IPDA:Intelligent Perception Delay Alignment Method Based on Spatio-Temporal Co-Sensing Calibration

Jianhang Liu[1,2*], Dianzheng Zhang[1,2], Hongxin Pan[1,2], Guangqian Jiang[1,2], Runda Fan[1,2]

[1]Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, China

[2]Shandong Key Laboratory of Intelligent Oil & Gas Industrial Software, Qingdao, China

liujianhang@upc.edu.cn, {z23070008,s23070001,z23070075,z23070060}@s.upc.edu.cn

Abstract

Intelligent perception among multiple agents enables them to extend their individual observation capabilities by sharing sensory information, thereby improving the completeness and accuracy of environmental understanding. However, real-world communication is often subject to non-negligible delays, which can degrade the effectiveness of perception. To mitigate this, delay alignment is commonly employed to synchronize delayed observations to a common timestamp. Yet, both alignment errors and inherent discrepancies between multi-view observations can lead to inconsistencies in the estimated position and orientation of shared targets. These inconsistencies can accumulate during feature fusion, ultimately reducing the accuracy and reliability of the perception results.To address this challenge, we propose IPDA, a delay-aware multi-agent intelligent perception method that performs joint calibration in both temporal and spatial domains. In the temporal dimension, we design a historical alignment attention mechanism to model dynamic delay correction across sequences, ensuring temporal coherence. In the spatial dimension, we introduce a discrepancy-quantized co-sensing network that captures and compensates for multi-view spatial deviations caused by viewpoint diversity and alignment inaccuracy. IPDA is evaluated on two large-scale intelligent perception benchmarks, DAIR-V2X and OPV2V. Experimental results demonstrate that our method effectively mitigates delay-induced inconsistencies and consistently outperforms state-of-the-art baselines under various delay conditions.

1 Introduction

Intelligent perception is a key technique in multi-agent perception systems, enabling agents to exchange information and jointly construct a more complete representation of their environment. This approach effectively addresses the limitations of single-agent perception, such as restricted field of view and limited long-range detection capability (Zhang et al. 2024a)(Xu et al. 2025). By integrating heterogeneous sensor observations from multiple agents, intelligent perception significantly enhances spatial coverage and representation accuracy, providing richer and more consistent environmental context in complex scenarios. Prior studies show that multi-agent sensor fusion can improve detection range by up to 2.5 times in tasks such as occluded object localization and cooperative segmentation (Yang et al. 2023; Lu et al. 2024; Yoo et al. 2025), and reduce blind-zone miss rates by 67% (Wang et al. 2025)(Ding et al. 2025), thereby enhancing the reliability of perception in distributed systems.

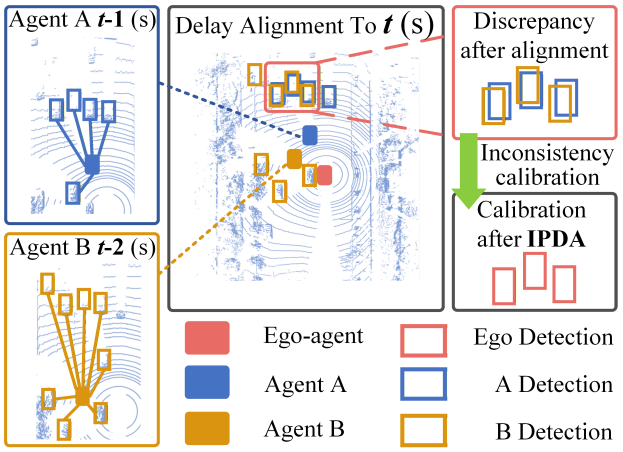

Figure 1: Based on the intelligent mechanism of Spatio-Temporal Co-Sensing Calibration, IPDA effectively reduces data inconsistency and significantly improves the accuracy of data fusion.

Despite these advances, intelligent perception systems face significant challenges such as bandwidth limitations, constrained computation, and time synchronization issues (Zhang et al. 2024b; Yu et al. 2023; Hong et al. 2024; Chiu et al. 2024). These practical constraints induce communication delays, which in real-world deployments can accumulate to 498 communication cycles plus over 130 milliseconds of additional latency in LTE-V2X systems (Lee et al. 2017), severely impacting the consistency and effectiveness of shared representations across agents. Existing solutions attempt to address such delay-induced misalignments. Some methods treat delay time as an input feature (Xu et al. 2022a), but are limited to single-frame alignment and cannot compensate for displacements under large motion or long delays. Other works exploit historical features

*Corresponding author: Jianhang Liu

to predict the current representation (Lei et al. 2022), but rely on fixed interval assumptions that perform poorly under variable-delay conditions. Although recent approaches begin to address irregular delay (Wei et al. 2023), a fundamental problem remains: forcing observations with heterogeneous delays to align at a unified timestamp introduces compounded errors and multi-view inconsistencies, leading to significant deviations in the perception of the same object by different agents (as illustrated in Figure 1).In severe cases, large inconsistencies may cause multiple agents to generate independent predictions for the same physical entity, resulting in false positives and fragmented detection, which compromises the stability of the intelligent perception system.

From a temporal perspective, delay alignment typically leverages historical observations for sequential correction. Due to overlap in successive frames, these alignments exhibit inherent continuity, enabling correlation modeling to reduce temporal inconsistency. From a spatial perspective, despite variation in agent positions and viewpoints, observations of shared targets remain partially correlated. This spatial redundancy can be explicitly modeled to resolve geometric inconsistencies introduced by viewpoint variation and delay misalignment.Based on these insights, this paper introduces IPDA, a delay-aware intelligent perception method based on Spatio-Temporal Co-Sensing Calibration, designed to reduce representation inconsistency in asynchronous multi-agent systems.

The main contributions of this paper are as follows:

- We propose a history-guided perception alignment model for temporal calibration. A historical alignment attention mechanism is introduced to capture dynamic correlations across past alignment results, thereby improving the stability and accuracy of temporal delay correction under varying latency.
- We propose a spatial co-sensing calibration method with discrepancy quantization. A delay discrepancy-aware Co-Sensing Calibration Network is constructed to model multi-agent viewpoint differences under delay, enabling accurate spatial calibration via graph-based optimization.
- Comprehensive experiments on the DAIR-V2X and OPV2V benchmarks validate that IPDA effectively mitigates delay-induced inconsistency, achieving consistent improvements over state-of-the-art methods across diverse delay scenarios.

2 Related Work

Latency Issues in Intelligent Perception. Intelligent perception has advanced with large-scale datasets like V2X-Sim (Li et al. 2022), OPV2V (Xu et al. 2022b), and DAIR-V2X (Yu et al. 2022), yet communication delays and observation asynchrony still cause spatio-temporal misalignment and impair fusion. Methods like V2VNet (Wang et al. 2020) and V2X-ViT (Xu et al. 2022a) leverage temporal cues or delay-aware encodings, but rely on single-frame inputs and ignore long-term context. SyncNet (Lei et al. 2022) uses ConvLSTM for feature propagation, but its fixed-step assumption limits flexibility. CoBEVFlow (Wei et al. 2023) handles irregular delays, yet neglects residual inconsistencies post-alignment. These limitations accumulate in fusion, degrading perception. To address this, we propose a spatio-temporal calibration framework with history-aware alignment attention and delay discrepancy quantification for robust consistency under asynchrony.

Simultaneous Localization and Mapping (SLAM). SLAM (Yuan et al. 2024; Keetha et al. 2024; Huang et al. 2024) estimates an agent's trajectory while mapping the environment, typically via graph-based optimization (Grisetti et al. 2010; Lu and Milios 1997; Levenberg 1944). Inspired by pose consistency modeling in multi-agent SLAM (Zhang et al. 2022; Hu et al. 2023), we adapt it to perception tasks to address delay-induced spatiotemporal misalignment.

3 Problem Formulation

Consider N agents in the scene, each capable of exchanging feature data derived from sensor observations via inter-agent communication. Upon receiving a message, the agent checks its timestamp to determine delay; if delayed, a delay alignment module estimates the current state. Finally, each agent fuses aligned data and outputs target detection results.

$$F_n^i = f_{\text{encoder}}(O_n^i), \tag{1a}$$

$$F_m^i = f_{\text{alignment}}(F_m^j), \tag{1b}$$

$$H_n^i = f_{\text{fusion}}(F_n^i, \{F_m^i\}_{m\in\mathcal{N}_n}), \tag{1c}$$

$$Y_n^i = f_{\text{decoder}}(H_n^i), \tag{1d}$$

where F_n^i is the feature of the nth agent at the ith timestamp, O_n^i is the original point cloud data of the nth agent at the ith timestamp, F_m^i is the feature of the mth agent at the ith timestamp after delay alignment, H_n^i is the aggregated feature at the ith timestamp of the nth agent after fusing information from other agents, $\mathcal{N}_n$ is the collaborating neighbors of the nth agent, and Y_n^i is the final output of the nth agent at the ith timestamp.Although Step (1b) aligns cross-agent data to a unified timestamp, delay variations, alignment noise, and multi-view discrepancies still cause pose inconsistencies in overlapping regions. These errors accumulate during fusion, severely degrading intelligent perception. The next section introduces how IPDA addresses this issue.

4 Spatio-Temporal Co-Sensing Calibration

4.1 Overall Architecture

Now we propose IPDA, a novel delay alignment method that addresses data inconsistency in delayed scenarios through a time-space cascade calibration architecture. For agent i, IPDA proceeds as follows:

$$F_n^i, B_n^i = f_{\text{detection}}(O_n^i), \tag{2a}$$

$$\hat{F}_m^i, \hat{B}_m^i, \hat{\phi}_m^i = f_{\text{temporal}}(F_m^j, B_m^j, \phi_m^j, [\bar{B}_m^{i-T\sim i}, E_m]), \tag{2b}$$

$$\{\tilde{\phi}_m^i\}_{\tilde{\phi}_m^i\in\varepsilon_n^i} = f_{\text{spatial}}(\{\hat{B}_m^i, \hat{\phi}_m^i\}_{m\in\mathcal{N}_n}), \tag{2c}$$

$$\tilde{F}_m^i = f_{\text{transform}}(\hat{F}_m^i, \tilde{\phi}_m^i), \tag{2d}$$

$$H_n^i = f_{\text{fusion}}(\{\tilde{F}_m^i\}_{m\in\mathcal{N}_n}), \tag{2e}$$

$$Y_n^i = f_{\text{decoder}}(H_n^i), \tag{2f}$$

where B_n^i is the set of bounding boxes output by agent n at timestamp i, i.e., $B_n^i = \{b_{nk}^i\}_{k=1}^{K_n^i}$, where each b_{nk}^i represents a single bounding box corresponding to object k observed by agent n at time i, and K_n^i denotes the total number of detected objects by agent n at time i. In this paper, each bounding box b_{nk}^i is represented as $b_{nk}^i = (x, y, \theta, \lambda_x, \lambda_y, \lambda_\theta, \omega, fixed)$, where (x, y, θ) denote the center position and yaw angle of the object, $(\lambda_x, \lambda_y, \lambda_\theta)$ are the discrepancy factors in each dimension used during optimization, ω is a delay weight parameter indicating the alignment confidence, and $fixed$ is a binary flag indicating whether the bounding box pose is fixed during optimization. Similarly, $b_{mk}^i \in B_m^i$ is the bounding box generated by agent m for object k at time i; $\hat{F}_m^i$, $\hat{B}_m^i$, and $\hat{\phi}_m^i$ represent the feature, bounding boxes output, and agent pose data of agent m at timestamp i after temporal co-sensing calibration, respectively. ϕ_m^j is the agent pose data of agent m at timestamp j. $\bar{B}_m^{i-T\sim i}$ is the historical alignment bounding boxes output of agent m stored in agent n over a time span T; E_m is the historical association encoding among the historical alignment bounding boxes; $\tilde{\phi}_m^i$ is the pose data of agent m at timestamp i after spatial co-sensing calibration; ε_n^i is the set of optimizable pose data of agent n at timestamp i, during the spatial co-sensing calibration. $\tilde{F}_m^i$ is the feature of agent m at timestamp i after spatial co-sensing calibration.

Following (Lu et al. 2023), Step (2a) performs object detection and outputs structured bounding boxes for later calibration. As it largely follows (Lu et al. 2023) with minor changes, internal details are omitted. Step (2b) temporally calibrates delayed inputs using historical data and alignment boxes $\bar{B}_m^{i-T\sim i}$ with encoding E_m, enhancing temporal consistency of features and boxes. Step (2c) spatially calibrates $\hat{\phi}_m^i$ and $\hat{B}_m^i$ across agents by incorporating delay-aware discrepancies to correct pose deviation. Step (2d) transforms features via corrected poses for spatial alignment. Step (2e) applies multi-scale maximum fusion across agents; its implementation is omitted. Finally, Step (2f) decodes the fused features into final detection outputs. See Figure 2 for the pipeline.

4.2 Temporal Co-Sensing Calibration

The temporal co-sensing calibration module primarily aims to align delay-affected data exchanged between agents along the temporal dimension and improve temporal consistency. It also prepares the aligned data and relevant auxiliary information for the subsequent spatial co-sensing calibration module. The core steps of this module include historical perception alignment and motion state judgement.

Historical Perception Alignment. Historical perception alignment first uses stored historical data to perform initial temporal alignment of delayed information. Specifically, the historical bounding box sequence $\{B_m^q\}_{q=j-k+1,j-k+2,\ldots,j}$, obtained from agent observations, is aligned using the method in (Wei et al. 2023) to produce $\bar{B}_m^i$. To improve alignment accuracy and temporal consistency, we introduce a historical alignment attention mechanism that integrates current and past alignment information and captures dynamic correlations across time via an attention-based structure. Each initially aligned box conducts self-attention over historical prediction embeddings within time span T.

$$\hat{B}_m^i = MHA(\bar{B}_m^i, [\bar{B}_m^{i-T\sim i}, E_m], [\bar{B}_m^{i-T\sim i}, E_m]), \tag{3}$$

$$E_m = MLP(\upsilon, \chi, \eta), \tag{4}$$

where $\bar{B}_m^i$ is the pre-aligned bounding boxes at timestamp i of agent m. $MHA(\cdot)$ is the multi-head attention for historical perception alignment. $E_m \in \mathbb{R}^{Z\times D}$ is the historical association encoding between the history aligned bounding boxes, where Z is the number of edges between bounding boxes and D is the dimensionality of each edge. Specifically, υ denotes the spatial distance, χ the relative orientation, and η the temporal gap between the historical aligned bounding boxes. Subsequently, using the temporally calibrated bounding boxes, the motion vector for each bounding box b is calculated via motion vector transformation to generate the motion intention vector $V_m^{j\to i}$, which describes the target's motion state at the current time. Based on this vector, the module further performs delay alignment on the received delayed feature data, aligning it to the current moment.

$$V_m^{j\to i} = M_m^{j\to i} \cdot (\hat{B}_m^i - B_m^j), \tag{5}$$

$$\hat{F}_m^i(p) = F_m^j(p + V_m^{j\to i}(p)), \tag{6}$$

where $M_m^{j\to i}$ is the affine transformation matrix generated based on the orientation and position of the bounding boxes from timestamp j to timestamp i for agent m. $V_m^{j\to i} \in \mathbb{R}^{H\times W\times 2}$ is the motion intention vector of agent m'bounding boxes from timestamp j to timestamp i, where H and W are the spatial dimensions and 2 indicates the number of channels. $p = [h, w]$ is a specific position in the feature map, and $V_m^{j\to i}(p)$ is the motion intention vector at position p from timestamp j to timestamp i. $\hat{F}_m^i$ is the temporally aligned feature data of agent m at timestamp i.

Since the subsequent spatial co-sensing calibration module requires the real-time pose data of each agent, we adopt the same alignment strategy used for bounding boxes to obtain the aligned pose data $\hat{\phi}_m^i$ of agent m at timestamp i. Meanwhile, the module dynamically updates the delay weight attributes ω of all bounding boxes b based on their corresponding delay information. Specifically, boxes with lower latency are assigned higher weights, while those with higher latency receive lower weights, thereby prioritizing more reliable data in the subsequent pose correction process:

$$\omega = \frac{1}{\Delta t(1 - e^{-\mu\Delta t}) + 1}, \tag{7}$$

where μ is the delay adjustment factor that controls the rate at which the weight decays with increasing delay. Δt denotes the time delay between the generation of the bounding boxes and their processing.

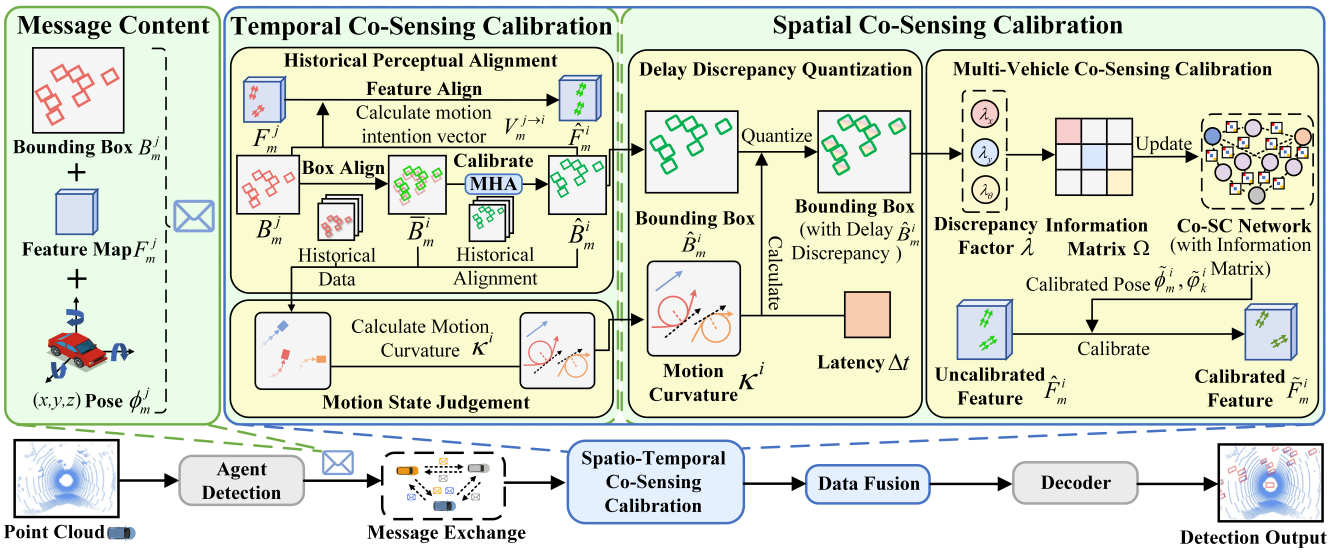

Figure 2: Overview of the proposed co-sensing calibration method.

Motion State Judgement. The motion state judgment module calculates the path curvature of each bounding box b by combining the agent's historical data with delay-aligned data to infer its motion status. This information is then used to adjust the discrepancy factor λ and the information matrix Ω for each bounding box b in the "Spatial Co-Sensing Calibration" module. Path curvature is a key geometric metric that characterizes the degree of bending in the agent's trajectory, thereby reflecting its turning behavior. Specifically, the curvature is computed by analyzing the positional variations of the same bounding box b_{mk} across the historical bounding boxes $\{B_m^q\}_{q=j-k+1,\ldots,j}$ and the delay-aligned bounding box $\hat{B}_m^i$.

For a planar path, the curvature ${\kappa^i}_{mk}$ of bounding box ${b^i}_{mk}$ is defined as the rate of change in the tangent direction. The specific calculation formula is as follows:

$$ {\kappa^i}_{mk} = \frac{4S}{a \cdot b \cdot c}, \tag{8} $$

$$ S = \frac{1}{2} \left| \det \begin{bmatrix} x^j & y^j & 1 \\ x^i & y^i & 1 \\ x^{i'} & y^{i'} & 1 \end{bmatrix} \right|, \tag{9} $$

$$ a = \sqrt{(x^i - x^j)^2 + (y^i - y^j)^2}, \tag{10} $$

$$ b = \sqrt{(x^{i'} - x^i)^2 + (y^{i'} - y^i)^2}, \tag{11} $$

$$ c = \sqrt{(x^{i'} - x^j)^2 + (y^{i'} - y^j)^2}, \tag{12} $$

where (x^i, y^i) is the position of the agent extracted from $\bar{B}_m^i$ at timestamp i, while (x^j, y^j) and $(x^{i'}, y^{i'})$ are the positions obtained from B_m^j and $\hat{B}_m^i$, respectively.

By computing path curvature, the system quantifies trajectory bending to assess motion behavior. Curvature stays low during straight-line motion but rises when turning, signaling a motion state shift. Real-time curvature changes, compared to a threshold, help infer whether the agent is turning or moving straight. During multi-agent calibration, historical curvature guides motion state classification, which adaptively adjusts the discrepancy factor λ and information matrix Ω to enhance alignment robustness under delay.

4.3 Spatial Co-Sensing Calibration

While temporal calibration aligns delayed data, it cannot fully resolve spatial inconsistencies, which may be amplified during fusion and degrade accuracy. To address this, spatial calibration is applied post-temporal alignment, leveraging overlapping views to correct misalignments via delay discrepancy quantification and multi-agent co-sensing calibration.

Delay Discrepancy Quantification. This module quantifies the spatial inconsistency of each bounding box b, mainly reflected in the deviation of (x, y, θ). In delayed scenarios, different motion states cause delay to affect (x, y, θ) differently. To address this, we utilize the motion curvature computed in the previous motion state judgement module to better capture delay-induced spatial misalignment. As spatial calibration is independent of time, we omit time-related superscripts in the following symbols for clarity.

Specifically, delay has distinct impacts on (x, y) and θ. When an agent moves straight, positional errors dominate—accumulating with delay time and speed—while yaw deviations remain small. In turning, high angular velocity makes yaw angle errors more significant, which may further amplify position errors nonlinearly. Thus, distinguishing how delay and motion state affect position and orientation is critical. To address this, we define a delay-motion influence function that models how delay Δt and curvature κ impact (x, y, θ) inconsistencies, and compute separate discrepancy factors λ_x, λ_y, and λ_θ for each bounding box b. For

example, the factors of b_{mk} are computed as follows:

$$\lambda_x(\kappa_{mk}, \Delta t) = \lambda_{\text{base}} \cdot \Delta t \cdot \exp(\alpha \cdot (\kappa_{mk} - \epsilon_\kappa)), \quad (13)$$

$$\lambda_y(\kappa_{mk}, \Delta t) = \lambda_{\text{base}} \cdot \Delta t \cdot \exp(\alpha \cdot (\kappa_{mk} - \epsilon_\kappa)), \quad (14)$$

$$\lambda_\theta(\kappa_{mk}, \Delta t) = \lambda_{\text{base}} \cdot \Delta t \cdot \exp(\beta \cdot (\kappa_{mk} - \epsilon_\kappa)), \quad (15)$$

where λ_{base} is the base discrepancy factor. α and β ($0 < \alpha < \beta$) are coefficients controlling how discrepancy factors grow with curvature for position and yaw, respectively. κ_{mk} is the motion curvature from the temporal co-sensing calibration module. ϵ_κ is the curvature threshold: if $\kappa_{mk} > \epsilon_\kappa$, the agent is turning; otherwise, it is moving straight. $\exp(\cdot)$ is the natural exponential function.

Finally, the computed discrepancy factors are embedded into each bounding box as key attributes indicating delay-induced spatial reliability. This allows explicit representation of spatio-temporal distortion per observation. This process enables fine-grained quantification of asynchronous spatial inconsistencies, which provides a solid basis for confidence modulation in the subsequent multi-agent co-sensing calibration, thereby enhancing fusion consistency and reliability.

Multi-Agent Co-Sensing Calibration. In this module, we construct the Co-Sensing Calibration Network and leverage the discrepancy factors λ computed in the previous module to adjust the information matrix Ω, thereby enabling spatial inconsistency calibration for the temporally aligned data. This process consists of two key steps: Co-Sensing Calibration Network construction and spatial inconsistency calibration.

Co-Sensing Calibration Network construction establishes the structural basis and information flow for calibration. Specifically, the ego-agent receives messages from other detection units and constructs a Co-Sensing Calibration Network $G\left(V_{\text{detector}}, V_{\text{target}}, \psi\right)$ to represent the association between detection and target units. Each node corresponds to a pose. The node set V_{detector} includes the ego-agent and m detection units transmitting data. The pose ϕ of each unit, temporally calibrated via the temporal co-sensing module, includes attributes $(x, y, \theta, fixed)$ for position, yaw, and optimization status. The node set V_{target} includes detected target units. As a target may be observed by multiple units, it may have several bounding boxes. To unify these into a consistent pose φ, we aggregate the corresponding boxes into a box set. The resulting φ, like ϕ, has attributes $(x, y, \theta, fixed)$. The edge set ψ encodes detection relationships. When a detection unit observes a target and outputs a bounding box, an edge γ_{mk} is added, indicating the relative pose of the k-th target unit as seen by the m-th detection unit, directly inferred from the detection result.

Delay-weighted spatial aggregation addresses how to incorporate delay characteristics of multiple bounding boxes when estimating a target unit's unique pose. To this end, we propose a spatial aggregation method that introduces delay-based weighting to enhance aggregation accuracy. A threshold $\omega_{\text{threshold}}$ is set to differentiate delayed bounding boxes. For boxes with $\omega > \omega_{\text{threshold}}$, they are marked as $fixed = True$ and directly adopted as part of the aggregation result, acting as hard constraint nodes during aggregation and graph optimization. For boxes with $\omega \leq \omega_{\text{threshold}}$, we perform weighted aggregation based on delay weight ω, assigning higher weights to less delayed boxes to improve final pose reliability. The $fixed$ status of detection unit ϕ is likewise determined by the average delay weight of its associated bounding boxes.

Aggregation was done using the weighted mean method:

$$\varphi_k = \frac{\Sigma_{b \in A_k}(\omega(b) \cdot b)}{\Sigma_{b \in A_k}\omega(b)}, \quad (16)$$

where φ_k is the pose information of target unit k, which is obtained through spatial aggregation. A_k is the set of all bounding boxes corresponding to the same target unit k, as detected by multiple detection units. $b = (x, y, \theta, \lambda_x, \lambda_y, \lambda_\theta, \omega, fixed)$ is the complete state information of a bounding box corresponding to target unit k, where $\omega(b)$ is its delay weight.

The goal of multi-agent co-sensing calibration is to correct spatial inconsistencies across agents. Prior to this, it is essential to compute the information matrix Ω for each edge in the Co-Sensing Calibration Network. Similar to graph-based SLAM (Grisetti et al. 2010)(Lu and Milios 1997), Ω quantifies the confidence in each detection unit's observation and guides pose adjustment. We model delay as the primary factor affecting confidence, and introduce a dynamic alignment term based on azimuth change $\Delta\theta$ to improve adaptability under complex motion. Since delay impacts position and orientation confidence differently, we use the discrepancy factor λ from each bounding box b (obtained in delay discrepancy quantification) to compute confidence prior to spatial aggregation. This confidence is then used to construct the corresponding Ω for each target unit. A functional relationship is defined between confidence and λ to guide the construction of Ω as follows:

$$\tau_x(\Delta t) = \frac{1}{1 + \lambda_x \cdot \Delta t}[\rho + \zeta \cdot \cos(\Delta\theta)], \quad (17)$$

$$\tau_y(\Delta t) = \frac{1}{1 + \lambda_y \cdot \Delta t}[\rho + \zeta \cdot \cos(\Delta\theta)], \quad (18)$$

$$\tau_\theta(\Delta t) = \frac{1}{1 + \lambda_\theta \cdot \Delta t}[\rho + \zeta \cdot \cos(\Delta\theta)], \quad (19)$$

$$\Omega_{mk} = diag([\tau_x, \tau_y, \tau_\theta]) \cdot \Omega_0 \in \mathbb{R}^{3\times3}, \quad (20)$$

where ρ and ζ are weighting coefficients. Ω_{mk} denotes the information matrix of detection unit m observing target unit k, and Ω_0 is the base matrix representing confidence under zero delay. Through this process, Ω is dynamically adjusted based on motion state and delay magnitude. By combining the relationship between discrepancy factor λ and confidence τ, position and yaw confidences can be adaptively modulated: when motion curvature $\kappa = \epsilon_\kappa$, we have $\lambda_x(\kappa) = \lambda_y(\kappa) = \lambda_\theta(\kappa) = \lambda_{\text{base}}$, thus $\tau_x(\Delta t) = \tau_y(\Delta t) = \tau_\theta(\Delta t)$. When $\kappa < \epsilon_\kappa$, due to $\alpha < \beta$, the yaw discrepancy decreases faster, i.e., $\lambda_x(\kappa) = \lambda_y(\kappa) > \lambda_\theta(\kappa)$, yielding $\tau_x(\Delta t) = \tau_y(\Delta t) < \tau_\theta(\Delta t)$; whereas when $\kappa > \epsilon_\kappa$, the yaw discrepancy increases faster, i.e., $\lambda_x(\kappa) = \lambda_y(\kappa) < \lambda_\theta(\kappa)$, leading to $\tau_x(\Delta t) = \tau_y(\Delta t) > \tau_\theta(\Delta t)$.

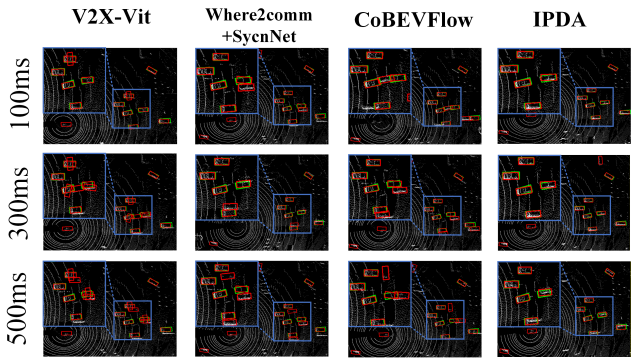

Figure 3: Detection results of V2X-ViT, CoBEVFlow, Where2comm+SyncNet and IPDA under 100ms, 300ms, and 500ms delays on DAIR-V2X. Green: ground truth; Red: detections.

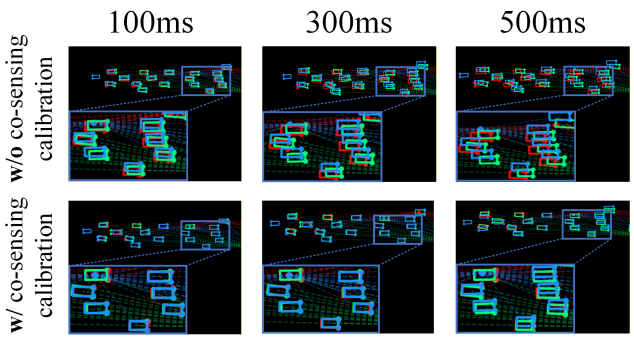

Figure 4: Effectiveness of Spatio-Temporal Co-Sensing Calibration in improving the accuracy of bounding box positioning under different delay conditions on the OPV2V dataset.

After computing the information matrix Ω, the system performs spatial inconsistency calibration. Ideally, target unit poses should be consistent across detection units in the Co-Sensing Calibration Network, meaning the posture consistency error vector $e_{mk} = \gamma_{mk}^{-1} \circ (\phi_m^{-1} \circ \varphi_k) \in \mathbb{R}^3$ should be zero. Here, $\circ$ denotes motion composition, equivalent to multiplying homogeneous transformation matrices, and $^{-1}$ denotes the inverse pose, corresponding to matrix inversion.

To promote the consistency of this posture, this paper considers the following least squares optimization problem:

$$\{\tilde{\phi}_m, \tilde{\varphi}_k\} = \underset{\{\phi_m, \varphi_k\}}{\arg\min} E(\varepsilon) = \sum_{(\phi_m, \varphi_k) \in \varepsilon_n} e_{mk}^T \Omega_{mk} e_{mk} \,. \quad (21)$$

The optimization adopts a graph-based SLAM paradigm and is solved using standard methods like Gauss-Newton or Levenberg-Marquardt (Levenberg 1944). Unlike traditional approaches that update all nodes uniformly, we selectively optimize only pose nodes marked as optimizable during delay-weighted spatial aggregation, treating fixed nodes as hard constraints. This anchors non-delayed nodes to maintain spatial consistency across agents. Spatial calibration is then performed using the corrected poses $\tilde{\phi}_m$ and $\tilde{\varphi}_k$, and the calibrated features are fused and decoded for final detection.

5 Experiments

5.1 Experiments Setup

Dataset. DAIR-V2X (Yu et al. 2022) is a real-world vehicle-road cooperative sensing dataset featuring multi-modal data from vehicle-mounted and roadside sensors (e.g., LiDAR, cameras). Each frame includes synchronized data at 10 Hz and 3D annotations for both the ego vehicle and roadside unit. To enhance coverage, Lu et al. (Lu et al. 2023) supplemented missing annotations beyond the vehicle-side view, enabling 360° detection. We adopt these augmented labels and set the sensing range to $x \in [-100.8, \mathrm{m}, 100.8, \mathrm{m}]$ and $y \in [-40, \mathrm{m}, 40, \mathrm{m}]$ to ensure broad-area perception.

OPV2V (Xu et al. 2022b), built on CARLA (Dosovitskiy et al. 2017) and OpenCDA (Xu et al. 2021), includes 73 scenarios across six road types and nine urban settings. Following the configuration in (Xu et al. 2022b), we set the detection range to $x \in [-140, \mathrm{m}, 140, \mathrm{m}]$ and $y \in [-40, \mathrm{m}, 40, \mathrm{m}]$.

Implementation Details and Loss Function. In the experiment, we adopt a PointPillars-based encoder (Lang et al. 2019) for point cloud feature extraction and use multi-scale max fusion for integrating multi-source features. To mitigate delay-induced inconsistency, we leverage the g2o library (Kümmerle et al. 2011) and apply the Levenberg-Marquardt algorithm (Levenberg 1944) for cross-agent consistency optimization. Training settings include: delay weight $\omega_{\text{base}} = 1.0$, temporal factor $\mu = 0.01$, spatial coefficients $\alpha = 0.125$, $\beta = 0.25$, and base discrepancy $\lambda_{\text{base}} = 0.025$. The curvature threshold $\epsilon_\kappa = 0.02$ is used to distinguish motion types. The initial information matrix Ω_0 is set as identity for confidence computation in the calibration module.We adopt standard detection losses, enhanced with alignment and consistency terms. MSE is used for temporal alignment supervision, while KL Divergence measures distributional consistency across agents.

5.2 Quantitative Results

Benchmark comparison. To evaluate the 3D detection performance of IPDA under latency-induced inconsistencies, we conducted experiments on the DAIR-V2X (Yu et al. 2022) and OPV2V (Xu et al. 2022b) datasets, comparing it with existing methods. As shown in Table 1, IPDA consistently achieves the best AP at both 0.5 and 0.7 IoU thresholds across all latency levels. On DAIR-V2X, IPDA shows modest gains under no-delay conditions (e.g., +2.85% at IoU 0.5), but exhibits strong robustness as latency increases, maintaining AP above 0.77 from 100ms to 500ms. At IoU 0.7, it achieves an average improvement of 15% and peaks at +16.31% at 200ms. On OPV2V, IPDA also leads all baselines, with average gains of 6.63% and 8.32% at IoU thresholds of 0.5 and 0.7, respectively, validating its effectiveness

Dataset	DAIR-V2X						OPV2V					
Model/Metric	AP@0.5↑											
Latency Time(ms)	0	100	200	300	400	500	0	100	200	300	400	500
Single	0.674						0.697					
Late Fusion	0.709	0.664	0.616	0.600	0.595	0.593	0.779	0.759	0.743	0.726	0.719	0.704
V2VNet(Wang et al. 2020)	0.626	0.623	0.622	0.617	0.612	0.608	0.801	0.773	0.751	0.746	0.739	0.730
V2X-ViT(Xu et al. 2022a)	0.725	0.714	0.704	0.693	0.681	0.681	0.822	0.790	0.777	0.763	0.759	0.753
DiscoNet(Mehr et al. 2019)	0.645	0.630	0.620	0.606	0.597	0.590	0.753	0.732	0.716	0.704	0.685	0.622
Where2comm(Hu et al. 2022)+SyncNet(Lei et al. 2022)	0.807	0.719	0.706	0.695	0.687	0.679	0.849	0.817	0.803	0.784	0.779	0.768
CoBEVFlow(Wei et al. 2023)	0.807	0.765	0.749	0.738	0.728	0.726	0.831	0.819	0.810	0.805	0.797	0.787
IPDA(Ours)	**0.830**	**0.820**	**0.809**	**0.794**	**0.790**	**0.778**	**0.876**	**0.866**	**0.861**	**0.859**	**0.857**	**0.851**
Model/Metric	AP@0.7↑											
Latency Time(ms)	0	100	200	300	400	500	0	100	200	300	400	500
Single	0.587						0.627					
Late Fusion	0.538	0.479	0.467	0.464	0.466	0.465	0.658	0.627	0.590	0.576	0.565	0.564
V2VNet(Wang et al. 2020)	0.556	0.555	0.554	0.552	0.548	0.548	0.693	0.650	0.634	0.622	0.615	0.603
V2X-ViT(Xu et al. 2022a)	0.556	0.553	0.550	0.545	0.541	0.541	0.732	0.701	0.683	0.677	0.669	0.660
DiscoNet(Mehr et al. 2019)	0.553	0.537	0.530	0.523	0.519	0.518	0.667	0.622	0.603	0.589	0.575	0.569
Where2comm(Hu et al. 2022)+SyncNet(Lei et al. 2022)	0.662	0.602	0.588	0.587	0.584	0.580	0.765	0.727	0.705	0.693	0.685	0.679
CoBEVFlow(Wei et al. 2023)	0.662	0.621	0.601	0.599	0.592	0.588	0.761	0.748	0.729	0.725	0.719	0.714
IPDA(Ours)	**0.743**	**0.714**	**0.699**	**0.688**	**0.683**	**0.678**	**0.830**	**0.804**	**0.792**	**0.785**	**0.780**	**0.771**

Table 1: Performance Comparison of IPDA and Baseline Methods at Expected Delays from 0ms to 500ms on DAIR-V2X(Yu et al. 2022) and OPV2V(Xu et al. 2022b) datasets.

Modules				AP@0.5/AP@0.7↑		
Historical alignment attention mechanism	Motion State Judgement	Delay Discrepancy Quantification	Spatial Aggregation	100ms	300ms	500ms
	Fixed	Fixed	IoU	0.689/0.543	0.663/0.524	0.626/0.496
✓	Fixed	Fixed	IoU	0.715/0.605	0.684/0.596	0.663/0.581
✓	Ours	Fixed	IoU	0.742/0.636	0.721/0.623	0.706/0.605
✓	Ours	Ours	IoU	0.802/0.648	0.796/0.634	0.747/0.618
✓	Ours	Ours	Ours	0.820/0.714	0.794/0.688	0.778/0.678

Table 2: Ablation Study Results on DAIR-V2X(Yu et al. 2022).

in mitigating delay-aligned inconsistencies and enhancing robustness.

5.3 Qualitative Results

Visualization of detection results. Figure 3 shows detection results from V2X-ViT (Xu et al. 2022a), Where2comm (Hu et al. 2022) + SyncNet (Lei et al. 2022), CoBEVFlow (Wei et al. 2023), and the proposed IPDA under 100ms, 300ms, and 500ms delays on DAIR-V2X (Yu et al. 2022). Red and green boxes denote predictions and ground truth, respectively. While existing methods offer some delay tolerance, they cannot resolve spatio-temporal inconsistencies from delayed misalignment, often resulting in box-splitting—where the same object is mistakenly detected as multiple targets. IPDA mitigates such inconsistency, enhances alignment across agents, and produces predictions with higher overlap, demonstrating superior performance under delay.

Consistency calibration visualization. Figure 4 shows the effect of the proposed Spatio-Temporal Co-Sensing Calibration on bounding box alignment under different delays in OPV2V (Xu et al. 2022b). Colored boxes indicate detections from different agents. With increasing delay, spatial misalignments become evident, especially in overlapping regions where agents observe the same targets, leading to unreliable fusion and degraded perception. After calibration, outputs are well aligned, maintaining high spatial consistency even under large delays. This validates the module's ability to reduce spatio-temporal discrepancies and enhance cooperative perception robustness.

5.4 Ablation Study

To evaluate the effectiveness of each IPDA module, we conduct ablation experiments on the DAIR-V2X (Yu et al. 2022) dataset (see Table 2). Module contributions are examined via substitution or removal: 1) delay discrepancy quantification captures dynamic scene changes better than fixed scene-level priors (e.g., highway/city/intersection); 2) motion state judgment adapts to agent dynamics more accurately than fixed curvature settings (e.g., zero/mean/extreme); 3) delay-weighted spatial aggregation better handles data with varying delays than IoU-based clustering; 4) historical alignment attention reduces discontinuities and drift, enhancing detection.

6 Conclusion

In this paper, we propose IPDA, a spatio-temporal co-sensing calibration method to resolve delay-aligned data inconsistency in cooperative agent sensing. IPDA adopts a temporal-spatial cascade architecture, modeling alignment correlations for temporal correction and constructing a Co-Sensing Calibration Network for spatial refinement. This enhances multi-agent data consistency and significantly improves fusion accuracy. Experiments show that IPDA achieves strong accuracy and robustness in complex traffic scenarios, supporting practical deployment. Future work will explore scalability, multimodal fusion, and real-time and security optimization in large-scale cooperative systems.

Acknowledgments

This work is supported by the Natural Science Foundation of Shandong Province under Grants ZR2025MS1093.

References

Chiu, H.-K.; Wang, C.-Y.; Chen, M.-H.; and Smith, S. F. 2024. Probabilistic 3D Multi-Object Cooperative Tracking for Autonomous Driving via Differentiable Multi-Sensor Kalman Filter. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 18458–18464.

Ding, Z.; Fu, J.; Liu, S.; Li, H.; Chen, S.; Li, H.; Zhang, S.; and Zhou, X. 2025. Point Cluster: A Compact Message Unit for Communication-Efficient Collaborative Perception. In *Proc. Int. Conf. Learn. Representations (ICLR)*.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An Open Urban Driving Simulator. In *Proc. Conf. Robot Learn.*, volume 78 of *Proceedings of Machine Learning Research*, 1–16.

Grisetti, G.; Kümmerle, R.; Stachniss, C.; and Burgard, W. 2010. A Tutorial on Graph-Based SLAM. *IEEE Intell. Transp. Syst. Mag.*, 2(4): 31–43.

Hong, S.; Liu, Y.; Li, Z.; Li, S.; and He, Y. 2024. Multi-agent Collaborative Perception via Motion-aware Robust Communication Network. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 15301–15310.

Hu, J.; Mao, M.; Bao, H.; Zhang, G.; and Cui, Z. 2023. CP-SLAM: Collaborative Neural Point-based SLAM System. In *Adv. Neural Inf. Process. Syst. (NIPS)*, volume 36, 39429–39442.

Hu, Y.; Fang, S.; Lei, Z.; Zhong, Y.; and Chen, S. 2022. Where2comm: Communication-Efficient Collaborative Perception via Spatial Confidence Maps. In *Adv. Neural Inf. Process. Syst. (NIPS)*, volume 35, 4874–4886.

Huang, H.; Li, L.; Cheng, H.; and Yeung, S.-K. 2024. Photo-slam: Real-time simultaneous localization and photorealistic mapping for monocular stereo and rgb-d cameras. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 21584–21593.

Keetha, N.; Karhade, J.; Jatavallabhula, K. M.; Yang, G.; Scherer, S.; Ramanan, D.; and Luiten, J. 2024. SplaTAM: Splat Track & Map 3D Gaussians for Dense RGB-D SLAM. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 21357–21366.

Kümmerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; and Burgard, W. 2011. G2o: A general framework for graph optimization. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 3607–3613.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. PointPillars: Fast Encoders for Object Detection From Point Clouds. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 12689–12697.

Lee, K.; Kim, J.; Park, Y.; Wang, H.; and Hong, D. 2017. Latency of Cellular-Based V2X: Perspectives on TTI-Proportional Latency and TTI-Independent Latency. *IEEE Access*, 5: 15800–15809.

Lei, Z.; Ren, S.; Hu, Y.; Zhang, W.; and Chen, S. 2022. Latency-Aware Collaborative Perception. In *Proc. Eur. Conf. Comput. Vision. (ECCV)*, 316–332. ISBN 978-3-031-19824-3.

Levenberg, K. 1944. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.*, 2(2): 164–168.

Li, Y.; Ma, D.; An, Z.; Wang, Z.; Zhong, Y.; Chen, S.; and Feng, C. 2022. V2X-Sim: Multi-Agent Collaborative Perception Dataset and Benchmark for Autonomous Driving. *IEEE Robot. Autom. Lett.*, 7(4): 10914–10921.

Lu, F.; and Milios, E. 1997. Globally Consistent Range Scan Alignment for Environment Mapping. *Auton. Robots*, 4(4): 333–349.

Lu, Y.; Hu, Y.; Zhong, Y.; Wang, D.; Wang, Y.; and Chen, S. 2024. An Extensible Framework for Open Heterogeneous Collaborative Perception. In *Proc. Int. Conf. Learn. Representations (ICLR)*.

Lu, Y.; Li, Q.; Liu, B.; Dianati, M.; Feng, C.; Chen, S.; and Wang, Y. 2023. Robust Collaborative 3D Object Detection in Presence of Pose Errors. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 4812–4818.

Mehr, E.; Jourdan, A.; Thome, N.; Cord, M.; and Guitteny, V. 2019. DiscoNet: Shapes Learning on Disconnected Manifolds for 3D Editing. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 3473–3482.

Wang, T.-H.; Manivasagam, S.; Liang, M.; Yang, B.; Zeng, W.; and Urtasun, R. 2020. V2VNet: Vehicle-to-Vehicle Communication for Joint Perception and Prediction. In *Proc. Eur. Conf. Comput. Vision. (ECCV)*, 605–621. ISBN 978-3-030-58536-5.

Wang, Z.; Wang, Y.; Wu, Z.; Ma, H.; Li, Z.; Qiu, H.; and Li, J. 2025. CMP: Cooperative Motion Prediction With Multi-Agent Communication. *IEEE Robot. Autom. Lett.*, 10(4): 3876–3883.

Wei, S.; Wei, Y.; Hu, Y.; Lu, Y.; Zhong, Y.; Chen, S.; and Zhang, Y. 2023. Asynchrony-Robust Collaborative Perception via Bird's Eye View Flow. In *Adv. Neural Inf. Process. Syst. (NIPS)*, volume 36, 28462–28477.

Xu, J.; Zhang, Y.; Cai, Z.; and Huang, D. 2025. CoSDH: Communication-Efficient Collaborative Perception via Supply-Demand Awareness and Intermediate-Late Hybridization. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*.

Xu, R.; Guo, Y.; Han, X.; Xia, X.; Xiang, H.; and Ma, J. 2021. OpenCDA: An Open Cooperative Driving Automation Framework Integrated with Co-Simulation. In *Proc. Int. Conf. Intell. Transp. Syst. (ITSC)*, 1155–1162.

Xu, R.; Xiang, H.; Tu, Z.; Xia, X.; Yang, M.-H.; and Ma, J. 2022a. V2X-ViT: Vehicle-to-Everything Cooperative Perception with Vision Transformer. In *Proc. Eur. Conf. Comput. Vision. (ECCV)*, 107–124. ISBN 978-3-031-19842-7.

Xu, R.; Xiang, H.; Xia, X.; Han, X.; Li, J.; and Ma, J. 2022b. OPV2V: An Open Benchmark Dataset and Fusion Pipeline for Perception with Vehicle-to-Vehicle Communication. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2583–2589.

Yang, K.; Yang, D.; Zhang, J.; Li, M.; Liu, Y.; Liu, J.; Wang, H.; Sun, P.; and Song, L. 2023. Spatio-Temporal Domain Awareness for Multi-Agent Collaborative Perception. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 23326–23335.

Yoo, J.; Feng, Z.; Pan, T.-Y.; Sun, Y.; Phoo, C. P.; Chen, X.; Campbell, M.; Weinberger, K. Q.; Hariharan, B.; and Chao, W.-L. 2025. Learning 3D Perception from Others' Predictions. In *Proc. Int. Conf. Learn. Representations (ICLR)*.

Yu, H.; Luo, Y.; Shu, M.; Huo, Y.; Yang, Z.; Shi, Y.; Guo, Z.; Li, H.; Hu, X.; Yuan, J.; and Nie, Z. 2022. DAIR-V2X: A Large-Scale Dataset for Vehicle-Infrastructure Cooperative 3D Object Detection. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 21329–21338.

Yu, H.; Tang, Y.; Xie, E.; Mao, J.; Luo, P.; and Nie, Z. 2023. Flow-Based Feature Fusion for Vehicle-Infrastructure Cooperative 3D Object Detection. In *Adv. Neural Inf. Process. Syst. (NIPS)*.

Yuan, Z.; Deng, J.; Ming, R.; Lang, F.; and Yang, X. 2024. SR-LIVO: LiDAR-Inertial-Visual Odometry and Mapping With Sweep Reconstruction. *IEEE Robot. Autom. Lett.*, 9(6): 5110–5117.

Zhang, G.; Chen, J.; Gao, G.; Li, J.; Liu, S.; and Hu, X. 2024a. SAFDNet: A Simple and Effective Network for Fully Sparse 3D Object Detection. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 14477–14486.

Zhang, J.; Yang, K.; Wang, Y.; Wang, H.; Sun, P.; and Song, L. 2024b. ERMVP: Communication-Efficient and Collaboration-Robust Multi-Vehicle Perception in Challenging Environments. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 12575–12584.

Zhang, T.; Zhang, L.; Chen, Y.; and Zhou, Y. 2022. CVIDS: A Collaborative Localization and Dense Mapping Framework for Multi-Agent Based Visual-Inertial SLAM. *IEEE Trans. Image Process.*, 31: 6562–6576.