# V2XPnP: Vehicle-to-Everything Spatio-Temporal Fusion for Multi-Agent Perception and Prediction

Anonymous ICCV submission

Paper ID 16

## Abstract

*Vehicle-to-everything (V2X) technologies offer a promising paradigm to mitigate the limitations of constrained observability in single-vehicle systems. Prior work primarily focuses on single-frame cooperative perception, which fuses agents' information across different spatial locations but ignores temporal cues and temporal tasks (e.g., temporal perception and prediction). In this paper, we focus on the spatio-temporal fusion in V2X scenarios and design one-step and multi-step communication strategies (when to transmit) as well as examine their integration with three fusion strategies - early, late, and intermediate (what to transmit), providing comprehensive benchmarks with 11 fusion models (how to fuse). Furthermore, we propose **V2XPnP**, a novel intermediate fusion framework within one-step communication for end-to-end perception and prediction. Our framework employs a unified Transformer-based architecture to effectively model complex spatio-temporal relationships across multiple agents, frames, and high-definition map. Moreover, we introduce the **V2XPnP Sequential Dataset** that supports all V2X collaboration modes and addresses the limitations of existing real-world datasets, which are restricted to single-frame or single-mode cooperation. Extensive experiments demonstrate our framework outperforms state-of-the-art methods in both perception and prediction tasks. The codebase and dataset will be released to facilitate future V2X research.*

## 1. Introduction

Autonomous driving systems are required to accurately perceive surrounding road users and predict their future trajectories to ensure safe and interactive driving. Despite recent advances in perception and prediction [10, 16, 38], single-vehicle systems still struggle with limited perception range and occlusion issues [28, 46], compromising driving performance and road safety. Consequently, vehicle-to-everything (V2X) technologies have emerged as a promising paradigm to address these challenges, which enable connected and automated vehicles (CAVs) and infrastructures to share complementary information and mitigate occlusions, thereby supporting holistic environment understanding [13, 17, 19].

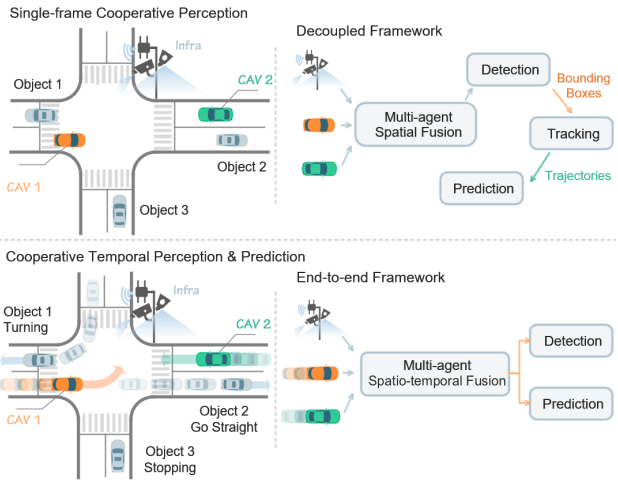



Figure 1. Illustration of V2X temporal tasks and our V2X spatio-temporal fusion framework. By incorporating temporal information, our framework enhances V2X communication and supports end-to-end perception and prediction beyond single-frame perception.

Despite their potential, existing works focus on frame-by-frame cooperative detection [26, 32, 43, 57], which aggregates information from agents at different spatial locations. However, these works overlook temporal cues across sequential frames, which are important for locating previously visible but currently undetected objects [48] and predicting object future trajectories [33]. Although some work [42, 51] incorporate short-term temporal cues (0.5 s) in single-frame perception to mitigate asynchrony, the broader challenge of efficiently aggregating multi-agent and multi-frame information and supporting long-term temporal tasks, such as motion prediction, remains largely unexplored. Therefore, we aim to address critical questions in multi-agent multi-frame cooperation: *(1) What information to transmit? (2) When to transmit it? (3) How to fuse information across multi-agent spatial and temporal dimensions?* To address ***what to transmit***, we expand the three fusion strategies in single-frame

cooperative perception (*i.e.*, early, late, and intermediate) to incorporate the temporal dimension. Regarding ***when to transmit***, we introduce one-step and multi-step communication strategies to capture multi-frame temporal information. For ***how to fuse***, we conduct a systematic analysis across various spatio-temporal fusion strategies, providing comprehensive benchmarks for cooperative perception and prediction tasks across all V2X collaboration modes.

Among these strategies, we advocate intermediate fusion within one-step communication, because it effectively balances the trade-off between accuracy and increased transmission load. Moreover, its capability to transmit intermediate spatial-temporal features makes it well-suited for end-to-end perception and prediction, allowing for feature sharing across multiple tasks, as shown in Fig. 1. Based on this strategy, we propose ***V2XPnP***, a V2X spatio-temporal fusion framework leveraging a unified Transformer structure for effective spatial-temporal fusion, encompassing temporal attention, self-spatial attention, multi-agent spatial attention, and map attention. Each agent first extracts its inter-frame and self-spatial features, which can support single-vehicle perception and prediction while reducing the communication load, and then the multi-agent spatial attention model fuses the single-agent feature across different agents.

Another challenge is the lack of real-world sequential datasets encompassing diverse V2X collaboration modes. In V2X scenarios, vehicles and infrastructures serve as primary agents, with collaboration modes that include vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and infrastructure-to-infrastructure (I2I) [19]. Most existing datasets [44, 49] are non-sequential, limited to single collaboration modes, and focus only on single-frame cooperative perception, lacking support for temporal tasks. To bridge this gap, we introduce the first large-scale real-world ***V2XPnP Sequential Dataset***, featuring four agents and supporting all collaboration modes. This dataset includes temporally consistent perception and trajectory data across 100 vehicle-centric (VC) scenarios and 63 infrastructure-centric (IC) scenarios, totaling 40k frames, along with point-cloud and vector maps from 24 collected intersections. The main contributions of this paper can be summarized as follows:

1. We present *V2XPnP*, a V2X spatio-temporal fusion framework with a novel intermediate fusion model within one-step communication. This framework is based on unified Transformer architecture integrating diverse attention fusion modules for V2X spatial-temporal information.
2. We introduce the first large-scale, real-world V2X sequential dataset featuring multiple agents and all V2X collaboration modes (*i.e.*, VC, IC, V2V, I2I), encompassing perception data, object trajectories, and map data.
3. We conduct extensive analysis across various spatio-temporal fusion strategies and benchmarks 11 fusion models for cooperative perception and prediction in all V2X collaboration modes, demonstrating the state-of-the-art performance of the proposed model.

## 2. Related Work

**End-to-end Perception and Prediction.** Safe autonomous driving fundamentally depends on accurate perception and prediction [18, 54]. In single-agent systems, significant efforts have been dedicated to temporal perception and prediction [29], leading to the development of various end-to-end frameworks. These frameworks enhance computational efficiency by sharing information across tasks and mitigating error propagation inherent in modular architectures. FaF [33] and PnPNet [30] focus on Lidar-based joint perception and prediction, while occupancy flow methods [2, 3] provide detailed spatial-temporal information. Recently, Bird's-Eye-View (BEV)-based approaches with camera data have gained prominence [12, 16, 21].

**V2X Perception and Prediction.** V2X perception has been extensively explored, with intermediate fusion emerging as a widely adopted strategy [10, 45, 46]. FFNet [51] and CoBEVFlow [42] utilize historical information (0.5s) from collaborators to mitigate asynchrony, and SCOPE integrates ego-history for detection. However, coordinating multi-agent systems for long-term temporal tasks remains an open challenge. In the prediction domain, deep learning models have been extensively studied for modeling inter-agent interactions [20, 37]. However, the limited short-term visibility of individual vehicles continues to restrict prediction accuracy. Cooperative prediction leveraging V2X has shown potential, though research remains preliminary [34, 41, 48, 53]. To integrate perception and prediction within an end-to-end framework, V2VNet [40] employs a graph neural network for spatio-temporal fusion. UniV2X [52] extended the end-to-end system to support downstream tasks, but with a simplified spatio-temporal fusion module. Despite these advancements, a comprehensive framework for V2X-based spatio-temporal fusion is still lacking.

**Real-world Driving Datasets.** Public datasets have been instrumental in advancing autonomous driving research. Early sequential datasets [1, 23] provided only object trajectories on highways [56], but lack perception data. The following datasets, such as nuScenes [5] and Waymo [39], introduced real-world urban data but were limited to single-agent perspectives, rendering them unsuitable for V2X research. Thus, simulated datasets like V2XSet [45] were developed. Recently, datasets including Dair-V2X [49], V2V4Real [47], RCooper [14], and V2X-Real [44] significant contributed to real-world data in the V2I, V2V, I2I and V2X modes. However, real-world sequential V2X datasets covering all collaboration modes remain scarce. V2X-Seq [50] is the only sequential dataset incorporated with various behavior and map data for prediction tasks; however, is limited to V2I data and has restricted accessibility with download constraints.

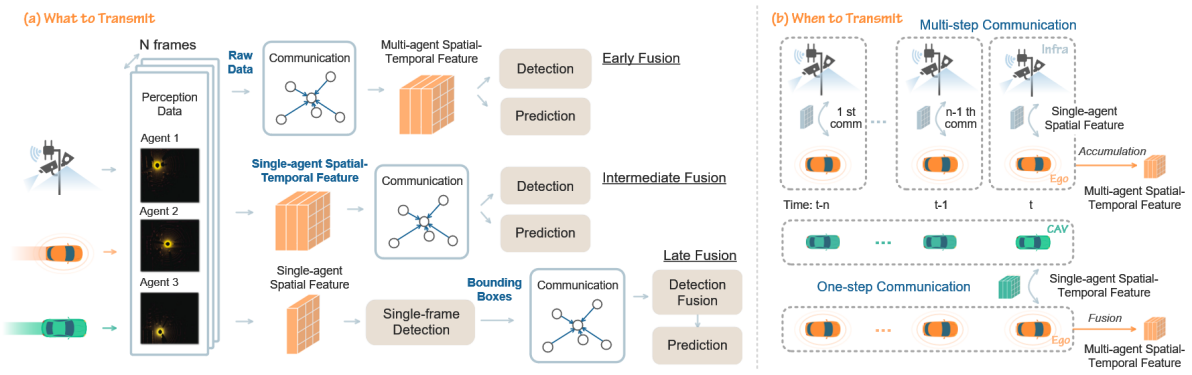



Figure 2. Illustration of various V2X fusion strategies for perception and prediction. (a) *What to Transmit*: early, intermediate, and late fusion, transmitting raw sensor data, intermediate BEV features, or bounding boxes. (b) *When to Transmit*: multi-step and one-step communication.

# 3. V2XPnP Fusion Framework

The cooperative temporal perception and prediction task requires the integration of temporal information across historical frames and spatial information from multiple agents. It is defined as follows: given map and historical sequences of raw perception data $\mathbf{P}_i^t, i \in \{1, \cdots, N\}$, from all $N$ agents within the communication range of the ego agent, the objective is to detect objects in the current frame and predict their future trajectories considering the map information.

## 3.1. V2X Spatio-Temporal Fusion Strategies

This section addresses what and when to transmit for spatio-temporal fusion, and Sec. 3.2 delves into the fusion process.

**What to Transmit.** In multi-agent spatial fusion, three fusion strategies are widely adopted in single-frame cooperative perception, *i.e.*, early, late, and intermediate fusion [15, 49], which involve the transmission of raw perception data, bounding boxes, and intermediate features, respectively. Adopting this framework, we extend these fusion approaches to the V2X spatio-temporal fusion context, as illustrated in Fig. 2. (1) *Early fusion* transmits the entire raw historical perception data to retain complete feature information but imposes the highest transmission load. (2) *Late fusion* shares only the final detected results at each historical frame, resulting in the lowest transmission load but losing most of the feature information. (3) *Intermediate fusion* transmits intermediate spatio-temporal BEV features, striking a balance between information quality and transmission load.

**When to Transmit.** Determining when to transmit in spatio-temporal fusion is more challenging due to the inclusion of temporal data as compared to the existing spatial fusion. A straightforward approach is *Multi-step Communication*, where each transmission only includes the current frame's data. However, the multiple steps cause the accumulation of delays and data loss, and obtaining complete historical data requires that neighboring agents stay within the ego agent's limited communication range throughout historical frames. In practice, the ego agent should aggregate as much data as possible from other agents within a single transmission, rather than relying on multiple exchanges to obtain complete spatio-temporal information. To address this, we propose a *One-step Communication* strategy, where individual agents share all their historical data within a single communication.

**Intermediate Fusion with One-step Communication.** Synthesizing the considerations for *what to transmit* and *when to transmit*, we adopt an intermediate fusion within the one-step communication strategy, which fuses temporal data from multiple frames before transmission, allowing each agent to share aggregated information without excessive communication overhead. In multi-step intermediate fusion, each agent transmits BEV feature maps at every timestep $\mathbf{F}_i^t \in \mathbb{R}^{H \times W \times C}$, which denotes agent $i$'s BEV feature at time $t$ with height $H$, width $W$, and channels $C$. The cumulative data shared across $T$ frames is a stacked sequence $\mathbf{F}_i^{seq} \in \mathbb{R}^{T \times H \times W \times C}$, resulting in a significantly higher transmission load than single-frame, along with potential delays and data loss. Conversely, in our proposed strategy, each agent first fuses its historical BEV features internally, reducing the sequence from $\mathbf{F}_i^{seq}$ to a single condensed BEV feature map $\mathbf{F}_i^{'} \in \mathbb{R}^{H \times W \times C}$. This reduction allows agents to transmit a compact data packet, comparable in size to single-frame cooperative perception, thereby conserving bandwidth while preserving essential spatio-temporal information.

## 3.2. V2XPnP Framework

The spatio-temporal features of intermediate fusion render it a natural fit for end-to-end perception and prediction. Accordingly, we propose a unified end-to-end perception and prediction framework to perform multiple tasks across spatio-temporal dimensions. The overall V2XPnP framework is illustrated in Fig. 3, which includes six components and is unfolded in this section. The detail of the spatio-temporal fusion model is provided in Sec. 3.3. Notably, each module in V2XPnP is modular, allowing for easy replacement.

**V2X Metadata Sharing.** Each agent in the V2X system is an observer and collaborator in the shared environment.

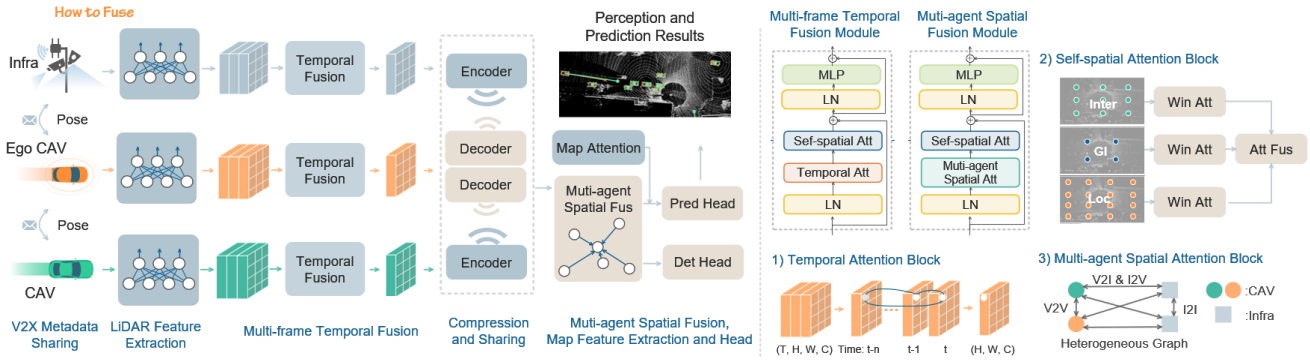



Figure 3. The V2XPnP framework and multi-agent spatio-temporal fusion model. The framework comprises various components for feature extraction, fusion, and decoding. Within our fusion model, we introduce multiple attention mechanisms to enhance the fusion process.

Each agent will first determine its collaborators with its communication range, and share the metadata, such as relative poses and extrinsic, to construct a spatial V2X graph. Each node and edge in the graph represents an agent and a communication channel. All the point clouds are transformed into the ego agent's coordinate frame before feature extraction.

**LiDAR Feature Extraction.** We utilize the PointPillar network [24] to extract the LiDAR feature for each agent $i$ at time $t$, which has low inference latency. The extracted features are structured into a 2D pseudo-image representation to produce salient feature maps $\mathbf{F}_i^t$, and the sequential feature is stacked as $\mathbf{F}_i^{seq} \in \mathbb{R}^{T\times H\times W\times C}$

**Multi-frame Temporal Fusion.** We propose a Transformer-based temporal fusion module to iteratively perform inter-frame and intra-agent BEV feature fusion through self-attention mechanisms. The spatio-temporal feature of each agent is extracted through the temporal dimension while minimizing communication overhead. The feature map for each agent after temporal fusion is $\mathbf{F}_i^{'} \in \mathbb{R}^{H\times W\times C}$.

**Compression and Sharing.** To reduce the transmission load, intermediate features are compressed using a $1 \times 1$ kernel convolution network in its channel dimension. The ego agent uses another convolution network to decompress the features, restoring them to their original dimensionality.

**Multi-agent Spatial Fusion.** The decompressed features are passed into a Transformer-based multi-agent spatial fusion network to learn inter-agent and intra-agent spatio-temporal interaction and update the multi-agent feature map $\mathbf{F}^{'}$.

**Map Feature Extraction.** The HD map is directly accessed for each agent without V2X fusion. We project the vectorized HD map to BEV space by incorporating the map polylines into each BEV feature grid. We first employ a multi-layer perceptron (MLP) to encode the surrounding map polylines for each grid $\mathbf{M} \in \mathbb{R}^{H\times W\times N_m\times n\times D}$, resulting in the map feature $\mathbf{F}_m \in \mathbb{R}^{H\times W\times N_m\times C}$. Here, $N_m$ and $n$ represent the number of map polylines and the waypoints per polyline, while $D$ represents waypoint attributes (*i.e.*, position and lane type). The map encoding is expressed as: $\mathbf{F}_m = \phi\left(\mathrm{MLP}\left(\mathbf{M}\right)\right)$, where $\phi$ denotes max-pooling on the waypoint axis. Then, a map-BEV attention is introduced to inject the map feature into the BEV feature. We concatenate the BEV and map feature $\mathbf{F}_{bm} = [\mathbf{F}^{'}, \mathbf{F}_m] \in \mathbb{R}^{H\times W\times(1+N_m)\times C}$, and add the position embedding $\mathbf{P}_m$ based on sinusoidal positional encoding. The final content feature map $\mathbf{F} \in \mathbb{R}^{H\times W\times C}$ is updated by multi-head self-attention (MHSA) as follows:

$$\mathbf{F} = \mathrm{MHSA}\big(\mathrm{Q}\colon [\mathbf{F}_{bm},\mathbf{P}_m], \mathrm{K}\colon [\mathbf{F}_{bm},\mathbf{P}_m], \mathrm{V}\colon \mathbf{F}_{bm}\big). \quad (1)$$

**Detection and Prediction Heads.** Finally, a detection head and a prediction head are connected to the final feature $\mathbf{F}$ to output states for each predefined anchor box: (1) The detection head contains two 2D convolution layers for bounding box regression and classification. The regression branch outputs $(x, y, z, w, l, h, \theta)$, which represents the bounding box position, size, and yaw. The classification branch outputs the confidence score of each anchor box, determining whether it corresponds to an object or background. (2) The prediction head outputs offset values for each anchor box at each future timestamp using two 2D convolution layers, and the final trajectory is generated by accumulating these offsets.

### 3.3. Spatio-Temporal Fusion Transformer

In this section, we introduce spatio-temporal fusion with a unified Transformer architecture. The proposed model comprises three blocks: temporal attention, self-spatial attention, and multi-agent spatial attention, as shown in Fig. 3, and two core fusion modules. (1) *Multi-frame temporal fusion*: Each agent first extracts their spatio-temporal features through iterative temporal and self-spatial attentions. (2) *Multi-agent spatial fusion*: rich BEV features from multiple agents are acquired via V2X and then fused through iterative multi-agent spatial and self-spatial attentions.

**Temporal Attention.** This block is designed to capture the inter-frame relationship and aggregate historical BEV features $\mathbf{F}_i^{seq}$ across the temporal dimension. The history timestamps are encoded with a learnable embedding, which

is added to each BEV feature frame to form $\mathbf{F}_i^{seq'}$. To preserve temporal cues, this block only fuses temporal features from the same spatial positions across frames, and spatial features are further extracted by the self-spatial Transformer. Temporal fusion is expressed as:

$$\mathbf{F}_i^{tem} = \mathrm{MHSA}\big(\mathrm{Q}\colon\ \mathrm{MLP}(\mathbf{F}_i^{seq'}), \mathrm{K}\colon\ \mathrm{MLP}(\mathbf{F}_i^{seq'}), \mathrm{V}\colon\ \mathrm{MLP}(\mathbf{F}_i^{seq'})\big). \quad (2)$$

**Self-spatial Attention.** To capture the intra-agent spatial BEV interaction, this block employs multi-scale window attention to capture spatial features at different resolutions and ranges. A large window focuses on global features for long-term behavior, and a small window preserves local finer information. Note that this block only fuses spatial features for each agent in a frame without inter-frame and inter-agent fusion. Specifically, we utilize local, intermediate, and global windows $P_k \in \{P_{loc}, P_{inter}, P_{gl}\}$ to partition the feature map through the $H$ and $W$ dimension, generating the self-spatial token $\mathbf{F}^{sp} \in \mathbb{R}^{\frac{H}{P_k} \times \frac{W}{P_k} \times {P_k}^2 \times C}$. With an additional relative position encoding, MHSA is operated among ${P_k}^2$ tokens. The final output is obtained by performing split attention to fuse features from different windows.

**Multi-agent Spatial Attention.** This block facilitates inter-agent fusion by aggregating BEV feature maps from multiple agents. Considering the different deployment positions and capabilities of vehicle and infrastructure sensors, the multi-agent spatial Transformer is heterogeneous with individual learnable weights for different interaction pairs (*i.e.*, V-I, V-V, I-V, and I-I). The attention token of $i$ th agent $\mathbf{F}_{i,m}^{sp}$ is modulated by its type $m$ embedding, and is weighted by the relation matrix $\mathbf{W}_{\text{att}}^{(e_{i,j})}$ between edge $e_{i,j}$ during aggregation with agent $j$ of type $n$:

$$\mathbf{Q}_i^m = \mathrm{MLP}(\mathbf{F}_{i,m}), \mathbf{K}_j^n = \mathrm{MLP}(\mathbf{F}_{j,n}), \mathbf{V}_j^n = \mathrm{MLP}(\mathbf{F}_{j,n}),$$
$$\mathbf{F}_{i,m}^{sp} = \sum_j \mathrm{Softmax}(\mathbf{Q}_i^m \cdot \mathbf{W}_{\text{att}}^{(e_{i,j})} \cdot \mathbf{K}_j^n) \cdot \mathbf{V}_j^n. \quad (3)$$

### 3.4. Learning Objective

The learning objective is comprised of temporal perception and prediction tasks. First, we define the perception loss as the combination of regression $\mathcal{L}_{\text{reg}}$ and classification loss $\mathcal{L}_{\text{cla}}$ of the predefined anchor box. Specifically, the smooth $\ell_1$-loss is leveraged for the regression part, and the focal loss [31] is utilized for classification. Second, we define the prediction loss $\mathcal{L}_{\text{pred}}$ as $\ell_2$-loss between the prediction points sequence with ground truth trajectory. The final loss function is the weighted sum of $\mathcal{L}_{\text{reg}}$ , $\mathcal{L}_{\text{cla}}$ , and $\mathcal{L}_{\text{pred}}$ .

## 4. V2XPnP Sequential Dataset

We introduce the V2XPnP-Sequential dataset, the first large-scale, real-world V2X sequential dataset featuring multiple agents and all collaboration modes. This dataset comprises 100 scenarios (49 2V+2I scenarios, 42 V2V scenarios, and 9 V+2I scenarios), each spanning 95 to 283 frames captured at 10 Hz. The dataset comprises two data sequences from CAV perception (point clouds and camera images) and two data sequences from infrastructure perception, as shown in Fig. 4(b). We also provide corresponding vector maps and point-cloud maps for all collection areas, as shown in Fig. 4(c). Ten object categories are included, and the average trajectory length and frequency of each category are shown in Fig. 4(d). Further details on data visualization, annotation, trajectory and map generation are provided in the supplementary.

### 4.1. Data Acquisition

V2X temporal tasks require diverse time-consistent perception data and object behavior data. We choose urban arterial roads, expressways, and intersections to collect V2X data. The sensor configurations for two CAVs and two infrastructures are shown in Fig. 4(a). By permuting and combining the behavior patterns of CAVs (such as overtaking, platooning, turning, etc.), we designed a total of about 60 interaction pairs between CAVs to collect data. From more than 66h driving logs, we annotate 32 representative scenarios. We also process the non-sequential data from V2X-Real [44] with a V2X sequential data processing pipeline, detailed in Sec. 4.2. Since perception tasks do not need to consider labeling consistency, identical objects may be assigned different IDs within a sequence, leading to data fragmentation and limiting support for temporal tasks. The final dataset is yielded by processing our collected data and V2X-Real data.

### 4.2. Sequential Data Processing

Time-consistent data is crucial for temporal tasks, and thus we develop a V2X sequential data processing pipeline to track objects across time and different agents' views. We construct a multi-agent spatio-temporal graph, where each node $n_{it}^k$ represents an annotated bounding box $i$ at time $t$ from agent $k$'s LiDAR data, and edges connect nodes that correspond to the same object. Identifying objects with the same ID then reduces to finding connected components in this graph, with each isolated component representing a unique object. Moreover, we build this graph by leveraging the temporal continuity in single-agent annotations and incorporating multi-agent data. For temporal consistency within a single agent, we add edges between $n_{it}^k$ and $n_{it+1}^k$, although annotation errors may sometimes assign different IDs to the same object. To address this, we integrate annotations from multiple agents, transforming annotations into a global coordinate frame and connecting nodes if their Intersection-over-Union (IoU) exceeds a threshold. Once the graph is complete, we identify connected components and assign each a unique tracking ID. To mitigate annotation biases, we refine object attributes based on their consensus.

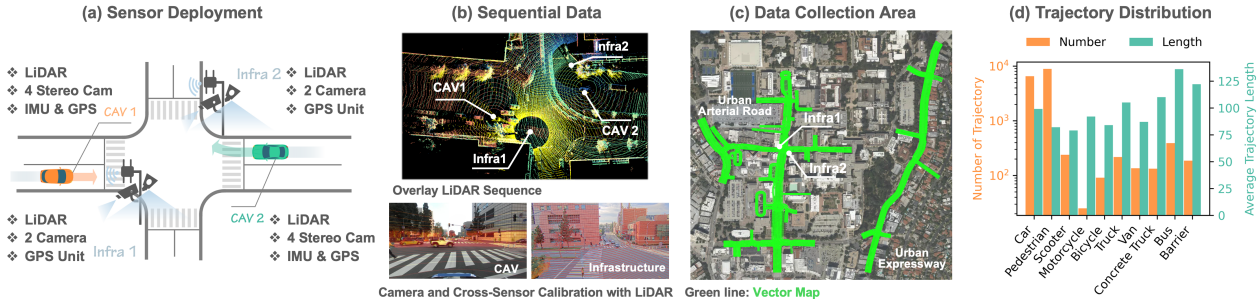



Figure 4. Illustration of the V2XPnP Sequential Dataset. (a) V2X data acquisition systems; (b) Sequential LiDAR and camera data; (c) Data collection area and vector map; (d) Total number and average tracking length of 3D tracked objects per category.

# 5. Experiments

## 5.1. Experimental Setup

**Evaluation Metrics**. Following the detection evaluation protocol in [44], we measure detection performance using Average Precision (AP) at an IoU threshold of 0.5. For prediction, we provide the results of commonly used metrics [11, 35, 36], including Average Displacement Error (ADE), Final Displacement Error (FDE), and Miss Rate (MR) within a 2-meter threshold. However, prediction accuracy is influenced by false positives and missed detections from the perception module. For example, a poorly performing perception module that detects only a simple object in a straight-line trajectory can misleadingly inflate the prediction accuracy. To address this, we employ the *End-to-end Perception and Prediction Accuracy* (EPA) metric [12] to jointly evaluate perception and prediction performance.

$$\text{EPA} = \frac{|\hat{S}_{\text{match, hit}}| - \alpha N_{\text{FP}}}{N_{\text{GT}}}, \tag{4}$$

where $|\hat{S}_{\text{match, hit}}|$ is the number of true positive objects with prediction $FDE < \tau_{EPA}$, $N_{FP}, N_{GT}$ represent the number of false positive objects and ground truth objects, respectively, and $\alpha$ is a penalty coefficient. A higher EPA value indicates superior object detection and prediction capabilities, and we set $\tau_{EPA} = 2$m, $\alpha = 0.5$ following [12].

**Collaboration Modes**. The V2XPnP Sequential Dataset supports various V2X collaboration modes by organizing data with specific interaction patterns. *Vehicle-Centric (VC)*: The ego CAV is the focal agent, communicating with other CAVs and infrastructure (Infra). *Infrastructure-Centric (IC)*: Infrastructure is the central entity, communicating with other Infras and CAVs. *Vehicle-to-Vehicle (V2V)*: The ego CAV communicates exclusively with other CAVs without involving Infra. *Infrastructure-to-Infrastructure (I2I)*: Infra shares data only with each other. Each VC and IC scenario includes 2-4 agents, which is close to real-world V2X settings and can evaluate model generalization across diverse V2X scenarios, whereas each V2V and I2I scenario has two agents.

**Implementation Details**. During the testing stage, we select a fixed agent as the ego agent in each cooperative scenario, while the ego agent is shuffled and randomly selected during training. Following the real-time setting [55], we set the communication range to 50 meters and evaluate surrounding agents within a range of $x \in [-70, 70]$m and $y \in [-40, 40]$m. Messages beyond 50 meters are discarded. Besides, the history length is 2s (2Hz), and the prediction horizon is 3s (2Hz). The train/validation/test data splits are 80/6/14 scenarios. Additional training and model details are provided in the supplementary materials.

## 5.2. Benchmark Methods

**End-to-end methods.** Most of the single-frame perception models cannot support the prediction task, thus, we first implement a baseline end-to-end model with the same LiDAR backbone and decoding heads as V2XPnP but utilizing the temporal fusion module FaF [33] - alternating 2D and 3D convolutions - as the *No fusion-FaF** baseline, which can extend the non-temporal model to support temporal tasks. Then, the early fusion configurations and several state-of-the-art intermediate fusion models are integrated as: *Early Fusion*, *FFNet** [51], *CoBEVFlow** [42], *V2X-ViT** [45]). More benchmark results of *DiscoNet** [27], *F-Cooper** [6], and *V2VNet** [40] are provided in the supplementary. These models marked with * are reimplemented in our framework with the same LiDAR backbone and decoding heads.

**Decoupled methods.** Transmitting final detection results renders late fusion incompatible with end-to-end models. Thus, we benchmark *Late Fusion* with a decoupled pipeline, where objects in each historical frame are detected using a single-frame perception module, and results are fused via non-maximum suppression. Assuming an ideal tracker to generate object trajectories from perception results and interpolate missing points, we implement an attention-based predictor for trajectory-level prediction tasks, following the prediction mainstream [37]. To further assess end-to-end performance, we also evaluate a decoupled *No-Fusion* model.

## 5.3. Results

Tab. 1 presents the benchmark results across four V2X collaboration modes. Since prediction performance inherently

Table 1. Benchmark results of cooperative perception and prediction models on V2XPnP Sequential (V2XPnP-Seq) Dataset

| Dataset | Method | E2E | Map | **AP@0.5** (%) ↑ | ADE (m) ↓ | FDE (m) ↓ | MR (%) ↓ | **EPA** (%) ↑ |
|---|---|---|---|---|---|---|---|---|
| V2XPnP-Seq-VC *(with V+2I at most)* | No Fusion | | ✓ | 43.9 | 1.87 | 3.24 | 33.8 | 24.3 |
| | No Fusion-FaF* [33] | ✓ | | 53.4 | 1.55 | 2.81 | 34.3 | 31.6 |
| | Late Fusion | | ✓ | 58.1 | 1.59 | 2.82 | 32.4 | 33.0 |
| | Early Fusion | ✓ | ✓ | 60.3 | 1.37 | 2.49 | 33.8 | 36.7 |
| | CoBEVFlow* [42] | ✓ | ✓ | 63.3 | 1.36 | 2.49 | 33.0 | 41.9 |
| | FFNet* [51] | ✓ | ✓ | 64.6 | 1.36 | 2.47 | 34.7 | 42.3 |
| | V2X-ViT* [45] | ✓ | ✓ | 69.6 | 1.39 | 2.56 | 35.2 | 44.7 |
| | V2XPnP (Ours) | ✓ | ✓ | **71.6** +2.0 | 1.35 | 2.36 | 31.7 | **48.2** +3.5 |
| V2XPnP-Seq-IC *(with 2V+I at most)* | No Fusion | | ✓ | 46.4 | 1.69 | 3.06 | 36.2 | 28.8 |
| | No Fusion-FaF* [33] | ✓ | | 56.7 | 1.34 | 2.65 | 41.4 | 31.7 |
| | Late Fusion | | ✓ | 55.9 | 1.39 | 2.44 | 30.1 | 32.9 |
| | Early Fusion | ✓ | ✓ | 60.5 | 1.39 | 2.63 | 32.8 | 39.5 |
| | CoBEVFlow* [42] | ✓ | ✓ | 57.6 | 1.38 | 2.58 | 31.0 | 32.5 |
| | FFNet* [51] | ✓ | ✓ | 61.0 | 1.18 | 2.18 | 35.1 | 37.5 |
| | V2X-ViT* [45] | ✓ | ✓ | 69.3 | 1.27 | 2.39 | 35.4 | 43.3 |
| | V2XPnP (Ours) | ✓ | ✓ | **71.0** +1.7 | 1.18 | 2.16 | 34.0 | **46.0** +2.7 |
| V2XPnP-Seq-V2V | No Fusion | | ✓ | 40.8 | 1.99 | 3.38 | 34.0 | 19.8 |
| | No Fusion-FaF* [33] | ✓ | | 51.9 | 1.67 | 3.12 | 39.3 | 27.5 |
| | Late Fusion | | ✓ | 55.3 | 1.75 | 3.07 | 34.0 | 30.5 |
| | Early Fusion | ✓ | ✓ | 53.0 | 1.64 | 3.11 | 40.2 | 26.9 |
| | CoBEVFlow* [42] | ✓ | ✓ | 58.7 | 1.72 | 3.15 | 40.3 | 33.6 |
| | FFNet* [51] | ✓ | ✓ | 56.5 | 1.68 | 3.12 | 39.8 | 31.2 |
| | V2X-ViT* [45] | ✓ | ✓ | 64.6 | 1.68 | 3.13 | 39.8 | 36.7 |
| | V2XPnP (Ours) | ✓ | ✓ | **70.5** +5.9 | 1.78 | 3.28 | 39.9 | **40.6** +3.9 |
| V2XPnP-Seq-I2I | No Fusion | | ✓ | 51.0 | 1.69 | 3.06 | 36.2 | 31.7 |
| | No Fusion-FaF* [33] | ✓ | | 56.6 | 1.34 | 2.65 | 41.4 | 31.7 |
| | Late Fusion | | ✓ | 61.3 | 1.41 | 2.50 | 30.0 | 41.6 |
| | Early Fusion | ✓ | ✓ | 64.6 | 1.57 | 2.98 | 39.9 | 37.7 |
| | CoBEVFlow* [42] | ✓ | ✓ | 58.4 | 1.31 | 2.61 | 41.5 | 33.0 |
| | FFNet* [51] | ✓ | ✓ | 66.1 | 1.41 | 2.59 | 36.3 | 40.9 |
| | V2X-ViT* [45] | ✓ | ✓ | 65.4 | 1.22 | 2.33 | 35.9 | 41.3 |
| | V2XPnP (Ours) | ✓ | ✓ | **69.2** +3.1 | 1.26 | 2.31 | 36.5 | **42.8** +1.2 |

Table 2. Comparison of one-step and multi-step communication

| Strategy | **AP@0.5** ↑ | ADE ↓ | FDE ↓ | MR ↓ | **EPA** ↑ |
|---|---|---|---|---|---|
| Multi-step | 68.2 | 1.56 | 2.84 | 31.8 | 43.0 |
| One-step | 71.6 | 1.35 | 2.36 | 31.7 | 48.2 |

Table 3. Ablation results of V2XPnP model

| Temp | Spatial | Map | **AP@0.5** ↑ | ADE ↓ | FDE ↓ | MR ↓ | **EPA** ↑ |
|---|---|---|---|---|---|---|---|
| | | | 43.9 | - | - | - | - |
| ✓ | | | 57.2 | 1.52 | 2.76 | 35.5 | 33.8 |
| ✓ | ✓ | | 71.3 | 1.48 | 2.70 | 36.2 | 44.4 |
| ✓ | ✓ | ✓ | **71.6** | **1.35** | **2.36** | **31.7** | **48.2** |

depends on detection quality, the difficulty posed by different detected objects can significantly impact prediction accuracy. Thus, *the EPA metric serves as the most appropriate indicator for assessing the overall performance*. More details and baseline results are provided in the supplementary.

**What to Transmit.** In both end-to-end and decoupled PnP frameworks, the cooperation perception and prediction performance are consistently better than non-fusion models in all collaboration modes, especially in the primary EPA metric, which demonstrates the benefits of cooperation in temporal perception and prediction. Moreover, intermediate fusion models (*e.g.*, V2X-ViT, FFNet, and V2XPnP) generally outperform other fusion strategies, while early fusion consistently surpasses late fusion. Our proposed V2XPnP model achieves the best performance, outperforming other competitive cooperative methods.

**When to Transmit.** Tab. 2 shows the performance of our model under two communication strategies. We find that the perception and prediction performance of one-step communication is improved compared to multi-step communication by 5.0% AP and 12% in EPA. This improvement arises because each agent first directly fuses their lossless raw temporal data before sharing, avoiding error accumulation from lossy intermediate information transformed in multi-step communication across temporal dimensions. Moreover, the one-step strategy compresses spatio-temporal feature transmission (under a 32× compression rate) from 5×0.269 Mb to 0.269 Mb compared to multi-step communication, while mitigating information loss when agents move out of communication range in historical frames. At a typical C-V2X data transmission rate [4], the transmission delay of one-step communication is approximately $10 \sim 20$ ms.

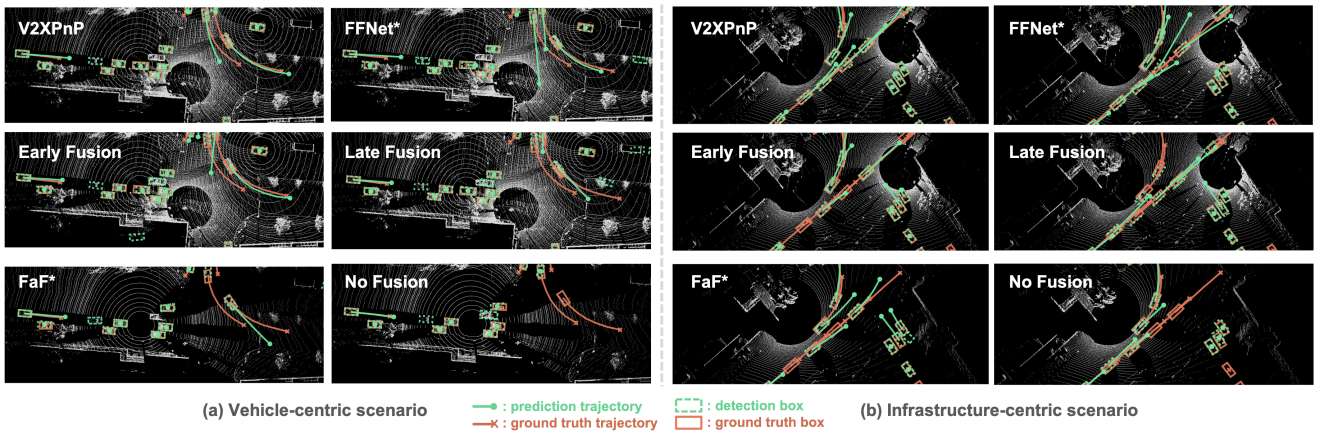



Figure 5. Qualitative results of different fusion models on the testing set. V2XPnP shows better perception and prediction results.

**How to Fuse.** Tab. 3 provides the ablation study of V2XPnP, showing the effectiveness of different components in V2XPnP. The temporal fusion module provides the history information for current-frame detection, while multi-agent spatial fusion alleviates occlusions and improves performance by incorporating other views. The map fusion module enhances trajectory prediction by guiding future trajectories to align with road structures. Our complete V2XPnP model with all these fusion modules performs the best.

**End-to-end vs. Decoupled Frameworks.** The end-to-end model consistently outperforms the decoupled framework. In no-fusion situations, the end-to-end model FaF* outperforms the decoupled model in detection by leveraging temporal cues. Furthermore, FaF* achieves performance comparable to late fusion with V2X spatial aggregation due to integrating temporal features. The intermediate fusion of spatio-temporal features in V2XPnP aligns well with the end-to-end architecture, showcasing its superior performance.

**Infrastructure vs. Vehicle Centric.** As shown in Tab. 1, models under VC and IC modes outperform those in V2V and I2I modes, because VC and IC can aggregate information from up to four agents rather than only two, resulting in enhanced environmental understanding. Notably, the evaluation protocol is consistent across all models within each mode. Additionally, stationary infrastructure-based agents in IC and I2I modes offer higher prediction accuracy by providing elevated sensing perspectives and less noisy data.

**Transmission Data Size and Robustness Test.** Results in Fig. 6 indicate that V2XPnP achieves a good balance between communication efficiency and accuracy, maintaining superior performance compared to full-size V2X-ViT* even at a $\times 128$ compression rate. Following the setting in [45], we provide the results with 100-500 ms time delay and positional/head Gaussian noise from (0.2m, 0.2°) to (1m, 1°). Both V2XPnP and V2X-ViT* maintain robust performance due to their spatial attention fusion, and V2XPnP performs better due to designed temporal attention.

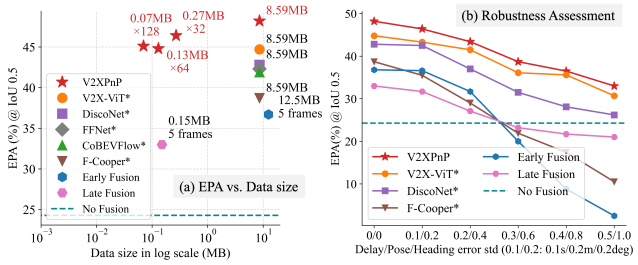



Figure 6. Transmission data size and communication noise test results. V2XPnP shows better performance with varying data compression rates and robustness under communication noise.

**Qualitative Results.** Fig. 5 visualizes the outcomes of cooperative perception and prediction across different fusion models. The No Fusion model is constrained by its limited field of view. The FaF model, leveraging temporal information within an end-to-end pipeline, performs better under occlusion. Late and early fusion models significantly benefit from multi-agent data integration, though late fusion remains impacted by error propagation, such as detection heading errors misleading trajectory direction. Notably, the end-to-end intermediate fusion model, particularly V2XPnP, performs better in both detection and prediction tasks.

## 6. Conclusions

We propose V2XPnP, a novel V2X spatio-temporal fusion framework for cooperative temporal perception and prediction. The core of this framework is a unified Transformer-based model for spatio-temporal fusion and map fusion. Furthermore, we examine various fusion strategies concerning what, when to transmit, and how to fuse, offering comprehensive benchmarks. Additionally, we introduce the V2X Sequential Dataset, which supports all V2X collaboration modes. Extensive experiments demonstrate the superior performance of the proposed framework, establishing its efficacy in advancing cooperative temporal tasks.

# References


[1] Next Generation Simulation (NGSIM) Vehicle Trajectories and Supporting Data | USDOT Open Data. 2

[2] Ben Agro, Quinlan Sykora, Sergio Casas, and Raquel Urtasun. Implicit occupancy flow fields for perception and prediction in self-driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1379–1388, 2023. 2

[3] Ben Agro, Quinlan Sykora, Sergio Casas, Thomas Gilles, and Raquel Urtasun. Uno: Unsupervised occupancy fields for perception and forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14487–14496, 2024. 2

[4] Fabio Arena and Giovanni Pau. An overview of vehicular communications. *Future internet*, 11(2):27, 2019. 7

[5] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 16, 18

[6] Qi Chen, Xu Ma, Sihai Tang, Jingda Guo, Qing Yang, and Song Fu. F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 88–100, 2019. 6, 12, 14

[7] Valter Crescenzi, Giansalvatore Mecca, Paolo Merialdo, et al. Roadrunner: Towards automatic data extraction from large web sites. In *VLDB*, pages 109–118, 2001. 18

[8] Nachiket Deo and Mohan M. Trivedi. Convolutional Social Pooling for Vehicle Trajectory Prediction. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1549–15498, Salt Lake City, UT, USA, 2018. IEEE. 12

[9] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021. 15, 18

[10] 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. 1, 2

[11] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312, 2021. 6

[12] Junru Gu, Chenxu Hu, Tianyuan Zhang, Xuanyao Chen, Yilun Wang, Yue Wang, and Hang Zhao. ViP3D: End-to-End Visual Trajectory Prediction via 3D Agent Queries. 2023. 2, 6

[13] Xu Han, Zonglin Meng, Xin Xia, Xishun Liao, Yueshuai He, Zhaoliang Zheng, Yutong Wang, Hao Xiang, Zewei Zhou, Letian Gao, et al. Foundation intelligence for smart infrastructure services in transportation 5.0. *IEEE Transactions on Intelligent Vehicles*, 2024. 1

[14] Ruiyang Hao, Siqi Fan, Yingru Dai, Zhenlin Zhang, Chenxi Li, Yuntian Wang, Haibao Yu, Wenxian Yang, Yuan Jirui, and Zaiqing Nie. Rcooper: A real-world large-scale dataset for roadside cooperative perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 18

[15] 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. 3

[16] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented Autonomous Driving, 2023. arXiv:2212.10156 [cs]. 1, 2

[17] Xun Huang, Jinlong Wang, Qiming Xia, Siheng Chen, Bisheng Yang, Cheng Wang, and Chenglu Wen. V2x-r: Cooperative lidar-4d radar fusion for 3d object detection with denoising diffusion. *arXiv preprint arXiv:2411.08402*, 2024. 1

[18] Yanjun Huang, Jiatong Du, Ziru Yang, Zewei Zhou, Lin Zhang, and Hong Chen. A Survey on Trajectory-Prediction Methods for Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, pages 1–1, 2022. Conference Name: IEEE Transactions on Intelligent Vehicles. 2

[19] Yangjie Ji, Zewei Zhou, Ziru Yang, Huang Yanjun, Zhang Yuanjian, Zhang Wanting, Xiong Lu, and Zhuoping Yu. Towards autonomous vehicles: a survey on cooperative vehicle-infrastructure system. *Iscience*, 2024. 1, 2

[20] Xiaosong Jia, Penghao Wu, Li Chen, Yu Liu, Hongyang Li, and Junchi Yan. HDGT: Heterogeneous Driving Graph Transformer for Multi-Agent Trajectory Prediction Via Scene Encoding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–15, 2023. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. 2

[21] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023. 2

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 14

[23] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2118–2125, 2018. ISSN: 2153-0017. 2

[24] 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. 4, 12

[25] E Li, Shuaijun Wang, Chengyang Li, Dachuan Li, Xiangbin Wu, and Qi Hao. Sustech points: A portable 3d point cloud

interactive annotation platform system. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1108–1115. IEEE, 2020. 16

[26] Yiming Li, Juexiao Zhang, Dekun Ma, Yue Wang, and Chen Feng. Multi-robot scene completion: Towards task-agnostic collaborative perception. In *6th Annual Conference on Robot Learning*. 1

[27] Yiming Li, Shunli Ren, Pengxiang Wu, Siheng Chen, Chen Feng, and Wenjun Zhang. Learning distilled collaboration graph for multi-agent perception. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021. 6, 12, 14

[28] Yiming Li, Dekun Ma, Ziyan An, Zixun Wang, Yiqi Zhong, Siheng Chen, and Chen Feng. V2X-Sim: Multi-Agent Collaborative Perception Dataset and Benchmark for Autonomous Driving. *IEEE Robotics and Automation Letters*, 7(4):10914–10921, 2022. Conference Name: IEEE Robotics and Automation Letters. 1, 18

[29] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. BEVFormer: Learning Bird's-Eye-View Representation from Multi-camera Images via Spatiotemporal Transformers. In *Computer Vision – ECCV 2022*, pages 1–18, Cham, 2022. Springer Nature Switzerland. 2

[30] Ming Liang, Bin Yang, Wenyuan Zeng, Yun Chen, Rui Hu, Sergio Casas, and Raquel Urtasun. PnPNet: End-to-End Perception and Prediction With Tracking in the Loop. pages 11553–11562, 2020. 2

[31] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 5, 13

[32] 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. 1

[33] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting With a Single Convolutional Net. pages 3569–3577, 2018. 1, 2, 6, 7

[34] Hongzhi Ruan, Haibao Yu, Wenxian Yang, Siqi Fan, Yingjuan Tang, and Zaiqing Nie. Learning Cooperative Trajectory Representations for Motion Forecasting, 2023. arXiv:2311.00371 [cs]. 2

[35] Saeed Saadatnejad, Mohammadhossein Bahari, Pedram Khorsandi, Mohammad Saneian, Seyed-Mohsen Moosavi-Dezfooli, and Alexandre Alahi. Are socially-aware trajectory prediction models really socially-aware? *Transportation Research Part C: Emerging Technologies*, 141:103705, 2022. 6

[36] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision – ECCV 2020*, pages 683–700, Cham, 2020. Springer International Publishing. 6

[37] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. MTR++: Multi-Agent Motion Prediction with Symmetric Scene Modeling and Guided Intention Querying, 2023. arXiv:2306.17770 [cs]. 2, 6

[38] Rui Song, Chenwei Liang, Hu Cao, Zhiran Yan, Walter Zimmer, Markus Gross, Andreas Festag, and Alois Knoll. Collaborative semantic occupancy prediction with hybrid feature fusion in connected automated vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17996–18006, 2024. 1

[39] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. pages 2446–2454, 2020. 2

[40] Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, James Tu, and Raquel Urtasun. V2VNet: Vehicle-to-Vehicle Communication for Joint Perception and Prediction. In *arXiv:2008.07519 [cs]*, 2020. 00010 ECC arXiv: 2008.07519. 2, 6, 12, 14

[41] Zehao Wang, Yuping Wang, Zhuoyuan Wu, Hengbo Ma, Zhaowei Li, Hang Qiu, and Jiachen Li. Cmp: Cooperative motion prediction with multi-agent communication. *arXiv preprint arXiv:2403.17916*, 2024. 2

[42] 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. In *Advances in Neural Information Processing Systems*, 2023. 1, 2, 6, 7, 12, 14

[43] Hao Xiang, Runsheng Xu, and Jiaqi Ma. HM-ViT: Heterogeneous Vehicle-to-Vehicle Cooperative perception with vision transformer, 2023. arXiv:2304.10628 [cs]. 1

[44] Hao Xiang, Zhaoliang Zheng, Xin Xia, Runsheng Xu, Letian Gao, Zewei Zhou, Xu Han, Xinkai Ji, Mingxi Li, Zonglin Meng, et al. V2x-real: a largs-scale dataset for vehicle-to-everything cooperative perception. *arXiv preprint arXiv:2403.16034*, 2024. 2, 5, 6, 18

[45] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2X-ViT: Vehicle-to-Everything Cooperative Perception with Vision Transformer. In *ECCV 2022*, pages 107–124, Cham, 2022. Springer Nature Switzerland. 2, 6, 7, 8, 12, 14, 18

[46] 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, 2022. 1, 2, 18

[47] Runsheng Xu, Xin Xia, Jinlong Li, Hanzhao Li, Shuo Zhang, Zhengzhong Tu, Zonglin Meng, Hao Xiang, Xiaoyu Dong, Rui Song, Hongkai Yu, Bolei Zhou, and Jiaqi Ma. V2V4Real: A Real-World Large-Scale Dataset for Vehicle-to-Vehicle Cooperative Perception. pages 13712–13722, 2023. 2, 18

[48] Kun Yang, Dingkang Yang, Jingyu Zhang, Mingcheng Li, Yang Liu, Jing Liu, Hanqi Wang, Peng Sun, and Liang Song. Spatio-Temporal Domain Awareness for Multi-Agent Collaborative Perception. pages 23383–2339. arXiv, 2023. 1, 2

[49] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, and Zaiqing Nie. DAIR-V2X: A Large-Scale Dataset for Vehicle-Infrastructure Cooperative 3D Object Detection. pages 21361–21370, 2022. 2, 3, 18

[50] Haibao Yu, Wenxian Yang, Hongzhi Ruan, Zhenwei Yang, Yingjuan Tang, Xu Gao, Xin Hao, Yifeng Shi, Yifeng Pan, Ning Sun, Juan Song, Jirui Yuan, Ping Luo, and Zaiqing Nie. V2X-Seq: A Large-Scale Sequential Dataset for Vehicle-Infrastructure Cooperative Perception and Forecasting. arXiv, 2023. arXiv:2305.05938 [cs]. 2, 18

[51] 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, 6, 7, 12, 14

[52] Haibao Yu, Wenxian Yang, aru Zhong, Zhenwei Yang, Siqi Fan, Ping Luo, and Zaiqing Nie. End-to-end autonomous driving through v2x cooperation. *arXiv preprint arXiv:2404.00717*, 2024. 2

[53] Xinyu Zhang, Zewei Zhou, Zhaoyi Wang, Yangjie Ji, Yanjun Huang, and Hong Chen. Co-mtp: A cooperative trajectory prediction framework with multi-temporal fusion for autonomous driving. *arXiv preprint arXiv:2502.16589*, 2025. 2

[54] Seth Z. Zhao, Hao Xiang, Chenfeng Xu, Xin Xia, Bolei Zhou, and Jiaqi Ma. Coopre: Cooperative pretraining for v2x cooperative perception, 2024. 2

[55] Zhaoliang Zheng, Xin Xia, Letian Gao, Hao Xiang, and Jiaqi Ma. Cooperfuse: A real-time cooperative perception fusion framework. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 533–538, 2024. 6

[56] Zewei Zhou, Ziru Yang, Yuanjian Zhang, Yanjun Huang, Hong Chen, and Zhuoping Yu. A comprehensive study of speed prediction in transportation system: From vehicle to traffic. *iScience*, 25(3):103909, 2022. 2

[57] Walter Zimmer, Gerhard Arya Wardana, Suren Sritharan, Xingcheng Zhou, Rui Song, and Alois C Knoll. Tumtraf v2x cooperative perception dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22668–22677, 2024. 1

# V2XPnP: Vehicle-to-Everything Spatio-Temporal Fusion for Multi-Agent Perception and Prediction

Supplementary Material

## Contents



## A. Implementation Details

In this section, we provide detailed configurations for cooperative perception and prediction tasks, including the baseline models used in our experiments and the proposed V2XPnP framework.

### A.1. Benchmark Model Details

**PointPillar Backbone.** For all experiments, we employ the anchor-based PointPillar model [24] as the LiDAR Feature Extraction backbone. The voxel resolution is set to 0.4 meters in both the $x$ and $y$ directions, with a maximum of 32 points per voxel and a total of 32,000 voxels. Additionally, we set the number of anchors per grid cell to 2.

**Intermediate Fusion Methods.** We implement several state-of-the-art single-frame intermediate fusion methods, including *V2VNet* [40], *F-Cooper* [6], *DiscoNet* [27],*CoBEVFlow* [42], *FFNet* [51], *V2X-ViT* [45], and our proposed V2XPnP model, integrating them with our end-to-end model to replace the spatio-temporal fusion module. The model settings and configurations for the fusion module adhere to the original implementations.

**Map Feature Extraction.** HD maps are represented as sets of polylines, with each polyline comprising 10 points. Because the map is projected onto the BEV space, each grid only contains the five nearest polylines. Each waypoint in a polyline contains seven attributes: $(x, y, d_x, d_y, type, x_{pre}, y_{pre})$, representing position, direction, lane type, and previous position. These attributes are encoded using MLP layers into a 256 hidden dimension feature, followed by 1 or 2 Transformer layers with two attention heads to model interactions among map elements.

**Decoupled Attention Predictor.** For the decoupled perception and prediction pipeline, we implement an attention-based predictor for trajectory-level prediction tasks. This predictor utilizes a 1D Convolution + LSTM Network [8] to encode temporal historical trajectories and a Transformer layer to capture the interaction among objects and the map, then an LSTM-based decoder generates the future predicted trajectories. All trajectory data, including historical and predicted trajectories, are represented in the local coordinate frame of each object.

### A.2. V2XPnP Model Details

**Temporal Attention.** To capture the temporal dependence, we initialize the historical timestamp sequence using Sinusoidal positional encodings conditioned on time and further process these encodings through a Linear layer. The temporal attention block in the multi-frame temporal fusion module has four attention heads. To enhance the inter-frame feature representation, we stack three temporal fusion modules with the temporal attention block.

**Self-spatial Attention.** This block is applied following either the temporal attention or the multi-agent spatial attention. In self-spatial attention, the feature map is partitioned into patches using common window sizes of $(2, 4, 8)$. Given the complexity of spatio-temporal fusion across multiple agents, the self-spatial attention module employs a higher number of attention heads $(16, 8, 4)$ after multi-agent spatial fusion, compared to the heads $(8, 4, 2)$ used following temporal attention.

**Multi-agent Spatial Attention.** Our dataset categorizes agents as infrastructure agents, denoted by negative labels (*i.e.*, $-1$ and $-2$), or connected automated vehicles (CAV)

agents, denoted by positive labels (*i.e.*, 1 and 2). To capture the heterogeneous dependencies among these agents, we construct a heterogeneous graph and employ distinct attention fusion parameters for each agent type. The multi-agent spatial attention utilizes eight attention heads, and we stack three multi-agent spatial fusion modules with the multi-agent spatial attention to capture the inter-agent relationships.

## A.3. Loss Function

This section provides the loss function employed in our multi-task model. The initial weights of regression loss $\mathcal{L}_{\text{reg}}$, classification loss $\mathcal{L}_{\text{cla}}$ and prediction loss $\mathcal{L}_{\text{pred}}$ are set as $w_{\text{reg}}, w_{\text{cla}}, w_{\text{pred}} = 2.0, 1.0, 2.0$. For single-task learning, the same loss function is used but weights exclusively on the components relevant to that task.

**Perception Loss.** The perception task loss combines classification and regression components, designed to align predicted anchor boxes with ground truth labels. For classification, which involves identifying objects and background elements, we employ Focal Loss [31] to address the imbalance between foreground and background samples. The Focal Loss is expressed as:

$$\mathcal{L}_{\text{cla}} = -\alpha(1-p_t)^{\gamma}\log(p_t), \tag{S1}$$

where $p_t$ is the predicted probability for the target anchor box, and $\alpha$ and $\gamma$ are balancing and focusing factors. Anchor-wise weights are applied to further enhance the balance between positive and negative samples.

For the regression component, we employ Smooth $\ell_1$-Loss to optimize the predicted bounding boxes to match the ground truth labels in terms of position and orientation, and a sine-cosine encoding is employed to handle rotational ambiguities. The Smooth $\ell_1$-Loss is defined as:

$$\mathcal{L}_{\text{reg}} = \begin{cases} 0.5 \cdot \frac{\Delta^2}{\beta}, & \text{if} |\Delta| < \beta, \\ |\Delta| - 0.5 \cdot \beta, & \text{otherwise}, \end{cases} \tag{S2}$$

where $\Delta = \text{prediction} - \text{target}$, and $\beta$ is a hyper-parameter controlling the transition between $\ell_1$ and $\ell_2$ loss.

**Prediction Loss.** We adopt the $\ell_2$-loss function to minimize the discrepancy between the predicted trajectory and the ground truth.

$$\mathcal{L}_{\text{pred}} = \frac{1}{N_{\text{det}}}\frac{1}{T_{\text{valid}}}\sum_{i=1}^{N_{\text{det}}}\sum_{t=1}^{T_{\text{valid}}}\|\boldsymbol{\mu}_t^i - \mathbf{x}_t^i\|^2, \tag{S3}$$

where $\boldsymbol{\mu}_t^i$ and $\mathbf{x}_t^i$ represent the predicted position and target position of the $i$-th object at time step $t$. $T_{\text{valid}}$ is the number of valid future time steps for the agent, and $N_{\text{det}}$ is the number of detected objects.

## A.4. Training Strategy

The end-to-end cooperative perception and prediction model addresses two distinct yet interrelated tasks while integrating information across both temporal and spatial dimensions. Training such an end-to-end model from scratch often results in suboptimal performance, due to the inherent complexity of jointly optimizing these tasks and dimensions. To effectively handle these challenges, we adopt a multi-stage training strategy to progressively refine the model's capabilities.

**Multi-Stage Training Strategy.** Initially, the end-to-end perception and prediction model is trained in a single-agent setting, focusing on temporal fusion without incorporating multi-agent spatial fusion. It simplifies the optimization process, enabling the model to learn robust temporal features in isolation. The resulting single-agent model then serves as a pre-trained model for subsequent multi-agent spatial fusion training in the V2X environment. This staged training strategy ensures that the model incrementally acquires the ability to handle the complexities of cooperative perception and prediction tasks.

**Stage 1: Single-Agent Multi-task Learning.** The single-agent model training stage addresses the core challenge of coordinating multi-task learning to capture complex patterns across perception and prediction tasks. Prediction task requires a comprehensive understanding of objects' temporal information and their intricate motion patterns, while detection focuses mainly on identifying objects in the current frame, with historical information providing supplementary context. Training both tasks jointly without proper initialization risks overfitting to simpler current-frame features, thereby neglecting the rich but complex temporal features essential for accurate prediction. Moreover, perception is foundational to prediction, as detecting an object is a prerequisite for predicting its motion. To effectively balance the two tasks, we adopt a task-specific training strategy. *(1) Single-Frame Perception Training:* the training begins by optimizing the model for single-frame perception, establishing a foundation for object detection. *(2) Temporal Prediction Training:* the prediction task is introduced by freezing the parameters of the detection backbone and training an additional temporal network and prediction head, guiding the model to focus more on the prediction task and effectively learn complex temporal dependencies. *(3) Joint Fine-Tuning:* the entire model is unfrozen, enabling end-to-end fine-tuning across both tasks.

**Stage 2: Multi-Agent Spatiotemporal Learning.** Based on the pre-trained single-agent model, the multi-agent fusion module is introduced and jointly trained with the entire model. At this stage, the primary focus is to balance the two tasks, ensuring that neither perception nor prediction dominates the training process. To achieve this, we employ a dynamic loss-weighting strategy that gradually increases the weight assigned to the prediction loss. This approach ensures balanced optimization, avoiding performance trade-offs between tasks and improving overall effectiveness across both perception and prediction objectives.

Table S1. Additional benchmark results of cooperative perception and prediction models on V2XPnP Sequential (V2XPnP-Seq) Dataset

| Dataset | Method | E2E | Map | **AP@0.5** (%) ↑ | ADE (m) ↓ | FDE (m) ↓ | MR (%) ↓ | **EPA** (%) ↑ |
|---|---|---|---|---|---|---|---|---|
| V2XPnP-Seq-VC *(with V+2I at most)* | V2VNet* [40] | ✓ | | 48.6 | 2.10 | 3.75 | 42.3 | 25.3 |
| | F-Cooper* [6] | ✓ | ✓ | 66.0 | 1.35 | 2.56 | 36.1 | 38.7 |
| | DiscoNet* [27] | ✓ | ✓ | 66.8 | 1.41 | 2.62 | 34.4 | 42.8 |
| | V2XPnP (Ours) | ✓ | ✓ | **71.6** | 1.35 | 2.36 | 31.7 | **48.2** |
| V2XPnP-Seq-IC *(with 2V+1 at most)* | V2VNet* [40] | ✓ | | 33.6 | 1.95 | 3.53 | 44.2 | 16.3 |
| | F-Cooper* [6] | ✓ | ✓ | 60.2 | 1.21 | 2.32 | 36.3 | 36.3 |
| | DiscoNet* [27] | ✓ | ✓ | 65.4 | 1.14 | 2.18 | 36.1 | 40.7 |
| | V2XPnP (Ours) | ✓ | ✓ | **71.0** | 1.18 | 2.16 | 34.0 | **46.0** |
| V2XPnP-Seq-V2V | V2VNet* [40] | ✓ | | 43.1 | 3.10 | 5.55 | 46.8 | 19.4 |
| | F-Cooper* [6] | ✓ | ✓ | 60.2 | 1.69 | 3.22 | 41.1 | 34.4 |
| | DiscoNet* [27] | ✓ | ✓ | 61.2 | 1.66 | 3.13 | 41.2 | 33.1 |
| | V2XPnP (Ours) | ✓ | ✓ | **70.5** | 1.78 | 3.28 | 39.9 | **40.6** |
| V2XPnP-Seq-I2I | V2VNet* [40] | ✓ | | 41.1 | 1.83 | 3.34 | 40.4 | 23.2 |
| | F-Cooper* [6] | ✓ | ✓ | 58.6 | 1.34 | 2.58 | 40.0 | 33.6 |
| | DiscoNet* [27] | ✓ | ✓ | 63.5 | 1.15 | 2.19 | 37.5 | 38.4 |
| | V2XPnP (Ours) | ✓ | ✓ | **69.2** | 1.26 | 2.31 | 36.5 | **42.8** |

**Training Details**. The model is trained using the Adam optimizer [22] with an initial learning rate of $2 \times 10^{-3}$ and a weight decay of $1 \times 10^{-4}$ with early stopping on NVIDIA L40S GPUs. We employ 4 training stages, as detailed before, and each training stage consists of 30 epochs with a batch size of 2. Early stopping is employed to prevent overfitting. We carefully tune the hyperparameters to ensure the stability and efficiency of the training process.

## B. Additional Benchmark Results

In this paper, we benchmark different spatiotemporal strategies with 11 fusion models in total:

- **No Fusion**: *No Fusion*, *No Fusion-FaF*
- **Early Fusion**: *Early Fusion*
- **Late Fusion**: *Late Fusion*
- **Intermediate Fusion**: *V2VNet* [40], *F-Cooper* [6], *DiscoNet* [27], *CoBEVFlow* [42], *FFNet* [51], *V2X-ViT* [45], and our proposed *V2XPnP*.

We present additional benchmark results for *V2VNet* [40], *F-Cooper* [6], and *DiscoNet* [27] across all collaboration modes, as shown in Tab. S1. Our proposed *V2XPnP* consistently outperforms these SOAT baselines in terms of EPA and AP across all collaboration modes. Notably, V2VNet* exhibits lower performance, likely due to the absence of a map and the loss of temporal features during explicit feature ROI matching.

## C. Cooperative Temporal Perception Task

In addition to the end-to-end perception and prediction task, the sequential nature of our V2XPnP-Sequential dataset facilitates other temporal tasks, including temporal perception and traditional prediction tasks. In this section, we introduce the cooperative temporal perception task and present benchmark results on the V2XPnP-Sequential dataset. Details on the traditional prediction task are provided in Sec. D.

### C.1. Problem Formulation

The cooperative temporal perception task is an extension of the single-frame perception task by incorporating historical context. Specifically, given historical $T$ frames raw perception data $\mathbf{P}_i^t, i \in \{1, \cdots, N\}$ from all $N$ agents within the communication range of the ego agent, the objective is to detect the surrounding objects in the current frame. The core challenge lies in effectively leveraging temporal information from $T$ past frames to enhance detection accuracy in the present frame.

### C.2. Benchmark Methods

For benchmarking, we adapt our end-to-end model, *V2XPnP*, by removing the prediction head, resulting in a model only for temporal perception. Various V2X fusion strategies are evaluated in this framework, as detailed in Tab. S2. Moreover, we provide another baseline *FaF*$^*$, which adopts a combination of 2D and 3D convolutions for temporal fusion. *FaF*$^*$ further integrates with the F-Cooper intermediate fusion method and early fusion method for V2X fusion comparison. We also provide the results of the *No Temp* model, which excludes temporal fusion and is evaluated using both F-Cooper and early fusion methods. Model parameters and experimental setups for this task are consistent with those used for the end-to-end cooperative perception and prediction task.

Table S2. Benchmark results for cooperative temporal perception. No Temp: single-frame perception, FaF*: temporal perception with alternating 2D and 3D convolutions, V2XPnP: temporal perception with temporal attention modules.

| Dataset | No Fusion (AP@0.5 (%) ↑) | | | Early Fusion (AP@0.5 (%) ↑) | | | Intermediate Fusion (AP@0.5 (%) ↑) | | |
|---|---|---|---|---|---|---|---|---|---|
| | No Temp | FaF* | V2XPnP | No Temp | FaF* | V2XPnP | No Temp | FaF* | V2XPnP |
| V2XPnP-Seq-VC | 43.9 | 57.1 | **60.3** | 63.5 | 67.0 | **71.0** | 65.1 | 70.3 | **74.0** |
| V2XPnP-Seq-IC | 46.4 | 61.1 | **64.7** | 61.0 | 65.5 | **71.4** | 61.1 | 67.1 | **73.2** |
| V2XPnP-Seq-V2V | 40.8 | 53.7 | **59.1** | 54.9 | 56.4 | **66.6** | 58.0 | 61.4 | **69.4** |
| V2XPnP-Seq-I2I | 51.0 | 61.2 | **64.7** | 63.4 | 66.0 | **71.6** | 58.5 | 62.9 | **72.4** |

### C.3. Benchmark Results

The results demonstrate that incorporating temporal cues significantly improves perception performance across all multi-agent fusion strategies. Notably, our *V2XPnP* model achieves superior results compared to other baselines, due to the careful design of temporal attention. However, we observe a slight performance drop when the same model is applied to the end-to-end cooperative perception and prediction task, compared to its use solely for temporal perception. The possible reason is the difficulty of optimizing both tasks to achieve optimal performance. Nevertheless, the end-to-end model still outperforms other baselines in both perception and prediction tasks. Future research should focus on optimizing the balance between multiple tasks to further enhance the performance of end-to-end models.

## D. Traditional Cooperative Prediction Task

### D.1. Problem Formulation

V2XPnP sequential dataset also supports the traditional prediction task. Compared to end-to-end models, which directly infer future states of objects from perception data, the traditional prediction task forecasts their future trajectories from historical trajectories. The cooperative prediction task is formulated as: given the map and the historical trajectories of all detected objects obtained from the ego agent and other agents (*e.g.*, CAVs and infrastructure units) within the communication range of the ego agent, the objective is to predict future trajectories of these detected objects.

### D.2. Benchmark Methods

To investigate the influence of perception results on prediction tasks, we provide two types of input for the prediction models: 1) Ground-truth historical trajectories of surrounding objects; 2) Perception-based historical trajectories generated by the upstream perception module. The first one is the common setting for the traditional trajectory prediction task, assuming full availability of accurate historical trajectories for prediction. However, it ignores real-world challenges such as occlusions and cumulative errors introduced by separate modules. To address this limitation and enable a more realistic evaluation, we designed the second setting, where CAVs can only derive the historical trajectories from the perception results and thus the perception uncertainty can propagate to the downstream prediction. Notably, regardless of the input type, the prediction model is trained using the complete future trajectory dataset aggregated from all agents.

In our experiment, the prediction model configuration and experimental setup align closely with the decoupled attention predictor. Following the LSTM baseline setting in the Waymo motion dataset [9], the LSTM model also serves as a strong baseline, which includes an LSTM encoder and LSTM decoder. We report benchmark results under three configurations: *No Fusion*, where no perception information is fused; *Ground Truth*, assuming perfect historical trajectories; *Late Fusion*, where the decoupled pipeline from the traditional prediction task is employed.

### D.3. Benchmark Results

The experimental results, summarized in Tab. S3, compare traditional prediction under three input settings: ground truth trajectories, perception without fusion, and perception with late fusion. The results indicate that as perception improves—from no fusion to late fusion—the prediction performance correspondingly increases. When the environment is fully observable, the task simplifies to the traditional prediction setup, achieving the best overall performance for both detection and prediction. A significant drop in performance is observed for perception-based prediction, highlighting the critical dependency of predictive tasks on perception accuracy. Moreover, the Attention predictor shows better robustness compared to the LSTM baseline under noisy perception inputs, thanks to the attention module for complex interaction feature capturing. We anticipate that this temporal prediction task will inspire further exploration of perception-based prediction approaches.

Table S3. Benchmark results for traditional prediction. No Fusion: prediction based on the no-fusion perception results. Late Fusion: prediction based on the late fusion perception results. Ground Truth: prediction based on the ground truth trajectories with no occlusion or perception errors.

| Dataset | Method | Attention Predictor | | | | LSTM Predictor | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | AP@0.5(%) ↑ | ADE(m) ↓ | FDE(m) ↓ | MR(%) ↓ | AP@0.5(%) ↑ | ADE(m) ↓ | FDE(m) ↓ | MR(%) ↓ |
| V2XPnP-Seq -VC | No Fusion | 43.9 | 1.87 | 3.24 | 33.8 | 43.9 | 2.91 | 4.77 | 35.0 |
| | Late Fusion | 58.1 | 1.59 | 2.81 | 34.3 | 58.1 | 2.76 | 4.60 | 33.7 |
| | Ground Truth | - | 0.60 | 1.26 | 23.0 | - | 0.66 | 1.31 | 23.0 |
| V2XPnP-Seq -IC | No Fusion | 46.4 | 2.10 | 3.75 | 42.3 | 46.4 | 2.11 | 3.67 | 35.8 |
| | Late Fusion | 55.9 | 1.39 | 2.44 | 30.1 | 55.9 | 2.61 | 4.40 | 32.7 |
| | Ground Truth | - | 0.63 | 1.35 | 26.2 | - | 0.61 | 1.31 | 25.0 |
| V2XPnP-Seq -V2V | No Fusion | 40.8 | 1.99 | 3.38 | 34.0 | 40.8 | 2.98 | 4.82 | 34.4 |
| | Late Fusion | 55.3 | 1.75 | 3.07 | 34.0 | 55.3 | 2.87 | 4.79 | 35.0 |
| | Ground Truth | - | 0.60 | 1.26 | 22.9 | - | 0.66 | 1.31 | 22.8 |
| V2XPnP-Seq -I2I | No Fusion | 51.0 | 1.69 | 3.06 | 36.2 | 51.0 | 2.11 | 3.67 | 35.9 |
| | Late Fusion | 61.3 | 1.41 | 2.50 | 30.0 | 61.3 | 2.44 | 4.18 | 32.1 |
| | Ground Truth | - | 0.63 | 1.35 | 26.2 | - | 0.61 | 1.31 | 25.0 |

# E. V2XPnP Sequential Dataset Details

## E.1. Dataset Visualization

Our V2XPnP-Sequential dataset provides two sensor sequences (LiDAR and camera) collected in dense urban environments, capturing diverse interactive behaviors over time. Fig. S1 illustrates two representative interaction scenarios in our dataset, presenting LiDAR and camera data from multiple agents at two key timestamps. The main intersection objects pair have been annotated with red and yellow blocks in different agents' views.

## E.2. Data Acquisition

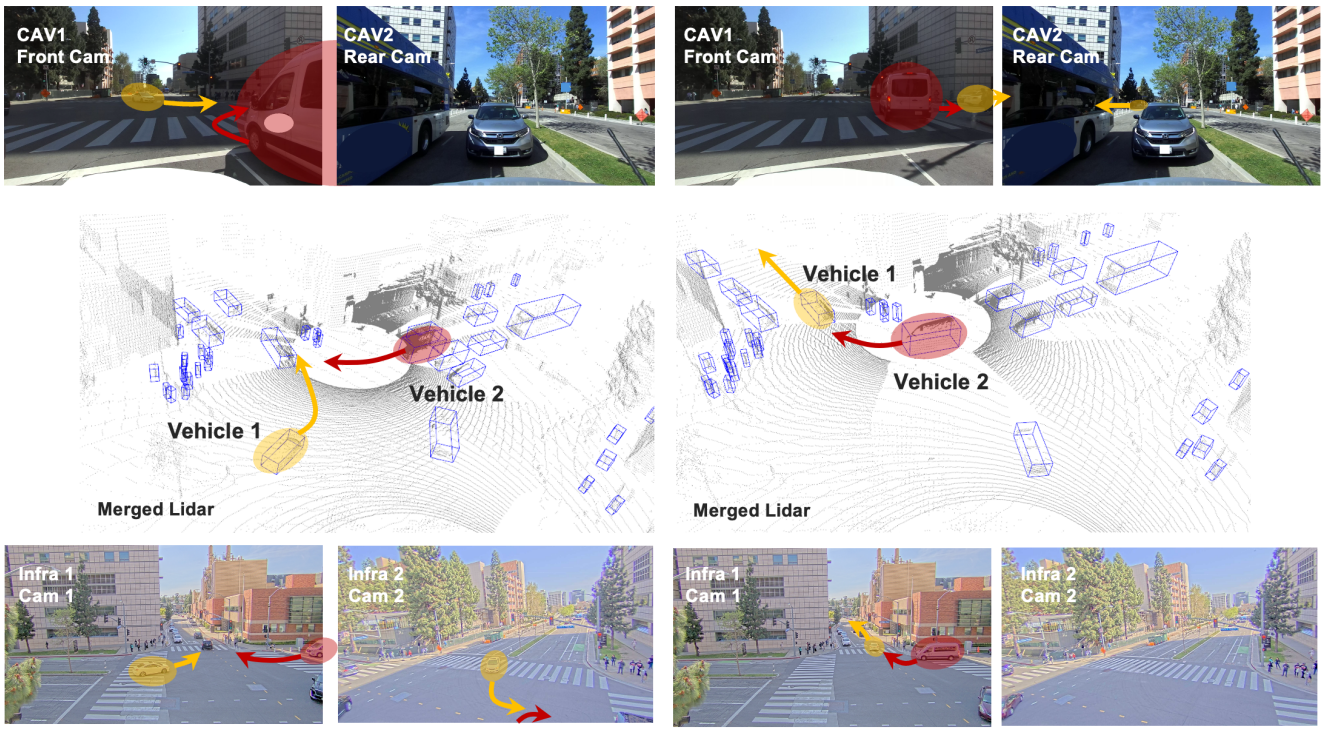
**Sensor Specifications.** The dataset was collected using four agents - two connected automated vehicles and two smart infrastructure units. Each CAV is equipped with a RoboSense 128-beam LiDAR, four stereo RGB cameras with 1920 × 1080 resolution, and an integrated GPS/IMU system. The four stereo cameras are mounted on the front, rear, left, and right sides of the CAV, providing a complete 360-degree field of view. Similarly, each infrastructure unit is configured with a 128- or 64-beam LiDAR, two Axis cameras with 1920 × 1080 resolution, and a GPS module. The sensor deployment of our data collection system is shown in Fig. 4(a).

**Coordinate System.** Our V2XPnP-Sequential dataset encompasses three coordinate systems: the LiDAR coordinate system, the camera coordinate system, and the map coordinate system. Each agent - vehicle or infrastructure - maintains its own local LiDAR and camera coordinate systems. The global map coordinate system serves as the reference for all annotations and maps. The transformation from each agent's local LiDAR coordinate to the map coordinate in each frame is achieved with the GPS/IMU data and the offline PCD map. We also conduct the 3D-2D calibration for LiDAR and camera, as shown in Fig. 4(b).

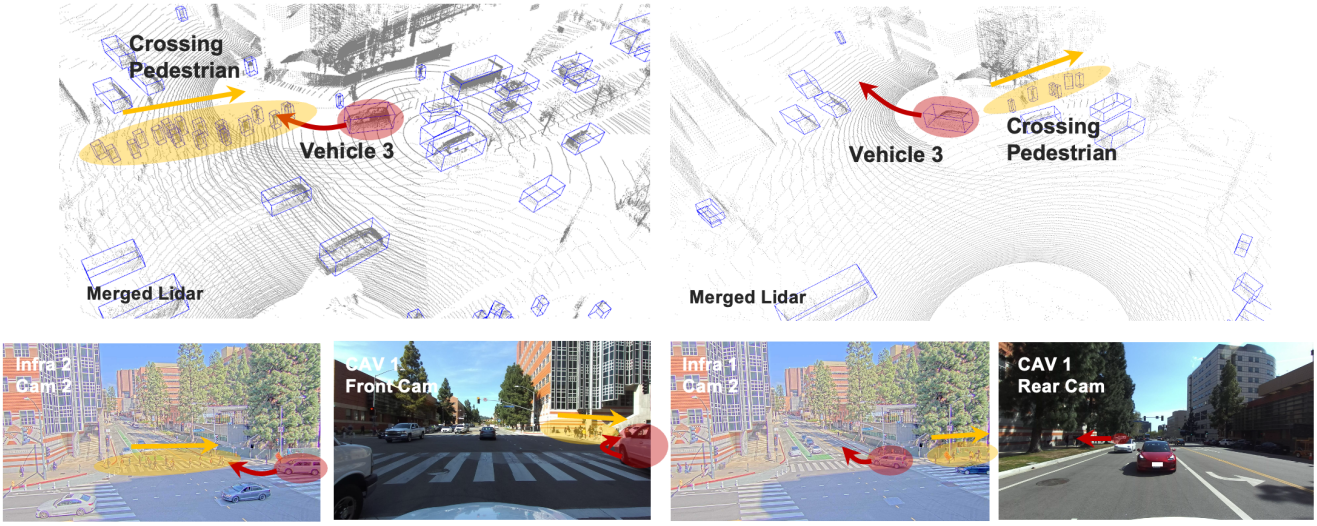
## E.3. Data Annotation and Processing

**Data Annotation.** The 3D bounding boxes in our V2XPnP-Sequential dataset are annotated using an open-source labeling tool, SUSTechPOINTS [25], by expert annotators. The first step is annotating the bounding boxes in the point clouds from the two CAVs and infrastructure units. Then, these bounding boxes, annotated in different agents' coordinate frames, are processed through a V2X sequential pipeline to assign consistent object IDs across agents and temporal frames. To ensure annotation quality, each object is subjected to eight rounds of review and revision. In total, ten object categories are included in our dataset: car, pedestrian, scooter, motorcycle, bicycle, truck, van, concrete truck, bus, and road barrier. Each object annotation includes the center of the bounding box $(x, y, z)$, sizes $(width, length, height)$, and orientation $(roll, yaw, pitch)$ in the global coordinates. Notably, we follow a general object definition in annotation, encompassing stationary objects such as parked vehicles and barriers, which are annotated similarly to movable objects but explicitly labeled as static. This aligns with public datasets like nuScenes [5], where static objects are tracked while maintaining consistent IDs.

**Trajectory Generation.** In addition to perception data, the dataset provides a ground-truth trajectory dataset derived from the fused perception data of all agents, capturing the trajectories of objects across all frames. This trajectory dataset is primarily utilized in traditional prediction tasks, which assume all history trajectories are observable to the ego agent. However, this assumption ignores the fact that the trajectories obtained from onboard sensors are incomplete due to occlu-




(a) Scene 1: A vehicle slows to wait for a turning flow in its opposite lane.




(b) Scene 2: A vehicle waits for a crossing pedestrian flow and then continue to turn.

Figure S1. Examples of interaction scenarios from the V2XPnP-Sequential dataset. The dataset multi-agent perception perspectives and captures diverse interaction behaviors among ten object classes in dense urban traffic environments.

Table S4. Comparison between the V2XPnP-Sequential dataset and other public available driving datasets

| Dataset | Year | Type | V2V | V2I | I2I | Trajectory | Map | Agent Number | Tracked Objects/Scene | 3D Boxes | RGB Images | LiDAR Frames | Categories |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| nuScenes [5] | 2019 | Real | | | | ✓ | ✓ | 1 | 75.75 | 1.4M | 1.4M | 400k | 23 |
| Waymo Open [9] | 2019 | Real | | | | ✓ | ✓ | 1 | - | 12M | 1M | 200k | 4 |
| OPV2V [46] | 2022 | Sim | ✓ | | | | | 2.89 | 26.5 | 230k | 44k | 11k | 1 |
| V2X-Sim [28] | 2022 | Sim | ✓ | ✓ | | ✓ | | 10 | - | 26.6k | 0 | 10k | 1 |
| V2XSet [45] | 2022 | Sim | ✓ | ✓ | | ✓ | | 2-7 | - | 230k | 44k | 11k | 1 |
| DAIR-V2X [49] | 2022 | Real | | ✓ | | | | 2 | 0 | 464k | 39k | 39k | 10 |
| V2V4Real [47] | 2023 | Real | ✓ | | | ✓ | ✓ | 2 | - | 240k | 40k | 20k | 5 |
| V2X-Seq [50] | 2023 | Real | | ✓ | | ✓ | ✓ | 2 | 110 | 464k | 71k | - | 10 |
| RCooper [14] | 2024 | Real | | | ✓ | ✓ | | 4 | - | - | 50k | 30k | 10 |
| V2X-Real [44] | 2024 | Real | ✓ | ✓ | ✓ | | | 4 | 0 | 1.2M | 171K | 33k | 10 |
| **V2XPnP-Seq** | **2024** | **Real** | ✓ | ✓ | ✓ | ✓ | ✓ | **4** | **136** | **1.45M** | **208k** | **40K** | **10** |

sion and limited perception range, and no specific datasets are designed to support this task. To support research in prediction with real-world sensor constraints, we provide a trajectory retrieve module in the V2XPnP-Sequential dataset to return observable trajectories of surrounding objects based on their actual visibility relationships.

**Map Generation.** The HD map generation involves two stages: point cloud (PCD) map generation and vector map generation. (1) To generate the PCD map, each LiDAR frame from the CAVs is preprocessed to remove dynamic objects, retaining only static elements essential for mapping. Then, a Normal Distributions Transform (NDT) scan-matching algorithm is employed to compute the relative transformation between consecutive frames, forming the basis of the LiDAR odometry. We also incorporate translation and heading information obtained from the vehicle's GPS/IMU system, integrating them through a Kalman filter to refine the pose estimation, mitigating the drift from the error accumulation in LiDAR data. Finally, the LiDAR sequences are fused to form the PCD map across all collection areas. (2) The aggregated PCD map is imported into RoadRunner [7] to generate vector maps. Road geometry is inferred and annotated based on intensity variations visualized by distinct color mappings within RoadRunner, and all the semantic attribution is annotated based on the collected camera data, such as road type (*e.g.*, driving, sidewalk, and parking) and line type (*e.g.*, solid and broken yellow line combination and solid white line). Finally, the generated maps are exported in the OpenDRIVE (Xodr) format and converted to Waymo map format [9], ensuring compatibility with downstream applications.

## E.4. Dataset Analysis

Tab. S4 presents the comparison of the V2XPnP-Sequential dataset with existing driving datasets. Our dataset tracks an average of 136 objects per scene, recording high-density and complex traffic scenarios. Furthermore, the dataset's extensive map and trajectory data further enhance its utility in cooperative perception and prediction research across all collaboration modes. The data distribution of ten object classes is shown in Fig. 4(d). The dataset covers 24 intersections of varying types, including roundabouts, T-junctions, and crossroads, as shown in Fig. S2. Notably, many collection areas have a significant gradient, which can facilitate the detection and prediction research in diverse terrain conditions.

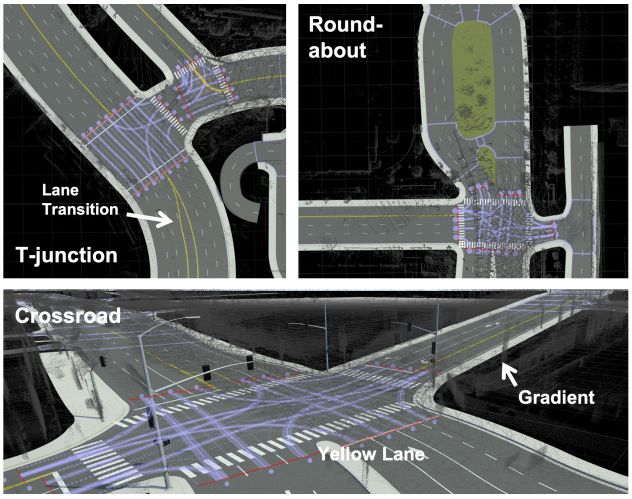



Figure S2. Examples of intersection types in the map, including T-junctions, roundabouts, and crossroads. The gray point clouds in the background represent the PCD map, while lane transitions and gradients are depicted in the map.

## E.5. Dataset Privacy Protection

The V2XPnP-Sequential dataset is designed with stringent privacy safeguards to ensure the anonymity of individuals and vehicles. Trajectory data only include object IDs and

positions, eliminating the possibility of tracking specific entities. All perception data undergoes privacy-preserving processing, with LiDAR annotations retaining only essential attributes such as object ID, agent type, and bounding box pose. Additionally, all image data has been anonymized, with human faces and other potentially sensitive details obscured or removed.