ and \hat{F}_j^{inter} , which are used to generate temporally aligned features $\hat{F}_j(t)$. To capture hierarchical motion patterns of varying granularities, PTAM

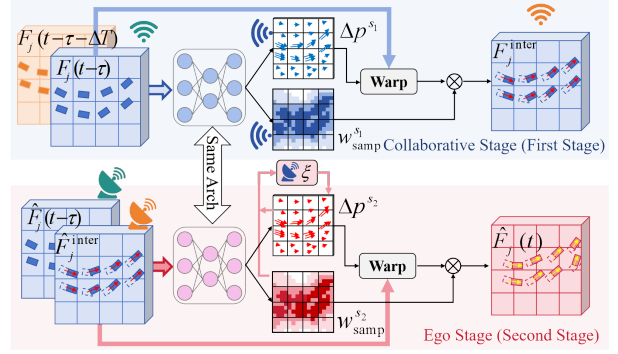


Figure 5. Pipeline of PTAM. One spatial scale is illustrated here. Three couples of ‘WiFi’ and ‘Radar’ symbols in different colors denote transmitting and receiving information between agents. Both historical motion tendency and scene dynamics are exploited via two-stage prediction, enabling precise latency compensation.

processes features simultaneously at three spatial scales.

Each Stage Implementation. At each stage with each spatial scale, we execute the core mechanism through two steps, *i.e.*, motion estimation and feature warping.

Step 1: Motion Estimation. This step shares the same architecture for both stages. For a united representation of both stages, we denote the input adjacent temporal features as the latest feature F_{latest} and its previous feature F_{prev} . The motion estimation begins with computing the temporal feature difference, defined as $\Delta F = F_{\text{latest}} - F_{\text{prev}}$, which is then separately concatenated with these two input features. These concatenated features undergo a sequence of operations to generate the motion field Δp and the sampling weight $w_{\text{samp}} \in \mathbb{R}^{1 \times H \times W}$.

Step 2: Feature Warping. This step is slightly different for the two stages. For the first stage executed at the collaborative agent, a unit temporal scaling factor, *i.e.*, $\xi = 1$, is employed to maintain consistency with the sampling period of sensor, *e.g.*, LIDAR, generating an intermediate feature representation $F_j^{\text{inter}} = w_{\text{samp}}^{s_1} \odot f_{\text{warp}}(F_j(t - \tau), \Delta p^{s_1})$, where s_1 indicates the first stage and $f_{\text{warp}}(\cdot)$ denotes the bilinear sampling operation. For the second stage at the ego agent, an adaptive temporal scaling factor ξ is required to handle the variable time intervals resulting from transmission delays. The motion difference field ΔM is defined as

$$\Delta M = (\Delta p^{s_2} \odot w_{\text{samp}}^{s_2}) - (\Delta p^{s_1} \odot w_{\text{samp}}^{s_1}), \quad (8)$$

where the superscript s_2 indicates the second stage and the element-wise product captures the effective motion at each stage by incorporating Δp and the corresponding w_{samp} . Then, to obtain a global context vector f_M , we use cascaded convolutional layers, residual blocks, and global pooling operations to process ΔM . And a temporal encoding f_T is obtained by adding sinusoidal positional embeddings, which capture transmission delay characteristics, to f_M . Finally, these features are concatenated and processed through

a MLP to predict the temporal scaling factor as

$$\xi = \text{ReLU}(\text{MLP}([f_M, f_T])). \quad (9)$$

And the final temporally aligned feature is acquired through $\hat{F}_j(t) = w_{\text{samp}}^{s_2} \odot f_{\text{warp}}(\hat{F}_j^{\text{inter}}, \Delta p^{s_1})$.

3.3.3. Multi-window Self-supervised Training Strategy.

To enhance the capability of PTAM in capturing diverse motion patterns through progressive-parallel alignment, we propose a multi-window self-supervised training strategy.

At each spatial scale s , the feature plane of size $h_s \times w_s$ is partitioned using two complementary window partitioning strategies. These two window sets are defined as

$$W_1 = \{w_{m',n'} \mid 0 \leq m' < \frac{h_s}{l}, 0 \leq n' < \frac{w_s}{l}\}, \quad (10)$$

$$W_2 = \{w_{p',q'} \mid 0 \leq p' < \frac{h_s}{l} - 1, 0 \leq q' < \frac{w_s}{l} - 1\}, \quad (11)$$

where $w_{m',n'}$ and $w_{p',q'}$ denote windows with top-left corner at position (m', n') and (p', q') with size $l \times l$, respectively. Herein, W_1 actually start partitioning features exactly from boundaries while W_2 has an offset of $l/2$ compared with W_1 . For each window w in both sets, we compute the cosine similarity $\cos(\cdot, \cdot)$ between predictions and ground truth features F_j^{gt} . The loss functions for intermediate and final predictions at scale s can be written as

$$\mathcal{L}_{\text{inter}}^s = \frac{1}{N_{\text{window}}} \sum_{w \in W_1 \cup W_2} \|1 - \cos(F_{j,w}^{\text{inter},s}, F_{j,w}^{\text{gt},s}(t))\|_2^2, \quad (12)$$

$$\mathcal{L}_{\text{final}}^s = \frac{1}{N_{\text{window}}} \sum_{w \in W_1 \cup W_2} \|1 - \cos(\hat{F}_{j,w}^s(t), F_{j,w}^{\text{gt},s}(t))\|_2^2, \quad (13)$$

where N_{window} denotes the total number of windows. Finally, the total temporal alignment loss is computed across all three scales, given by $\mathcal{L}_{\text{temporal}} = \sum_{s=1,2,3} (\mathcal{L}_{\text{inter}}^s + \mathcal{L}_{\text{final}}^s)$.

3.4. Instance-focused Feature Aggregation Module (IFAM)

Suppose that domain-and-time aligned features are obtained by CDAM and PTAM, but how to fully exploit the semantic information of these aligned features is still crucial for improving perception performance. Herein, the IFAM is designed to enhance the structural representation of foreground objects for robust CP. Suppose that $H_a \in \mathbb{R}^{C \times H \times W}$ is the BEV feature of the a -th agent, its foreground and background features are identified as $H_a^{\text{fore}} = H_a \odot M_a$, $H_a^{\text{back}} = H_a \odot (\mathbf{1} - M_a)$, where $M_a = \Phi(H_a)$ is the foreground mask and $\Phi(\cdot)$ is the foreground estimator mentioned in Section 3.2.2 (OD module). To strengthen the structural details in foreground regions, a set of 3×3 convolution kernels is applied. Then we have $H_a^{\text{enh}} = \text{StructConv}(H_a^{\text{fore}})$, where StructConv consists of one vanilla convolution to preserve basic feature

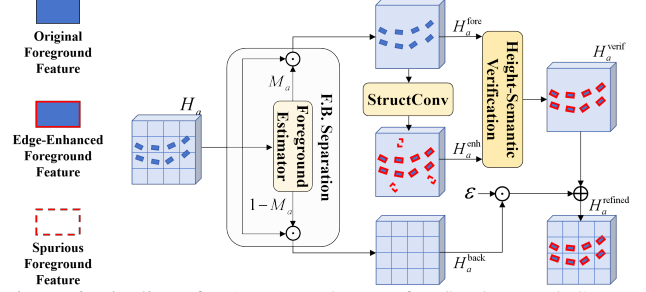


Figure 6. Pipeline of IFAM. F.B. denotes fore/background. Structural enhancement benefits from F.B. separation, reducing noise and improving detection via height-semantic verification.

intensity and four specialized convolutions: a central difference convolution, horizontal and vertical difference convolutions, and an angular difference convolution. Although this strengthens the structural details, it may potentially occur false targets that interfere with subsequent detection.

To suppress these spurious foreground features while preserving enhanced structural details, a foreground verification mechanism is proposed based on the pillar-encoding [12, 25] since its each channel dimension inherently contains coupled height-semantic information, which can be used to select the correct foreground features. Specifically, given the original and enhanced foreground features (H_a^{fore} and H_a^{enh}), their concatenation along the channel dimension is $H_a^{\text{cat}} = [H_a^{\text{fore}}, H_a^{\text{enh}}] \in \mathbb{R}^{2C \times H \times W}$. Then spatial and channel attention modules are employed in parallel. For spatial attention W_a^s , max- and average-pooling operations are performed across the channel dimension followed by concatenation and convolution. For channel attention W_a^c , spatial average pooling is applied followed by two convolutions. Then the initial attention weights W_a^{init} are derived as $W_a^{\text{init}} = W_a^s \oplus W_a^c$. And the verification weights are

$$W_a^{\text{verif}} = \text{Sigmoid}(\text{GConv}(\text{CS}([H_a^{\text{cat}}, W_a^{\text{init}}])), \quad (14)$$

where CS denotes channel shuffle [40] that promotes cross-group information exchange, breaking fixed channel combinations for comprehensive feature verification. And group convolution $\text{GConv}(\cdot)$ [7] enables independent weight generation for different feature groups, allowing specialized verification of height-semantic patterns. Through this verification mechanism, the regions exhibiting height-semantic relationships (e.g., those consistent with typical vehicle shapes and heights) are preserved, while the regions containing unnatural representation combinations of spurious foreground features are mitigated. Then the verified foreground feature of the k -th agent ultimately is expressed as

$$H_a^{\text{verif}} = \text{Conv}_{1 \times 1}((W_a^{\text{verif}} \odot H_a^{\text{fore}} \oplus (\mathbf{1} - W_a^{\text{verif}}) \odot H_a^{\text{enh}}) \oplus H_a^{\text{fore}} \oplus H_a^{\text{enh}}). \quad (15)$$

The individually refined BEV features are obtained by combining background features as $H_a^{\text{refined}} = H_a^{\text{verif}} \oplus \epsilon H_a^{\text{back}}$,

Method Metric(AP ₅₀ /AP ₇₀)	V2XSim	V2XSet	DAIR-V2X-C
DiscoNet (NeurIPS'21) [17]	83.56 / 66.12	82.34 / 64.79	68.50 / 53.57
AttFuse (ICRA'22) [32]	81.70 / 66.24	84.37 / 66.27	67.36 / 52.96
V2X-VIT (ECCV'22) [31]	82.32 / 64.41	82.42 / 63.14	71.54 / 51.65
CoBEVT (CoRL'22) [30]	81.00 / 65.06	84.84 / 65.14	69.21 / 46.66
Where2com (NeurIPS'22) [9]	83.82 / 65.52	85.19 / 61.59	68.39 / 52.48
AdaFusion (WACV'23) [24]	78.89 / 58.62	86.28 / 57.06	71.16 / 47.74
HM-VIT (ICCV'23) [29]	- / -	- / -	76.10 / -
CoBEVFlow (NeurIPS'23) [28]	- / -	- / -	73.80 / 59.90
FFNet (NeurIPS'23) [39]	85.56 / 68.64	83.57 / 66.23	77.19 / 60.17
DI-V2X (AAAI'24) [16]	- / -	- / -	78.82 / -
MRCNet (CVPR'24) [8]	85.33 / 69.82	85.00 / 66.31	- / -
CodeFilling (CVPR'24) [11]	- / -	- / -	79.90 / 61.06
HEAL (ICLR'24) [22]	88.67 / 75.85	88.40 / 72.13	79.00 / 63.12
DATA (Ours)	88.91 / 77.69	89.53 / 74.98	79.94 / 65.48

Table 1. Performance comparison without communication latency.

where ϵ is a learnable parameter balancing the contribution of background features. And the refined features of all N agents are progressively fused through a shared 1×1 convolution to produce the ultimate CP representation.

4. Experiments

4.1. Datasets and Evaluation Metrics

Experiments are conducted on three datasets: DAIR-V2X-C [38], V2XSET [31], and V2XSIM [18]. Detailed dataset information are presented in Section 2.1 of Supplementary Material. For evaluation, Average Precision (AP) is measured at Intersection-over-Union (IoU) thresholds of 0.50 and 0.70 for the car category on all datasets.

4.2. Implementation

The backbone employs PointPillar [12] architecture with a grid size of $0.4m \times 0.4m$. In CDAM, distance threshold $d_{th} = 50$ m and $N_{max} = 2$ vehicles are downsampled. The two downsampling ratios are $\beta_{in} = 0.6$ and $\beta_{out} = 0.8$. For PTAM, the multi-window self-supervised training adopts a window size of $l = 16$. The training procedure comprises three sequential stages: detection model training, PTAM training, and transmission module training, which are presented in Section 2 of Supplementary Material.

4.3. Quantitative Results

Performance without Latency. Table 1 presents performance comparison with SOTA methods under zero latency setting. DATA achieves 0.94% and 2.36% improvements in AP₅₀ and AP₇₀ compared to HEAL [22] on real-world DAIR-V2X-C validation set. For simulation datasets V2XSIM and V2XSET, where Gaussian noise ($\mu_{pos} = 0, \sigma_{pos} = 0.2$ m for position, $\mu_{rot} = 0, \sigma_{rot} = 0.2^\circ$ for orientation) is added to mimic real-world conditions, DATA also demonstrates significant improvements. Specifically, it outperforms MRCNet, by 3.58% and 7.87% in AP₅₀ and AP₇₀ on V2XSIM, and by 4.53% and 8.67% on V2XSET.

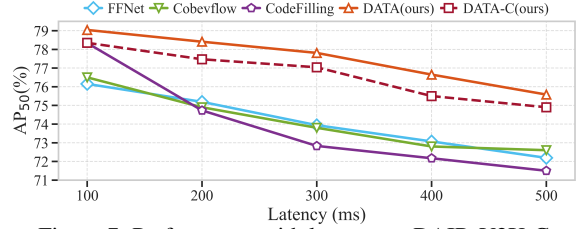


Figure 7. Performance with latency on DAIR-V2X-C.

These consistent improvements across both complex simulated and real-world scenarios demonstrate the effectiveness of DATA, which first achieves domain-invariant feature extraction through CDAM to enhance feature quality, then leverages IFAM to effectively highlight foreground feature semantics for improved CP system performance.

Performance with Latency. As shown in Figure 7, DATA effectively maintains high-quality CP under temporal asynchrony on DAIR-V2X-C dataset. Building upon the strong feature extraction and fusion capabilities in zero-latency scenarios, the PTAM further ensures reliable feature prediction during transmission with varying delays, realizing the minimal 3.46% AP₅₀ decrease as latency increases from 100ms to 500ms. And DATA maintains 75.58% AP₅₀ at 500ms delay, while baseline methods, like FFNet (72.19%) and CodeFilling (71.50%), present more significant drops. Also, DATA achieves robust performance even with compressed transmission (74.90% AP₅₀ at 500ms of DATA-C). These results demonstrate that DATA effectively addresses temporal challenges in real-world CP by acquiring and utilizing high-quality features.

Robustness to Pose Errors. To investigate the effects of different pose noise components on CP performance, experiments are conducted on V2XSET and V2XSIM. Gaussian noise with varying standard deviations is separately applied to collaborator positions and orientations, with localization noise σ_{loacl} ranging from 0.2 m to 0.6 m and heading noise σ_{head} ranging from 0.2° to 1.0° . As shown in Figure 8, the compared methods suffer from significant performance degradation. Their AP₇₀ declines rapidly with increasing noise magnitude, dropping by over 30% and 9% when σ_{local} and σ_{head} reaches 0.6 m and 1.0° , respectively, on both V2XSIM and V2XSET. This indicates that spatial feature misalignment severely impacts their CP accuracy. DATA demonstrates robust resilience against pose uncertainties on both datasets: on V2XSIM, it maintains 66.53% AP₇₀ at $\sigma_{local} = 0.6$ m and 75.23% AP₇₀ at $\sigma_{head} = 1.0^\circ$; similarly on V2XSET, it achieves 67.22% AP₇₀ at $\sigma_{local} = 0.6$ m and 71.17% AP₇₀ at $\sigma_{head} = 1.0^\circ$. This robustness stems from two complementary aspects. First, the observability-guided domain alignment learns domain-invariant features from consistent observation, extracting essential geometric patterns from noisy observations. Meanwhile, the instance-focused feature enhancement, through foreground-background separation and semantic refinement, strength-

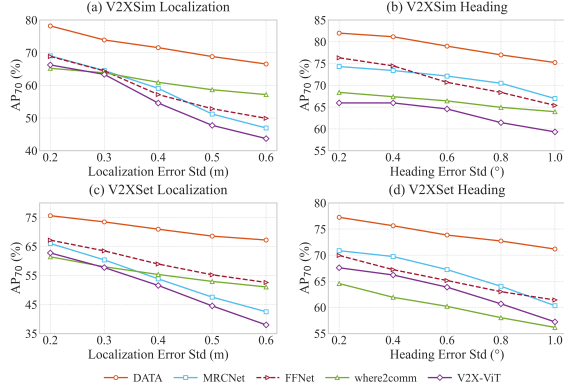


Figure 8. Performance comparison with pose errors.

ens the structural completeness of object representations, ensuring reliable detection under pose perturbations.

4.4. Ablation Study

Ablation on Domain Alignment and Feature Aggregation. To investigate the contribution of each module, experiments are conducted on DAIR-V2X-C validation set without latency. As shown in Table 2, compared with the baseline model, incorporating IFAM brings significant performance gains, *i.e.*, 3.70% improvement in AP₇₀, since its geometric-aware feature enhancement strengthens structural completeness and enables more precise object localization. Adding OD further improves the detection performance by 1.96% AP₅₀ and 0.62% AP₇₀ since it guides the feature alignment to focus on regions where both agents maintain comparable observation capabilities, fundamentally addressing cross-domain feature discrepancies. The integration of PHD yields 0.96% and 1.25% improvements in AP₅₀ and AP₇₀. This demonstrates that maintaining consistent distribution of point clouds across different spatial regions, while preserving inherent scene structures, is crucial for extracting reliable domain-invariant features.

Ablation on Proximal Region Threshold. The PHD module aims to balance point clouds density across different spatial regions while preserving scene structure. The results in Table 4 demonstrate that the 50 *m* threshold achieves optimal performance (79.94% AP₅₀), representing a good trade-off between density balancing and distribution preservation. Smaller thresholds (10 *m*, 30 *m*) provide insufficient coverage of proximal objects, while larger thresholds (70 *m*, 90 *m*) may introduce excessive downsampling that disrupts the original scene structures.

Ablation on Progressive Temporal Alignment. To investigate the impact of window size on temporal alignment performance, experiments are conducted, comparing First-Stage and Two-Stage models, on DAIR-V2X-C validation set with 300 ms communication delay. Table 3 shows that the Two-stage model outperforms the First-stage model for all window sizes, where the Two-Stage model achieves optimal performance (77.81% AP₅₀) with window size 16,

Config	AP ₅₀	AP ₇₀	Config	AP ₅₀	AP ₇₀
Baseline	76.80	59.91	+ P. + O.	78.96 ^{+2.16}	62.68 ^{+2.77}
+ P.	77.54 ^{+0.74}	62.16 ^{+2.25}	+ P. + I.	78.30 ^{+1.50}	63.65 ^{+3.74}
+ O.	78.52 ^{+1.72}	62.58 ^{+2.67}	+ O. + I.	78.98 ^{+2.18}	64.23 ^{+4.32}
+ I.	77.02 ^{+0.22}	63.61 ^{+3.70}	+ P. + O. + I.	79.94^{+3.14}	65.48^{+5.57}

Table 2. Comprehensive ablation study of CDAM and IFAM on DAIR-V2X-C. P.: PHD, O.: OD, I.: IFAM.

Model Type (AP ₅₀ %)	w/o PTAM	Window Size <i>l</i> (AP ₅₀ %)				
		w.m.	32	16	8	4
First-Stage (Collab.)	71.65	76.23	76.68	76.80	76.62	76.54
Two-Stage (Ego & Collab.)	71.65	77.14	77.62	77.81	77.59	76.64

Table 3. Performance on 300 ms latency of Two model types across different window sizes. “w.m.” represents whole map supervision.

Distance Threshold d_{th}	10m	30m	50m	70m	90m
Metric (AP ₅₀ %)	79.33	79.44	79.94	79.18	78.69

Table 4. Ablation of proximal-region distance thresholds (d_{th}). surpassing the whole map supervision and the First-Stage approach by 0.67% and 1.01%, respectively. The performance degrades when increasing or decreasing the window size, indicating that this intermediate window size optimally balances local motion capture and contextual information preservation. Especially with small window sizes, like 4, both models face challenges since small windows cannot adequately capture the vehicle objects. And the Two-Stage model shows more significant performance degradation (76.64% AP₅₀ at size 4) compared to the One-Stage approach (76.54% AP₅₀) due to the additional impact of error accumulation through prediction stages. These findings validate that Two-Stage progressive alignment can achieve temporal coherence through intermediate prediction states when trained with appropriate window partitions.

5. Conclusion

In this paper, we propose DATA, which is a novel framework that addresses the fundamental domain and time misalignment in feature-level collaborative perception. We develop systematic solutions for feature alignment during the acquisition phase. The proposed CDAM reduces domain gaps in feature extraction through density-aware point cloud sampling and observability-guided domain alignment, while PTAM ensures temporal coherence during feature transmission through progressive motion refinement. Building upon well-aligned features, IFAM further enhances their semantic expressiveness to maximize fusion performance. Extensive experiments on both real-world and simulation datasets demonstrate the effectiveness of DATA, achieving SOTA performance under various challenging conditions, validating the robustness of our systematic approach on feature alignment and enhancement in CP.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62271142 and U2341206, in part by the Nanjing U35 Strong Foundation Engineering under Grant 4204002501, in part by the Key Research and Development Program of Jiangsu Province under Grant BE2023021 and BE2023011-2.

References

- [1] Giuseppe Araniti, Claudia Campolo, Massimo Condoluci, Antonio Iera, and Antonella Molinaro. Lte for vehicular networking: A survey. *IEEE communications magazine*, 51(5): 148–157, 2013. 5
- [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. *IEEE Transactions on Intelligent Transportation Systems*, 2024. 1
- [3] Ziming Chen, Yifeng Shi, and Jinrang Jia. Transiff: An instance-level feature fusion framework for vehicle-infrastructure cooperative 3d detection with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18205–18214, 2023. 2
- [4] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 5
- [5] Xiangbo Gao, Runsheng Xu, Jiachen Li, Ziran Wang, Zhiwen Fan, and Zhengzhong Tu. Stamp: Scalable task and model-agnostic collaborative perception. *arXiv preprint arXiv:2501.18616*, 2025. 2
- [6] 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*, 15(6):131–151, 2023. 1
- [7] 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*, pages 770–778, 2016. 6
- [8] Shixin Hong, Yu Liu, Zhi Li, Shaohui Li, and You He. Multi-agent collaborative perception via motion-aware robust communication network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15301–15310, 2024. 2, 7
- [9] 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, 2022. 2, 7
- [10] Yue Hu, Yifan Lu, Runsheng Xu, Weidi Xie, Siheng Chen, and Yanfeng Wang. Collaboration helps camera overtake lidar in 3d detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2023. 1
- [11] Yue Hu, Juntong Peng, Sifei Liu, Junhao Ge, Si Liu, and Siheng Chen. Communication-efficient collaborative perception via information filling with codebook. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15481–15490, 2024. 2, 7
- [12] 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*, pages 12697–12705, 2019. 6, 7
- [13] Kwongjong Lee, Joonki Kim, Yosub Park, Hanho Wang, and Daesik Hong. Latency of cellular-based v2x: Perspectives on tti-proportional latency and tti-independent latency. *Ieee Access*, 5:15800–15809, 2017. 5
- [14] Zixing Lei, Shunli Ren, Yue Hu, Wenjun Zhang, and Siheng Chen. Latency-aware collaborative perception. In *European Conference on Computer Vision*, pages 316–332. Springer, 2022. 1, 2, 5
- [15] Zixing Lei, Zhenyang Ni, Ruize Han, Shuo Tang, Chen Feng, Siheng Chen, and Yanfeng Wang. Robust collaborative perception without external localization and clock devices. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7280–7286. IEEE, 2024. 2
- [16] Xiang Li, Junbo Yin, Wei Li, Chengzhong Xu, Ruigang Yang, and Jianbing Shen. Di-v2x: Learning domain-invariant representation for vehicle-infrastructure collaborative 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3208–3215, 2024. 1, 2, 7
- [17] 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. 1, 7
- [18] Yiming Li, Dekun Ma, Ziyang 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. 1, 2, 7
- [19] S Liu, C Gao, Y Chen, X Peng, X Kong, K Wang, R Xu, W Jiang, H Xiang, J Ma, et al. Towards vehicle-to-everything autonomous driving: A survey on collaborative perception. *arXiv 2023. arXiv preprint arXiv:2308.16714*. 1
- [20] 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*, pages 4106–4115, 2020. 1
- [21] 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)*, pages 6876–6883. IEEE, 2020. 1
- [22] Yifan Lu, Yue Hu, Yiqi Zhong, Dequan Wang, Yanfeng Wang, and Siheng Chen. An extensible framework for open heterogeneous collaborative perception. *arXiv preprint arXiv:2401.13964*, 2024. 2, 7
- [23] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 4

- [24] 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*, pages 1186–1195, 2023. 7
- [25] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: Real-time and high-performance pillar-based 3d object detection. In *European Conference on Computer Vision*, pages 35–52. Springer, 2022. 6
- [26] Tianhang Wang, Guang Chen, Kai Chen, Zhengfa Liu, Bo Zhang, Alois Knoll, and Changjun Jiang. Umc: A unified bandwidth-efficient and multi-resolution based collaborative perception framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8187–8196, 2023. 1
- [27] 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*, pages 605–621. Springer, 2020. 1, 2
- [28] Sizhe Wei, Yuxi Wei, Yue Hu, Yifan Lu, Yiqi Zhong, Siheng Chen, and Ya Zhang. Asynchrony-robust collaborative perception via bird’s eye view flow. *Advances in Neural Information Processing Systems*, 36:28462–28477, 2023. 1, 2, 7
- [29] Hao Xiang, Runsheng Xu, and Jiaqi Ma. Hm-vit: Heteromodal vehicle-to-vehicle cooperative perception with vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 284–295, 2023. 2, 7
- [30] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. Cobevt: Cooperative bird’s eye view semantic segmentation with sparse transformers. *arXiv preprint arXiv:2207.02202*, 2022. 7
- [31] 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*, pages 107–124. Springer, 2022. 1, 2, 7
- [32] 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)*, pages 2583–2589. IEEE, 2022. 1, 7
- [33] 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)*, pages 6035–6042. IEEE, 2023. 2
- [34] 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*, pages 13712–13722, 2023. 1, 2, 3
- [35] Dingkan 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*, 36:25151–25164, 2023. 1
- [36] Lei Yang, Tao Tang, Jun Li, Peng Chen, Kun Yuan, Li Wang, Yi Huang, Xinyu Zhang, and Kaicheng Yu. Bevheight++: Toward robust visual centric 3d object detection. *arXiv preprint arXiv:2309.16179*, 2023. 2
- [37] Lei Yang, Kaicheng Yu, Tao Tang, Jun Li, Kun Yuan, Li Wang, Xinyu Zhang, and Peng Chen. Bevheight: A robust framework for vision-based roadside 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21611–21620, 2023. 2
- [38] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, et al. 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*, pages 21361–21370, 2022. 1, 2, 7
- [39] Haibao Yu, Yingjuan Tang, Enze Xie, Jilei Mao, Ping Luo, and Zaiqing Nie. Flow-based feature fusion for vehicle-infrastructure cooperative 3d object detection. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 7
- [40] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 6
- [41] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 408–417, 2017. 5
- [42] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2349–2358, 2017. 5