

# V2X-ViT: Vehicle-to-Everything Cooperative Perception with Vision Transformer

Runsheng Xu<sup>1\*</sup>, Hao Xiang<sup>1\*</sup>, Zhengzhong Tu<sup>2\*</sup>, Xin Xia<sup>1</sup>,  
Ming-Hsuan Yang<sup>3,4</sup>, and Jiaqi Ma<sup>1†</sup>

<sup>1</sup> University of California, Los Angeles

<sup>2</sup> University of Texas at Austin

<sup>3</sup> Google Research

<sup>4</sup> University of California, Merced

**Abstract.** In this paper, we investigate the application of Vehicle-to-Everything (V2X) communication to improve the perception performance of autonomous vehicles. We present a robust cooperative perception framework with V2X communication using a novel vision Transformer. Specifically, we build a holistic attention model, namely V2X-ViT, to effectively fuse information across on-road agents (i.e., vehicles and infrastructure). V2X-ViT consists of alternating layers of heterogeneous multi-agent self-attention and multi-scale window self-attention, which captures inter-agent interaction and per-agent spatial relationships. These key modules are designed in a unified Transformer architecture to handle common V2X challenges, including asynchronous information sharing, pose errors, and heterogeneity of V2X components. To validate our approach, we create a large-scale V2X perception dataset using CARLA and OpenCDA. Extensive experimental results demonstrate that V2X-ViT sets new state-of-the-art performance for 3D object detection and achieves robust performance even under harsh, noisy environments. The code is available at <https://github.com/DerrickXuNu/v2x-vit>.

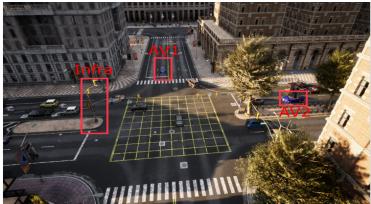
**Keywords:** V2X, Vehicle-to-Everything, Cooperative perception, Autonomous driving, Transformer

## 1 Introduction

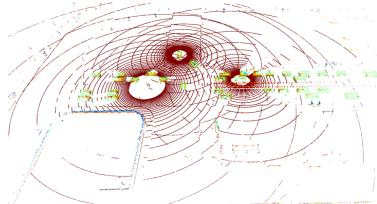
Perceiving the complex driving environment precisely is crucial to the safety of autonomous vehicles (AVs). With recent advancements of deep learning, the robustness of single-vehicle perception systems has demonstrated significant improvement in several tasks such as semantic segmentation [37,11], depth estimation [61,12], and object detection and tracking [22,27,55,14]. Despite recent advancements, challenges remain. Single-agent perception system tends to suffer from occlusion and sparse sensor observation at a far distance, which can potentially cause catastrophic consequences [57]. The cause of such a problem is that an individual vehicle can only perceive the environment from a single

---

\* Equal contribution. † Corresponding author: [jiaqima@ucla.edu](mailto:jiaqima@ucla.edu)



(a) Snapshot of Simulation



(b) Aggregated LiDAR point cloud

**Fig. 1: A data sample from the proposed V2XSet.** (a) A simulated scenario in CARLA where two AVs and infrastructure are located at different sides of a busy intersection. (b) The aggregated LiDAR point clouds of these three agents.

perspective with limited sight-of-view. To address these issues, recent studies [45, 6, 50, 5, 54, 23] leverage the advantages of multiple viewpoints of the same scene by investigating Vehicle-to-Vehicle (V2V) collaboration, where visual information (*e.g.*, detection outputs, raw sensory information, intermediate deep learning features, details see Sec. 2) from multiple nearby AVs are shared for a complete and accurate understanding of the environment.

Although V2V technologies have the prospect to revolutionize the mobility industry, it ignores a critical collaborator – roadside infrastructure. The presence of AVs is usually unpredictable, whereas the infrastructure can always provide supports once installed in key scenes such as intersections and crosswalks. Moreover, infrastructure equipped with sensors on an elevated position has a broader sight-of-view and potentially less occlusion. Despite these advantages, including infrastructure to deploy a robust V2X perception system is non-trivial. Unlike V2V collaboration, where all agents are homogeneous, V2X systems often involve a heterogeneous graph formed by infrastructure and AVs. The configuration discrepancies between infrastructure and vehicle sensors, such as types, noise levels, installation height, and even sensor attributes and modality, make the design of a V2X perception system challenging. Moreover, the GPS localization noises and the asynchronous sensor measurements of AVs and infrastructure can introduce inaccurate coordinate transformation and lagged sensing information. Failing to properly handle these challenges will make the system vulnerable.

In this paper, we introduce a unified fusion framework, namely V2X Vision Transformer or **V2X-ViT**, for V2X perception, that can jointly handle these challenges. Fig. 2 illustrates the entire system. AVs and infrastructure capture, encode, compress, and send intermediate visual features with each other, while the ego vehicle (*i.e.*, receiver) employs V2X-Transformer to perform information fusion for object detection. We propose two novel attention modules to accommodate V2X challenges: 1) a customized heterogeneous multi-agent self-attention module that explicitly considers agent types (vehicles and infrastructure) and their connections when performing attentive fusion; 2) a multi-scale window attention module that can handle localization errors by using multi-resolution windows in parallel. These two modules will adaptively iteratively fuse visual features to capture inter-agent interaction and per-agent spatial relationship,

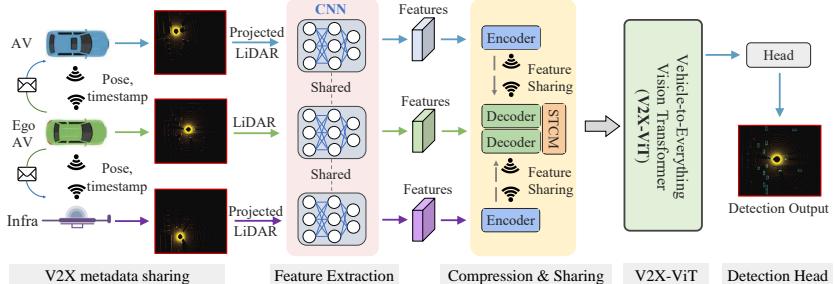


Fig. 2: **Overview of our proposed V2X perception system.** It consists of five sequential steps: V2X metadata sharing, feature extraction, compression & sharing, V2X-ViT, and the detection head. The details of each individual component are illustrated in Sec. 3.1.

correcting the feature misalignment caused by localization error and time delay. Moreover, we also integrate a delay-aware positional encoding to handle the time delay uncertainty further. Notably, all these modules are incorporated in a single transformer that learns to address these challenges end-to-end.

To evaluate our approach, we collect a new large-scale open dataset, namely V2XSet, that explicitly considers real-world noises during V2X communication using the high-fidelity simulator CARLA [10], and a cooperative driving automation simulation tool OpenCDA. Fig. 1 shows a data sample in the collected dataset. Experiments show that our proposed V2X-ViT significantly advances the performance on V2X LiDAR-based 3D object detection, achieving a 21.2% gain of AP compared to single-agent baseline and performing favorably against leading intermediate fusion methods by at least 7.3%. Our contributions are:

- We present the first unified transformer architecture (V2X-ViT) for V2X perception, which can capture the heterogeneity nature of V2X systems with strong robustness against various noises. Moreover, the proposed model achieves state-of-the-art performance on the challenging cooperative detection task.
- We propose a novel heterogeneous multi-agent attention module (HMSA) tailored for adaptive information fusion between heterogeneous agents.
- We present a new multi-scale window attention module (MSwin) that simultaneously captures local and global spatial feature interactions in parallel.
- We construct V2XSet, a new large-scale open simulation dataset for V2X perception, which explicitly accounts for imperfect real-world conditions.

## 2 Related work

**V2X perception.** Cooperative perception studies how to efficiently fuse visual cues from neighboring agents. Based on its message sharing strategy, it can be divided into 3 categories: 1) early fusion [6] where raw data is shared and

gathered to form a holistic view, 2) intermediate fusion [45, 50, 41, 5] where intermediate neural features are extracted based on each agent’s observation and then transmitted, and 3) late fusion [33, 34] where detection outputs (*e.g.*, 3D bounding box position, confidence score) are circulated. As early fusion usually requires large transmission bandwidth and late fusion fails to provide valuable scenario context [45], intermediate fusion has attracted increasing attention because of its good balance between accuracy and transmission bandwidth. Several intermediate fusion methods have been proposed for V2V perception recently. OPV2V [50] implements a single-head self-attention module to fuse features, while F-Cooper employs *maxout* [15] fusion operation. V2VNet [45] proposes a spatial-aware message passing mechanism to jointly reason detection and prediction. To attenuate outlier messages, [41] regresses vehicles’ localization errors with consistent pose constraints. DiscoNet [25] leverages knowledge distillation to enhance training by constraining the corresponding features to the ones from the network for early fusion. However, intermediate fusion for V2X is still in its infancy. Most V2X methods explored late fusion strategies to aggregate information from infrastructure and vehicles. For example, a late fusion two-level Kalman filter is proposed by [31] for roadside infrastructure failure conditions. Xiangmo *et al.* [58] propose fusing the lane mark detection from infrastructure and vehicle sensors, leveraging Dempster-Shafer theory to model the uncertainty.

**LiDAR-based 3D object detection.** Numerous methods have been explored to extract features from raw points, voxels, bird-eye-view (BEV) images, and their mixtures. PointRCNN [36] proposes a two-stage strategy based on raw point clouds, which learns rough estimation in the first stage and then refines it with semantic attributes. The authors of [60, 51] propose to split the space into voxels and produce features per voxel. Despite having high accuracy, their inference speed and memory consumption are difficult to optimize due to reliance on 3D convolutions. To avoid computationally expensive 3D convolutions, [22, 52] propose an efficient BEV representation. To satisfy both computational and flexible receptive field requirements, [35, 59, 53] combine voxel-based and point-based approaches to detect 3D objects.

**Transformers in vision.** The Transformer [43] is first proposed for machine translation [43], where multi-head self-attention and feed-forward layers are stacked to capture long-range interactions between words. Dosovitskiy *et al.* [9] present a Vision Transformer (ViT) for image recognition by regarding image patches as visual words and directly applying self-attention. The full self-attention in ViT [43, 9, 13], despite having global interaction, suffers from heavy computational complexity and does not scale to long-range sequences or high-resolution images. To ameliorate this issue, numerous methods have introduced locality into self-attention, such as Swin [30], CSwin [8], Twins [7], window [46, 39], and sparse attention [42, 40, 49]. A hierarchical architecture is usually adopted to progressively increase the receptive fields for capturing longer dependencies.

While these vision transformers have proven efficient in modeling homogeneous structured data, their efficacy to represent heterogeneous graphs has been less studied. One of the developments related to our work is the heterogeneous

graph transformer (HGT) [18]. HGT was originally designed for web-scale Open Academic Graph where the nodes are text and attributes. Inspired by HGT, we build a customized heterogeneous multi-head self-attention module tailored for graph attribute-aware multi-agent 3D visual feature fusion, which is able to capture the heterogeneity of V2X systems.

### 3 Methodology

In this paper, we consider V2X perception as a heterogeneous multi-agent perception system, where different types of agents (*i.e.*, smart infrastructure and AVs) perceive the surrounding environment and communicate with each other. To simulate real-world scenarios, we assume that all the agents have imperfect localization and time delay exists during feature transmission. Given this, our goal is to develop a robust fusion system to enhance the vehicle’s perception capability and handle these aforementioned challenges in a unified end-to-end fashion. The overall architecture of our framework is illustrated in Fig. 2, which includes five major components: 1) metadata sharing, 2) feature extraction, 3) compression and sharing, 4) V2X vision Transformer, and 5) a detection head.

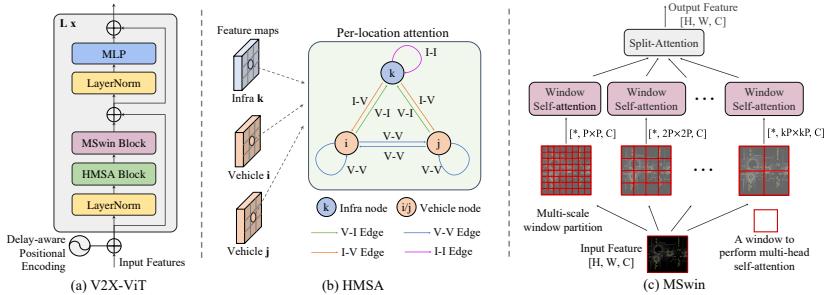
#### 3.1 Main architecture design

**V2X metadata sharing.** During the early stage of collaboration, every agent  $i \in \{1 \dots N\}$  within the communication networks shares metadata such as poses, extrinsics, and agent type  $c_i \in \{I, V\}$  (meaning infrastructure or vehicle) with each other. We select one of the connected AVs as the ego vehicle ( $e$ ) to construct a V2X graph around it where the nodes are either AVs or infrastructure and the edges represent directional V2X communication channels. To be more specific, we assume the transmission of metadata is well-synchronized, which means each agent  $i$  can receive ego pose  $x_e^{t_i}$  at the time  $t_i$ . Upon receiving the pose of the ego vehicle, all the other connected agents nearby will project their own LiDAR point clouds to the ego-vehicle’s coordinate frame before feature extraction.

**Feature extraction.** We leverage the anchor-based PointPillar method [22] to extract visual features from point clouds because of its low inference latency and optimized memory usage [50]. The raw point clouds will be converted to a stacked pillar tensor, then scattered to a 2D pseudo-image and fed to the PointPillar backbone. The backbone extracts informative feature maps  $\mathbf{F}_i^{t_i} \in \mathbb{R}^{H \times W \times C}$ , denoting agent  $i$ ’s feature at time  $t_i$  with height  $H$ , width  $W$ , and channels  $C$ .

**Compression and sharing.** To reduce the required transmission bandwidth, we utilize a series of  $1 \times 1$  convolutions to progressively compress the feature maps along the channel dimension. The compressed features with the size  $(H, W, C')$  (where  $C' \ll C$ ) are then transmitted to the ego vehicle ( $e$ ), on which the features are projected back to  $(H, W, C)$  using  $1 \times 1$  convolutions.

There exists an inevitable time gap between the time when the LiDAR data is captured by connected agents and when the extracted features are received by the ego vehicle (details see appendix). Thus, features collected from surrounding



**Fig. 3: V2X-ViT architecture.** (a) The architecture of our proposed V2X-ViT model. (b) Heterogeneous multi-agent self-attention (HMSA) presented in Sec. 3.2. (c) Multi-scale window attention module (MSwin) illustrated in Sec. 3.2.

agents are often temporally misaligned with the features captured on the ego vehicle. To correct this delay-induced global spatial misalignment, we need to transform (*i.e.*, rotate and translate) the received features to the current ego-vehicle’s pose. Thus, we leverage a spatial-temporal correction module (STCM), which employs a differential transformation and sampling operator  $\Gamma_\xi$  to spatially warp the feature maps [19]. An ROI mask is also calculated to prevent the network from paying attention to the padded zeros caused by the spatial warp. **V2X-ViT**. The intermediate features  $\mathbf{H}_i = \Gamma_\xi(\mathbf{F}_i^{t_i}) \in \mathbb{R}^{H \times W \times C}$  aggregated from connected agents are fed into the major component of our framework *i.e.*, V2X-ViT to conduct an iterative inter-agent and intra-agent feature fusion using self-attention mechanisms. We maintain the feature maps in the same level of high resolution throughout the entire Transformer as we have observed that the absence of high-definition features greatly harms the objection detection performance. The details of our proposed V2X-ViT will be unfolded in Sec. 3.2. **Detection head**. After receiving the final fused feature maps, we apply two  $1 \times 1$  convolution layers for box regression and classification. The regression output is  $(x, y, z, w, l, h, \theta)$ , denoting the position, size, and yaw angle of the predefined anchor boxes, respectively. The classification output is the confidence score of being an object or background for each anchor box. We use the smooth  $\ell_1$  loss for regression and a focal loss [28] for classification.

### 3.2 V2X-Vision Transformer

Our goal is to design a customized vision Transformer that can jointly handle the common V2X challenges. Firstly, to effectively capture the heterogeneous graph representation between infrastructure and AVs, we build a heterogeneous multi-agent self-attention module that learns different relationships based on node and edge types. Moreover, we propose a novel spatial attention module, namely multi-scale window attention (MSwin), that captures long-range interactions at various scales. MSwin uses multiple window sizes to aggregate spatial information, which greatly improves the detection robustness against localization errors. Lastly, these two attention modules are integrated into a single V2X-ViT block

in a factorized manner (illustrated in Fig. 3a), enabling us to maintain high-resolution features throughout the entire process. We stack a series of V2X-ViT blocks to iteratively learn inter-agent interaction and per-agent spatial attention, leading to a robust aggregated feature representation for detection.

**Heterogeneous multi-agent self-attention** The sensor measurements captured by infrastructure and AVs possibly have distinct characteristics. The infrastructure’s LiDAR is often installed at a higher position with less occlusion and different view angles. In addition, the sensors may have different levels of sensor noise due to maintenance frequency, hardware quality *etc.* To encode this heterogeneity, we build a novel heterogeneous multi-agent self-attention (HSMA) where we attach types to both nodes and edges in the directed graph. To simplify the graph structure, we assume the sensor setups among the same category of agents are identical. As shown in Fig. 3b, we have two types of nodes and four types of edges, *i.e.*, node type  $c_i \in \{I, V\}$  and edge type  $\phi(e_{ij}) \in \{V-V, V-I, I-V, I-I\}$ . Note that unlike traditional attention where the node features are treated as a vector, we only reason the interaction of features *in the same spatial position* from different agents to preserve spatial cues. Formally, HSMA is expressed as:

$$\mathbf{H}_i = \text{Dense}_{c_i} \left( \underset{\forall j \in N(i)}{\text{ATT}}(i, j) \cdot \mathbf{MSG}(i, j) \right) \quad (1)$$

which contains 3 operators: a linear aggregator  $\text{Dense}_{c_i}$ , attention weights estimator  $\text{ATT}$ , and message aggregator  $\mathbf{MSG}$ . The Dense is a set of linear projectors indexed by the node type  $c_i$ , aggregating multi-head information.  $\text{ATT}$  calculates the importance weights between pairs of nodes conditioned on the associated node and edge types:

$$\text{ATT}(i, j) = \underset{\forall j \in N(i)}{\text{softmax}} \left( \underset{m \in [1, h]}{\parallel} \text{head}_{\text{ATT}}^m(i, j) \right) \quad (2)$$

$$\text{head}_{\text{ATT}}^m(i, j) = \left( \mathbf{K}^m(j) \mathbf{W}_{\phi(e_{ij})}^{m, \text{ATT}} \mathbf{Q}^m(i)^T \right) \frac{1}{\sqrt{C}} \quad (3)$$

$$\mathbf{K}^m(j) = \text{Dense}_{c_j}^m(\mathbf{H}_j) \quad (4)$$

$$\mathbf{Q}^m(i) = \text{Dense}_{c_i}^m(\mathbf{H}_i) \quad (5)$$

where  $\parallel$  denotes concatenation,  $m$  is the current head number and  $h$  is the total number of heads. Notice that Dense here is indexed by both node type  $c_{i/j}$ , and head number  $m$ . The linear layers in  $\mathbf{K}$  and  $\mathbf{Q}$  have distinct parameters. To incorporate the semantic meaning of edges, we calculate the dot product between Query and Key vectors weighted by a matrix  $\mathbf{W}_{\phi(e_{ij})}^{m, \text{ATT}} \in \mathbb{R}^{C \times C}$ . Similarly, when parsing messages from the neighboring agent, we embed infrastructure and vehicle’s features separately via  $\text{Dense}_{c_j}^m$ . A matrix  $\mathbf{W}_{\phi(e_{ij})}^{m, \text{MSG}}$  is used to project the features based on the edge type between source node and target node:

$$\mathbf{MSG}(i, j) = \parallel_{m \in [1, h]} \text{head}_{\text{MSG}}^m(i, j) \quad (6)$$

$$\text{head}_{\text{MSG}}^m(i, j) = \text{Dense}_{c_j}^m(\mathbf{H}_j) \mathbf{W}_{\phi(e_{ij})}^{m, \text{MSG}}. \quad (7)$$

**Multi-scale window attention** We present a new type of attention mechanism tailored for efficient long-range spatial interaction on high-resolution detection, called multi-scale window attention (MSwin). It uses a pyramid of windows, each of which caps a different attention range, as illustrated in Fig. 3c. The usage of variable window sizes can greatly improve the detection robustness of V2X-ViT against localization errors (see ablation study in Fig. 5b). Attention performed within larger windows can capture long-range visual cues to compensate for large localization errors, whereas smaller window branches perform attention at finer scales to preserve local context. Afterward, the split-attention module [56] is used to adaptively fuse information coming from multiple branches, empowering MSwin to handle a range of pose errors. Note that MSwin is applied on each agent independently without considering any inter-agent fusion; therefore we omit the agent subscript in this subsection for simplicity.

Formally, let  $\mathbf{H} \in \mathbb{R}^{H \times W \times C}$  be an input feature map of a single agent. In branch  $j$  out of  $k$  parallel branches,  $\mathbf{H}$  is partitioned using window size  $P_j \times P_j$ , into a tensor of shape  $(\frac{H}{P_j} \times \frac{W}{P_j}, P_j \times P_j, C)$ , which represents a  $\frac{H}{P_j} \times \frac{W}{P_j}$  grid of non-overlapping patches each with size  $P_j \times P_j$ . We use  $h_j$  number of heads to improve the attention power at  $j$ -th branch. More detailed formulation can be found in Appendix. Following [30, 17], we also consider an additional relative positional encoding  $\mathbf{B}$  that acts as a bias term added to the attention map. As the relative position along each axis lies in the range  $[-P_j + 1, P_j - 1]$ , we take  $\mathbf{B}$  from a parameterized matrix  $\hat{\mathbf{B}} \in \mathbb{R}^{(2P_j-1) \times (2P_j-1)}$ .

To attain per-agent multi-range spatial relationship, each branch partitions input tensor  $\mathbf{H}$  with different window sizes *i.e.*  $\{P_j\}_{j=1}^k = \{P, 2P, \dots, kP\}$ . We progressively decrease the number of heads when using a larger window size to save memory usage. Finally, we fuse the features from all the branches by a Split-Attention module [56], yielding the output feature  $\mathbf{Y}$ . The complexity of the proposed MSwin is *linear* to image size  $HW$ , while enjoying long-range multi-scale receptive fields and adaptively fuses both local and (sub)-global visual hints in parallel. Notably, unlike Swin Transformer [30], our multi-scale window approach requires no masking, padding, or cyclic-shifting, making it more efficient in implementations while having larger-scale spatial interactions.

**Delay-aware positional encoding** Although the global misalignment is captured by the spatial warping matrix  $\Gamma_\xi$ , another type of local misalignment, arising from object motions during the delay-induced time lag, also needs to be considered. To encode this temporal information, we leverage an adaptive delay-aware positional encoding (DPE), composed of a linear projection and a learnable embedding. We initialize it with sinusoid functions conditioned on time

delay  $\Delta t_i$  and channel  $c \in [1, C]$ :

$$\mathbf{p}_c(\Delta t_i) = \begin{cases} \sin\left(\Delta t_i/10000^{\frac{2c}{C}}\right), & c = 2k \\ \cos\left(\Delta t_i/10000^{\frac{2c}{C}}\right), & c = 2k + 1 \end{cases} \quad (8)$$

A linear projection  $f : \mathbb{R}^C \rightarrow \mathbb{R}^C$  will further warp the learnable embedding so it can generalize better for unseen time delay [18]. We add this projected embedding to each agents' feature  $\mathbf{H}_i$  before feeding into the Transformer so that the features are temporally aligned beforehand.

$$\text{DPE}(\Delta t_i) = f(\mathbf{p}(\Delta t_i)) \quad (9)$$

$$\mathbf{H}_i = \mathbf{H}_i + \text{DPE}(\Delta t_i) \quad (10)$$

## 4 Experiments

### 4.1 V2XSet: An open dataset for V2X cooperative perception

To the best of our knowledge, no fully public V2X perception dataset exists suitable for investigating common V2X challenges such as localization error and transmission time delay. DAIR-V2X [2] is a large-scale real-world V2I dataset without V2V cooperation. V2X-Sim [24] is an open V2X simulated dataset but does not simulate noisy settings and only contains a single road type. OPV2V [50] contains more road types but are restricted to V2V cooperation. To this end, we collect a new large-scale dataset for V2X perception that explicitly considers these real-world noises during V2X collaboration using CARLA [10] and OpenCDA [48] together. In total, there are 11,447 frames in our dataset (33,081 samples if we count frames per agent in the same scene), and the train/validation/test splits are 6,694/1,920/2,833, respectively. Compared with existing datasets, V2XSet incorporates both V2X cooperation and realistic noise simulation. Please refer to the supplementary material for more details.

### 4.2 Experimental setup

The evaluation range in  $x$  and  $y$  direction are  $[-140, 140]$  m and  $[-40, 40]$  m respectively. We assess models under two settings: 1) *Perfect Setting*, where the pose is accurate, and everything is synchronized across agents; 2) *Noisy Setting*, where pose error and time delay are both considered. In the *Noisy Setting*, the positional and heading noises of the transmitter are drawn from a Gaussian distribution with a default standard deviation of 0.2 m and 0.2° respectively, following the real-world noise levels [47, 26, 1]. The time delay is set to 100 ms for all the evaluated models to have a fair comparison of their robustness against asynchronous message propagation.

**Evaluation metrics.** The detection performance is measured with Average Precisions (AP) at Intersection-over-Union (IoU) thresholds of 0.5 and 0.7. In

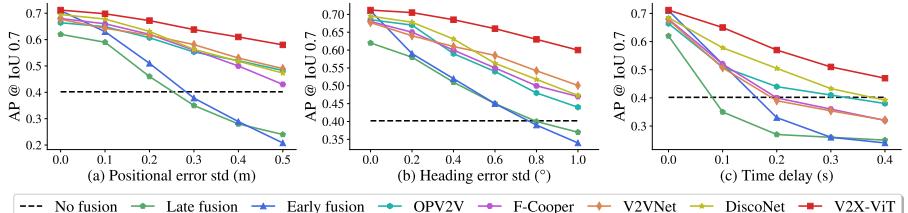


Fig. 4: Robustness assessment on positional and heading errors.

Table 1: 3D detection performance comparison on V2XSet. We show Average Precision (AP) at IoU=0.5, 0.7 on *Perfect* and *Noisy* settings, respectively.

| Models         | Perfect      |              | Noisy        |              |
|----------------|--------------|--------------|--------------|--------------|
|                | AP0.5        | AP0.7        | AP0.5        | AP0.7        |
| No Fusion      | 0.606        | 0.402        | 0.606        | 0.402        |
| Late Fusion    | 0.727        | 0.620        | 0.549        | 0.307        |
| Early Fusion   | 0.819        | 0.710        | 0.720        | 0.384        |
| F-Cooper [5]   | 0.840        | 0.680        | 0.715        | 0.469        |
| OPV2V [50]     | 0.807        | 0.664        | 0.709        | 0.487        |
| V2VNet [45]    | 0.845        | 0.677        | 0.791        | 0.493        |
| DiscoNet [25]  | 0.844        | 0.695        | 0.798        | 0.541        |
| V2X-ViT (Ours) | <b>0.882</b> | <b>0.712</b> | <b>0.836</b> | <b>0.614</b> |

this work, we focus on LiDAR-based vehicle detection. Vehicles hit by at least one LiDAR point from any connected agent will be included as evaluation targets.

**Implementation details.** During training, a random AV is selected as the ego vehicle, while during testing, we evaluate on a fixed ego vehicle for all the compared models. The communication range of each agent is set as 70 m based on [20], whereas all the agents out of this broadcasting radius of ego vehicle is ignored. For the PointPillar backbone, we set the voxel resolution to 0.4 m for both height and width. The default compression rate is 32 for all intermediate fusion methods. Our V2X-ViT has 3 encoder layers with 3 window sizes in MSwin: 4, 8, and 16. We first train the model under the *Perfect Setting*, then fine-tune it for *Noisy Setting*. We adopt Adam optimizer [21] with an initial learning rate of  $10^{-3}$  and steadily decay it every 10 epochs using a factor of 0.1. All models are trained on Tesla V100.

**Compared methods.** We consider *No Fusion* as our baseline, which only uses ego-vehicle’s LiDAR point clouds. We also compare with *Late Fusion*, which gathers all detected outputs from agents and applies Non-maximum suppression to produce the final results, and *Early Fusion*, which directly aggregates raw LiDAR point clouds from nearby agents. For *intermediate fusion* strategy, we evaluate four state-of-the-art approaches: OPV2V [50], F-Cooper [5] V2VNet [45], and DiscoNet [25]. For a fair comparison, all the models use PointPillar as the backbone, and every compared V2V methods also receive infrastructure data, but they do not distinguish between infrastructure and vehicles.

### 4.3 Quantitative evaluation

**Main performance comparison.** Tab. 1 shows the performance comparisons on both *Perfect* and *Noisy Setting*. Under the *Perfect Setting*, all the cooperative methods significantly outperform *No Fusion* baseline. Our proposed V2X-ViT outperforms SOTA intermediate fusion methods by 3.8%/1.7% for AP@0.5/0.7. It is even higher than the ideal *Early fusion* by 0.2% AP@0.7, which receives complete raw information. Under noisy setting, when localization error and time delay are considered, the performance of *Early Fusion* and *Late Fusion* drastically drop to 38.4% and 30.7% in AP@0.7, even worse than single-agent baseline *No Fusion*. Although OPV2V [50], F-Cooper [5] V2VNet [45], and DiscoNet [25] are still higher than *No fusion*, their performance decrease by 17.7%, 21.1%, 18.4% and 15.4% in AP@0.7, respectively. In contrast, V2X-ViT performs favorably against the *No fusion* method by a large margin, *i.e.* 23% and 21.2% higher in AP@0.5 and AP@0.7. Moreover, when compared to the *Perfect Setting*, V2X-ViT only drops by less than 5% and 10% in AP@0.5 and AP@0.7 under *Noisy Setting*, demonstrating its robustness against normal V2X noises. The real-time performance of V2X-ViT is also shown in Tab. 4. The inference time of V2X-ViT is 57 ms, and by using only 1 encoder layer, V2X-ViT<sub>S</sub> can still beat DiscoNet while reaching only 28 ms inference time, which achieves real-time performance.

**Sensitivity to localization error.** To assess the models’ sensitivity to pose error, we sample noises from Gaussian distribution with standard deviation  $\sigma_{xyz} \in [0, 0.5]$  m,  $\sigma_{\text{heading}} \in [0^\circ, 1.0^\circ]$ . As Fig. 4 depicts, when the positional and heading errors stay within a normal range (*i.e.*,  $\sigma_{xyz} \leq 0.2$  m,  $\sigma_{\text{heading}} \leq 0.4^\circ$  [1, 26, 47]), the performance of V2X-ViT only drops by less than 3%, whereas other intermediate fusion methods decrease at least 6%. Moreover, the accuracy of *Early Fusion* and *Late Fusion* degrade by nearly 20% in AP@0.7. When the noise is massive (*e.g.*, 0.5 m and 1°std), V2X-ViT can still stay around 60% detection accuracy while the performance of other methods significantly degrades, showing the robustness of V2X-ViT against pose errors.

**Time delay analysis.** We further investigate the impact of time delay with range [0, 400] ms. As Fig. 4c shows, the AP of *Late Fusion* drops dramatically below *No Fusion* with only 100 ms delay. *Early Fusion* and other intermediate fusion methods are relatively less sensitive, but they still drop rapidly when delay keeps increasing and are all below the baseline after 400 ms. Our V2X-ViT, in contrast, exceeds *No Fusion* by 6.8% in AP@0.7 even under 400 ms delay, which is much larger than usual transmission delay in real-world system[38]. This clearly demonstrates its great robustness against time delay.

**Infrastructure *vs.* vehicles.** To analyze the effect of infrastructure in the V2X system, we evaluate the performance between V2V, where only vehicles can share information, and V2X, where infrastructure can also transmit messages. We denote the number of agents as the total number of infrastructure and vehicles that can share information. As shown in Fig. 5a, both V2V and V2X have better performance when the number of agents increases. The V2X system has better APs compared with V2V in our collected scenes. We argue this is due to the

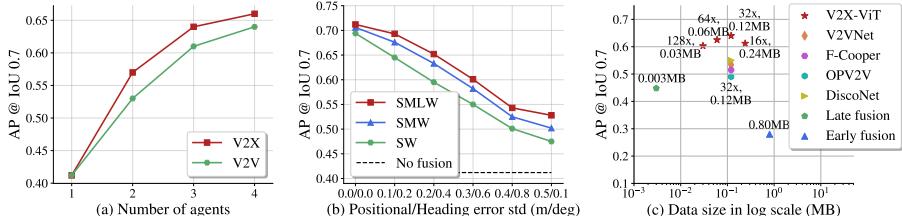


Fig. 5: **Ablation studies.** (a) AP vs. number of agents. (b) MSwin for localization error with window sizes:  $4^2$  (S),  $8^2$  (M),  $16^2$  (L). (c) AP vs. data size.

Table 2: **Component ablation study.** MSwin, SpAttn, HMSA, DPE represent adding i) multi-scale window attention, ii) split attention, iii) heterogeneous multi-agent self-attention, and iv) delay-aware positional encoding, respectively.

|   | MSwin | SpAttn | HMSA | DPE | AP0.5 / AP0.7        |
|---|-------|--------|------|-----|----------------------|
| ✓ |       |        |      |     | 0.719 / 0.478        |
| ✓ |       | ✓      |      |     | 0.748 / 0.519        |
| ✓ | ✓     |        | ✓    |     | 0.786 / 0.548        |
| ✓ | ✓     | ✓      | ✓    | ✓   | 0.823 / 0.601        |
|   |       |        |      |     | <b>0.836 / 0.614</b> |

better sight-of-view and less occlusion of infrastructure sensors, leading to more informative features for reasoning the environmental context.

**Effects of transmission size.** The size of the transmitted message can significantly affect the transmission delay, thereby affecting the detection performance. Here we study the model’s detection performance with respect to transmitted data size. The data transmission time is calculated by  $t_c = f_s/v$ , where  $f_s$  denotes the feature size and transmission rate  $v$  is set to 27 Mbps [3]. Following [32], we also include another system-wise asynchronous delay that follows a uniform distribution between 0 and 200 ms. See supplementary materials for more details. From Fig. 5c, we can observe: 1) Large bandwidth requirement can eliminate the advantages of cooperative perception quickly, e.g., *Early Fusion* drops to 28%, indicating the necessity of compression; 2) With the default compression rate (32x), our V2X-ViT outperforms other intermediate fusion methods substantially; 3) V2X-ViT is insensitive to large compression rate. Even under a 128x compression rate, our model can still maintain high performance.

#### 4.4 Qualitative evaluation

**Detection visualization.** Fig. 6 shows the detection visualization of OPV2V, V2VNet, DiscoNet, and V2X-ViT in two challenging scenarios under *Noisy setting*. Our model predicts highly accurate bounding boxes which are well-aligned

Table 3: **Effect of DPE w.r.t. time delay on AP@0.7.**

| Delay/Model | w/o DPE | w/ DPE |
|-------------|---------|--------|
| 100 ms      | 0.639   | 0.650  |
| 200 ms      | 0.558   | 0.572  |
| 300 ms      | 0.496   | 0.514  |
| 400 ms      | 0.458   | 0.478  |

Table 4: **Inference time** measured on GPU Tesla V100.

| Model                | Time | AP0.7(prf/nsy) |
|----------------------|------|----------------|
| V2X-ViT <sub>S</sub> | 28ms | 0.696 / 0.591  |
| V2X-ViT              | 57ms | 0.712 / 0.614  |

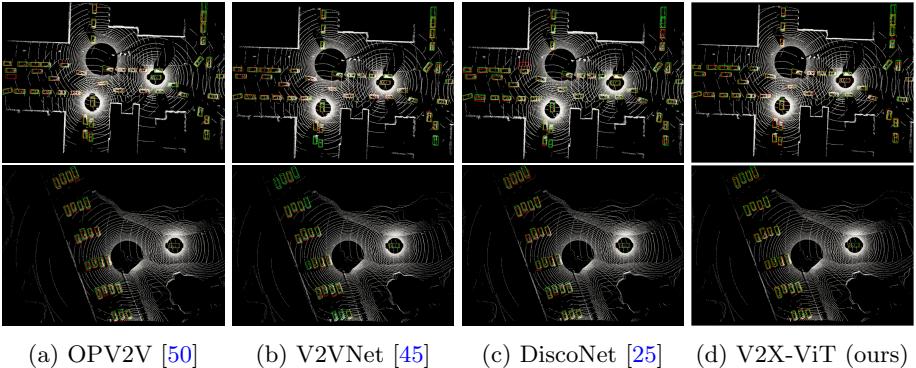


Fig. 6: **Qualitative comparison in a congested intersection and a highway entrance ramp.** Green and red 3D bounding boxes represent the ground truth and prediction respectively. Our method yields more accurate detection results. More visual examples are provided in the supplementary materials.

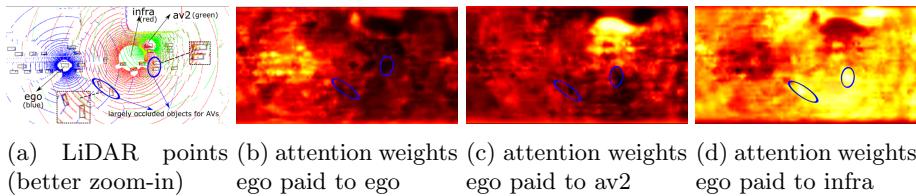


Fig. 7: **Aggregated LiDAR points and attention maps for ego.** Several objects are occluded (blue circle) from both AV's perspectives, whereas infra can still capture rich point clouds. V2X-ViT learned to pay more attention to infra on occluded areas, shown in (d). We provide more visualizations in Appendix.

with ground truths, while other approaches exhibit larger displacements. More importantly, V2X-ViT can identify more dynamic objects (more ground-truth bounding boxes have matches), which proves its capability of effectively fusing all sensing information from nearby agents. Please see Appendix for more results.

**Attention map visualization.** To understand the importance of infra, we also visualize the learned attention maps in Fig. 7, where brighter color means more attention ego pays. As shown in Fig. 7a, several objects are largely occluded (circled in blue) from both ego and AV2's perspectives, whereas infrastructure can still capture rich point clouds. Therefore, V2X-ViT pays much more attention to infra on occluded areas (Fig. 7d) than other agents (Figs. 7b and 7c), demonstrating the critical role of infra on occlusions. Moreover, the attention map for infra is generally brighter than the vehicles, indicating more importance on infra seen by the trained V2X-ViT model.

#### 4.5 Ablation studies

**Contribution of major components in V2X-ViT.** Now we investigate the effectiveness of individual components in V2X-ViT. Our base model is Point-

Pillars with naive multi-agent self-attention fusion, which treats vehicles and infrastructure equally. We evaluate the impact of each component by progressively adding i) MSwin, ii) split attention, iii) HMSA, and iv) DPE on the *Noisy Setting*. As Tab. 2 demonstrates, all the modules are beneficial to the performance gains, while our proposed MSwin and HMSA have the most significant contributions by increasing the AP@0.7 4.1% and 6.6%, respectively.

**MSwin for localization error.** To validate the effectiveness of the multi-scale design in MSwin on localization error, we compare three different window configurations: i) using a single small window branch (SW), ii) using a small and a middle window (SMW), and iii) using all three window branches (SMLW). We simulate the localization error by combining different levels of positional and heading noises. From Fig. 5b, we can clearly observe that using a large and small window in parallel remarkably increased its robustness against localization error, which validates the design benefits of MSwin.

**DPE Performance under delay.** Tab. 3 shows that DPE can improve the performance under various time delays. The AP gain increases as delay increases.

## 5 Conclusion

In this paper, we propose a new vision transformer (V2X-ViT) for V2X perception. Its key components are two novel attention modules *i.e.* HMSA and MSwin, which can capture heterogeneous inter-agent interactions and multi-scale intra-agent spatial relationship. To evaluate our approach, we construct V2XSet, a new large-scale V2X perception dataset. Extensive experiments show that V2X-ViT can significantly boost cooperative 3D object detection under both perfect and noisy settings. This work focuses on LiDAR-based cooperative 3D vehicle detection, limited to single sensor modality and vehicle detection task. Our future work involves multi-sensor fusion for joint V2X perception and prediction.

**Broader impacts and limitations.** The proposed model can be deployed to improve the performance and robustness of autonomous driving systems by incorporating V2X communication using a novel vision Transformer. However, for models trained on simulated datasets, there are known issues on data bias and generalization ability to real-world scenarios. Furthermore, although the design choice of our communication approach (*i.e.*, project LiDAR to others at the beginning) has an advantage of accuracy (see supplementary for details), its scalability is limited. In addition, new concerns around privacy and adversarial robustness may arise during data capturing and sharing, which has not received much attention. This work facilitates future research on fairness, privacy, and robustness in visual learning systems for autonomous vehicles.

**Acknowledgement.** This material is supported in part by the Federal Highway Administration Exploratory Advanced Research (EAR) Program, and by the US National Science Foundation through Grants CMMI # 1901998. We thank Xiaoyu Dong for her insightful discussions.

## Appendix

In this supplementary material, we first discuss the design choice and scalability issue of our method (Appendix A). Then, we provide more model details and analysis (Appendix B) in regards to the proposed architecture, including the mathematical details of the spatial-temporal correction module (Appendix B.1), details of the proposed MSwin (Appendix B.2), and the overall architectural specifications (Appendix B.3). Afterwards, additional information and visualizations of the proposed V2XSet dataset are shown in Appendix C. In the end, we present more quantitative experiments, qualitative detection results, attention map visualizations, and details about the effects of the transmission size experiment in Appendix D.

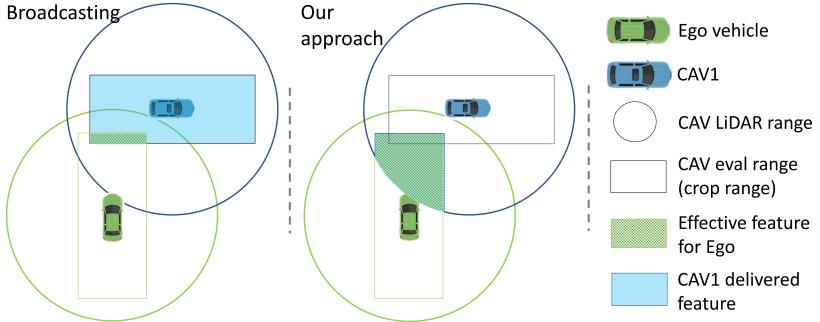
## A Discussion of design choice

**Scalability of ego vehicles.** Our approach can be scalable in two ways: 1) Decentralized: the ablation studies conducted in this paper (Fig. 5a) and OPV2V [50] indicate that, when the number of collaborators is larger than 4, the performance gain becomes marginal while the computation still increases linearly. In practice, each agent only needs to share features with a limited number of agents. For example, Who2Com [29] studies which agent to request/transmit data, largely reducing computation. Moreover, the computation of selected PointPillar backbone is efficient, e.g., around 4 ms for 4 agents with full parallelization and 16 ms in sequence computing on RTX3090. 2) Centralized: Within a certain communication range, only one ego agent is selected to aggregate all the features from neighbors to predict bounding boxes and share the results with other agents. This solution requires only one computation node for a group of agents, thus being scalable.

Table T0: Comparison between our design choice and broadcasting approach.

|                  | DiscoNet (broad. / ours) | V2X-ViT (broad. / ours) |
|------------------|--------------------------|-------------------------|
| AP@0.7 (perfect) | 0.610 / 0.695            | 0.623 / 0.712           |

**Design choices for communication.** Compared to the broadcasting approach (*i.e.*, compute the features in each cav’s own space and transform the feature maps directly on the ego side), our approach has more advantages in terms of detection accuracy. Most LiDAR detection methods often largely crop the LiDAR range based on the evaluation range to reduce computation. As the figure below shows, the CAVs crop the LiDAR data based on their own evaluation range in the broadcasting method, which leads to redundant data. Our approach, on the contrary, always does cropping based on the ego’s evaluation range, thus guaranteeing more effective feature transmission. We further validate this by comparing our framework with the broadcasting approach. The Tab. T0 below shows that our design outperforms broadcasting by 8.5% and 8.9% for DiscoNet and V2X-ViT.



## B Model Details and Analysis

### B.1 Spatial-Temporal Correction Module

During the early stage of collaboration, when each connected agent  $i$  receives ego vehicle's pose at time  $t_i$ , the observed point clouds of agent  $i$  will be projected to ego vehicle's pose  $x_e^{t_i}$  at time  $t_i$  before feature extraction. However, due to the time delay, the ego vehicle observes the data at a different time  $t_e$ . Thus, the received features from connected agents are centered around a delayed ego vehicle's pose (*i.e.*,  $x_e^{t_i}$ ) while the ego vehicle's features are centered around current pose (*i.e.*,  $x_e^{t_e}$ ), leading to a delay-induced spatial misalignment. To correct this misalignment between the received features and ego-vehicle's features, a global transformation  $\xi_{x_e^{t_i}, x_e^{t_e}} \in \mathfrak{se}(3)$  from ego vehicle's past pose  $x_e^{t_i}$  to its current pose  $x_e^{t_e}$  is required. To this end, we employ a differential 2D transformation  $\Gamma_\xi(\cdot)$  to warp the intermediate features spatially [19]. To be more specific, we will transform features' positions by using affine transformation:

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \Gamma_\xi\left(\begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix}\right) = \begin{bmatrix} R_{11} & R_{12} & \delta x \\ R_{21} & R_{22} & \delta y \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix} \quad (11)$$

where  $(x_s, y_s)$  and  $(x_t, y_t)$  are the source and target coordinates. As the calculated coordinates may not be integers, we use bilinear interpolation to sample input feature vectors. An ROI mask is also calculated to prevent the network from paying attention to the padded zeros caused by the spatial warp. This mask will be used in heterogeneous multi-agent self-attention to mask padded values' attention weights as zeros.

### B.2 Multi-Scale Window Attention (MSwin)

**Detailed formulation.** Let  $\mathbf{H} \in \mathbb{R}^{H \times W \times C}$  be an input feature of a single agent. Let  $h_j$  be the number of attention heads used in branch  $j$  (*i.e.* head dimension  $d_{h_j} = C/h_j$ ), applying self-attention within each non-overlapping window  $P_j \times P_j$

for branch  $j$  out of  $k$  branches on feature  $\mathbf{H}$  can be formulated as:

$$\mathbf{H} = [\mathbf{H}^1, \mathbf{H}^2, \dots, \mathbf{H}^{HW/(P_j)^2}], \quad \text{for branch } j \quad (12)$$

$$\hat{\mathbf{H}}_m^i = \text{Attention}(\mathbf{H}^i \mathbf{W}_m^Q, \mathbf{H}^i \mathbf{W}_m^K, \mathbf{H}^i \mathbf{W}_m^V), \quad i = 1, \dots, HW/(P_j)^2 \quad (13)$$

$$\mathbf{Y}_m = [\hat{\mathbf{H}}_m^1, \hat{\mathbf{H}}_m^2, \dots, \hat{\mathbf{H}}_m^{HW/(P_j)^2}], \quad m = 1, \dots, h_j \quad (14)$$

$$\mathbf{Y}^j = [\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_{h_j}], \quad (15)$$

where  $\hat{\mathbf{H}}_m^i \in \mathbb{R}^{P_j^2 \times d_{h_j}}$  and  $\mathbf{W}_m^Q, \mathbf{W}_m^K, \mathbf{W}_m^V$  represent the query, key, and value projection matrices.  $\mathbf{Y}_m$  is the output of the  $m$ -th head for branch  $j$ . Afterwards, the outputs for all heads  $1, 2, \dots, h_j$  are concatenated to obtain the final output  $\mathbf{Y}^j$ . Here the Attention operation denotes the relative self-attention, similar to the usage in Swin [30]:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}} + \mathbf{B}\right)\mathbf{V}\right) \quad (16)$$

where  $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{P_j^2 \times d}$  denote the query, key, and value matrices.  $d$  is the dimension of query/key, while  $P_j^2$  denotes the window size for branch  $j$ . Following [30, 17], we also consider an additional relative positional encoding  $\mathbf{B}$  that acts as a bias term added to the attention map. As the relative position along each axis lies in the range  $[-P_j + 1, P_j - 1]$ , we take  $\mathbf{B}$  from a parameterized matrix  $\hat{\mathbf{B}} \in \mathbb{R}^{(2P_j-1) \times (2P_j-1)}$ . To adaptively fuse features from all the  $k$  branches, we adopt the split-attention module [56] for the parallel feature aggregation:

$$\mathbf{Y} = \text{SplitAttention}(\mathbf{Y}^1, \mathbf{Y}^2, \dots, \mathbf{Y}^k), \quad (17)$$

**Time complexity.** As mentioned in the paper, we have  $k$  parallel branches. Each branch has  $P_j \times P_j$  window size and  $h_j$  heads where  $P_j = jP$  and  $h_j = h/j$ . After partitioning, the input tensor  $\mathbf{H} \in \mathbb{R}^{H \times W \times C}$  is split into  $h_j$  features with shape  $(\frac{H}{P_j} \times \frac{W}{P_j}, P_j \times P_j, C/h_j)$ . Following [30], we focus on the computation for vector-matrix multiplication and attention weight calculation. Thus, the complexity of MSwin can be written as:

$$\begin{aligned} & \mathcal{O}\left(\sum_{j=1}^k \frac{HW}{P_j^2} \times \frac{C}{h_j} \times (P_j \times P_j)^2 \times h_j + 4 \frac{HW}{P_j^2} \times P_j^2 \times \left(\frac{C}{h_j}\right)^2 \times h_j\right) \\ &= \mathcal{O}\left(\sum_{j=1}^k P_j^2 HWC + \frac{4HWC^2}{h_j}\right) = \mathcal{O}\left(\sum_{j=1}^k j^2 P^2 HWC + \frac{4HWC^2 j}{h}\right) \\ &= \mathcal{O}\left(\frac{1}{3} k^3 P^2 HWC + \frac{2HWC^2 k^2}{h}\right) \end{aligned} \quad (18)$$

where the first term corresponds to attention weight calculation, the second term is associated with vector-matrix multiplication, and the last equality is due to the fact that  $\sum_{j=1}^k j^2 = \mathcal{O}(\frac{k^3}{3})$  and  $\sum_{j=1}^k j = \mathcal{O}(\frac{k^2}{2})$ . Thus the overall complexity is

$$\text{FLOPs(MSwin)} = \mathcal{O}\left(\left(\frac{k^3 P^2 C}{3} + \frac{2k^2 C^2}{h}\right)HW\right) \sim \mathcal{O}(HW), \quad (19)$$

which is linear with respect to the image size. The comparison of time complexity of different types of transformers is shown in Tab. T1 where  $N$  denotes the number of input pixels, or (here  $N = HW$ ). Our MSwin obtains multi-scale spatial interactions with a linear complexity with respect to  $N$ , while other long-range attention mechanisms like ViT [9], Axial [44], and CSwin [8] requires more than linear complexity, which are not scalable to high-resolution dense prediction tasks such as object detection and segmentation.

Table T1: Computational complexity comparisons of our proposed MSwin attention with (a) full attention in ViT [9], (b) Axial [44], (c) Swin [30], (d) CSwin [8].

| Attention Models | Complexity  |
|------------------|---|
| ViT [9]          | $\mathcal{O}(4HWC^2 + 2(HW)^2C) \sim \mathcal{O}(N^2)$                        |
| Axial [44]       | $\mathcal{O}(HWC(4C + H + W)) \sim \mathcal{O}(N\sqrt{N})$                    |
| Swin [30]        | $\mathcal{O}(4HWC^2 + 2P^2HWC) \sim \mathcal{O}(N)$                           |
| CSwin [8]        | $\mathcal{O}(HWC(4C + sH + sW)) \sim \mathcal{O}(N\sqrt{N})$                  |
| MSwin (ours)     | $\mathcal{O}(\frac{1}{3}k^3P^2HWC + \frac{2HWC^2k^2}{h}) \sim \mathcal{O}(N)$ |

**Effective receptive field.** The comparisons of receptive fields between different transformers are shown in Fig. 8. Swin [30] enlarge the receptive fields by using shifted window but it requires sequential blocks to accumulate. Axial Transformer [44] conducts attention on both row-wise and column-wise directions. Similarly, CSwin [8] proposes to perform attention on horizontal and vertical stripes with asymmetrical receptive range in different directions, but requires polynomial time complexity— $\mathcal{O}(N^{1.5})$ . In contrast, our proposed MSwin can aggregate features from multi-scale branches to increase fields in parallel, which has more symmetrical receptive fields and linear complexity with respect to  $N$ .

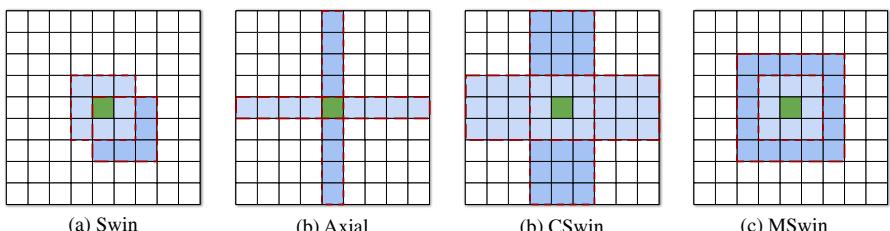


Fig. 8: Visualizations of approximated receptive fields (blue shaded pixels) for the green pixel for (a) Swin [30] (b) Axial [44], (c) CSwin [8] and (d) MSwin attention. MSwin obtains multi-scale long-range interactions at linear complexity.

Table T2: Detailed architectural specifications for V2X-ViT.

|                           | Output size                         | V2X-ViT framework   |
|---------------------------|-------------------------------------|---|
| PointPillar Encoder       | $M \times 352 \times 96 \times 256$ | $\begin{bmatrix} \text{Voxel samp. reso. 0.4m, Scatter, 64} \\ \text{Conv3x3, 64, stride 2, BN, ReLU} \end{bmatrix} \times 3$ $\begin{bmatrix} \text{Conv3x3, 128, stride 2, BN, ReLU} \end{bmatrix} \times 5$ $\begin{bmatrix} \text{Conv3x3, 256, stride 2, BN, ReLU} \end{bmatrix} \times 8$ $\begin{bmatrix} \text{ConvT3x3, 128, stride 1, BN, ReLU} \end{bmatrix} \times 1$ $\begin{bmatrix} \text{ConvT3x3, 128, stride 2, BN, ReLU} \end{bmatrix} \times 1$ $\begin{bmatrix} \text{ConvT3x3, 128, stride 4, BN, ReLU} \end{bmatrix} \times 1$ |
|                           | $M \times 176 \times 48 \times 256$ | $\begin{bmatrix} \text{Concat3, 384} \end{bmatrix}$ $\begin{bmatrix} \text{Conv3x3, 256, stride 2, ReLU} \end{bmatrix} \times 1$ $\begin{bmatrix} \text{Conv3x3, 256, stride 1, ReLU} \end{bmatrix}$  |
| Delay-aware Pos. Encoding | $M \times 176 \times 48 \times 256$ | $\begin{bmatrix} \text{sin-cos pos. encoding} \\ \text{Linear, 256} \end{bmatrix} \times 1$   |
| Transformer Backbone      | $M \times 176 \times 48 \times 256$ | $\begin{bmatrix} \text{HSMA, dim 256, head 8} \\ \text{MSwin, dim 256, head } \{16, 8, 4\}, \\ \text{ws. } \{4 \times 4, 8 \times 8, 16 \times 16\} \\ \text{MLP, dim 256} \end{bmatrix} \times 3$  |
| Detection Head            | $176 \times 48 \times 16$           | $\text{Cls. head: } \begin{bmatrix} \text{Conv1x1, 2, stride 1} \end{bmatrix}$ $\text{Regr. head: } \begin{bmatrix} \text{Conv1x1, 14, stride 1} \end{bmatrix}$   |

### B.3 Architectural Configurations

Given all these definitions, the entire V2X-ViT model can be formulated as:

$$z_i = \text{PointPillar}(x_i), \quad x_i \in \mathbb{R}^{P \times 4}, z_i \in \mathbb{R}^{H \times W \times C} \quad \text{for agent } i \quad (20)$$

$$\mathbf{z}_0 = \text{STCM}([z_0, \dots, z_M]) + \text{DPE}([\Delta t_0, \dots, \Delta t_M]), \quad \text{for ego AV} \quad (21)$$

$$\mathbf{z}'_\ell = \mathbf{z}_{\ell-1} + \text{MSwin}(\text{HSMA}(\text{LN}(\mathbf{z}_0))), \quad \mathbf{z}_0 \in \mathbb{R}^{M \times H \times W \times C} \quad \ell = 1, \dots, L \quad (22)$$

$$\mathbf{z}_\ell = \mathbf{z}'_\ell + \text{MLP}(\text{LN}(\mathbf{z}'_\ell)), \quad \ell = 1, \dots, L \quad (23)$$

$$\mathbf{y} = \text{Head}(\mathbf{z}_L), \quad (24)$$

where the input  $x_i$  denotes the raw LiDAR point clouds captured on each agent, which are fed into the PointPillar Encoder [22], yielding visually informative 2D features  $z_i$  for each agent  $i$ . These tensors are then compressed, shared, decompressed, and further fed into the spatial-temporal correction module (STCM) to spatially warp the features. Then, we add the delay-aware positional encoded (DPE) features conditioned on each agent’s time delay  $\Delta t_i$  to the output of STCM. Afterwards, the gathered features from  $M$  agents are processed using our proposed V2X-ViT, which consists of  $L$  layers of V2X-ViT blocks. Each V2X-ViT block contains a HSMA, a MSwin, and a standard MLP network [9]. Following [9,30], we use the Layer Normalization [4] before feeding into the Transformer/MLP module. We show the detailed specifications of V2X-ViT architecture in Table T2.

## C V2XSet Dataset

**Statistics** We gather 55 representative scenes covering 5 different roadway types and 8 towns in CARLA. Each scene is limited to 25 seconds, and in each scene, there are at least 2 and at most 7 intelligent agents that can communicate with each other. Each agent is equipped with 32-channel LiDAR and has 120 meters data range. We mount sensors on top of each AV while we only deploy infrastructure sensors in the intersection, mid-block, and entrance ramp at the height of 14 feet since these scenarios are typically more congested and challenging [16]. We record LiDAR point clouds at 10 Hz and save the corresponding positional data and timestamp.

**Infrastructure deployment.** The infrastructure sensors are installed on the traffic light poles or street light poles at the intersection, mid-block, and entrance ramp at the height of 14 feet. For road type like rural curvy road, there is no infrastructure installed and only V2V collaboration exists.

**Dataset visualization.** As Fig. 9 displays, there are 5 different roadway types in V2XSet dataset (*i.e.*, straight segment, curvy segment, midblock, entrance ramp, and intersection), covering the most common driving scenarios in real life. We collect more intersection scenes than other types as it is usually more challenging due to the high traffic volume and severe occlusions. Data samples from different roadway types can be found in Fig. 10. From the figure, we can observe that the infrastructure sensors at the entrance ramp and intersection have different measurement patterns especially near its installation position compared with vehicle sensors. This is caused by the different installation heights between vehicle and infrastructure sensors. Such observation again shows the necessity of capturing the heterogeneity nature of V2X system.

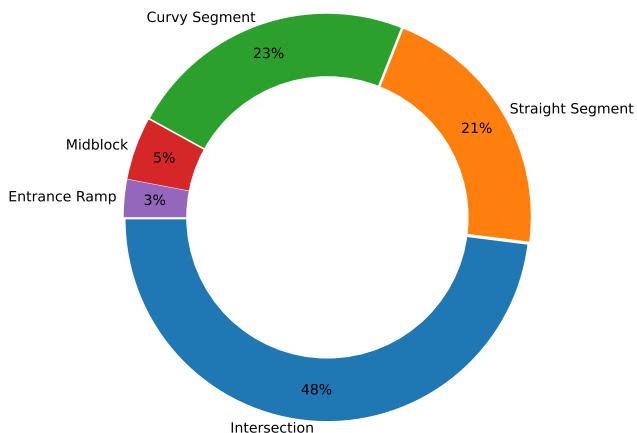


Fig. 9: Data distribution of 5 roadway types in the proposed dataset.

## D More Experimental Results

### D.1 Performance for identifying dynamic objects

We group the test set based on object speeds  $v$  (km/h) and compare AP@IoU=0.7 under the noisy setting for all intermediate fusion models. As shown in Tab. T3, V2X-ViT outperforms all other intermediate fusion methods under various speed range. It is noticeable that the objects with higher speed range generally have lower AP scores as the same time delay can produce more positional misalignments for the high-speed vehicles.

Table T3: Perception performance for objects with different speed (km/h), measured in AP@0.7 under noisy setting.

| Model    | $v < 20$     | $20 \leq v \leq 40$ | $v > 40$     |
|----------|--------------|---------------------|--------------|
| F-Cooper | 0.539        | 0.487               | 0.354        |
| OPV2V    | 0.552        | 0.498               | 0.346        |
| V2VNet   | 0.598        | 0.518               | 0.406        |
| DiscoNet | 0.639        | 0.580               | 0.420        |
| V2X-ViT  | <b>0.693</b> | <b>0.634</b>        | <b>0.488</b> |

### D.2 Performance for different road types

We also group the test scenes based on their road types and calculate the AP@IoU=0.7 scores under the noisy setting. As shown in Tab. T4, V2X-ViT ranks the first for all 5 road categories, demonstrating its detection robustness on different scenes.

Table T4: Perception performance for different road types, measured in AP@0.7 under noisy setting.

| Model    | Straight     | Curvy        | Intersection | Midblock     | Entrance     |
|----------|--------------|--------------|--------------|--------------|--------------|
| F-Cooper | 0.483        | 0.558        | 0.458        | 0.431        | 0.375        |
| OPV2V    | 0.478        | 0.604        | 0.492        | 0.460        | 0.380        |
| V2VNet   | 0.496        | 0.556        | 0.517        | 0.489        | 0.360        |
| DiscoNet | 0.519        | 0.594        | 0.572        | 0.472        | 0.440        |
| V2X-ViT  | <b>0.645</b> | <b>0.686</b> | <b>0.615</b> | <b>0.530</b> | <b>0.487</b> |

### D.3 Qualitative results

Figs. 11 to 13 demonstrate more detection visualizations of V2VNet[45], OPV2V [50], F-Cooper [5], DiscoNet [25], and our V2X-ViT in different scenarios under *Noisy Setting*. V2X-ViT yields more robust performance in general

with fewer regression displacements and fewer undetected objects. When the scenario is challenging with high-density traffic flow and more occlusions (*e.g.*, Scene 7 in Fig. 13), our model can still identify most of the objects accurately.

#### D.4 Attention visualization

Fig. 14 shows more attention map visualizations of V2X-ViT under *noisy setting*. The LiDAR points of ego vehicle, the other connected autonomous vehicle (cav), and infrastructure are plotted in blue, green, and red respectively. The brighter color in the attention map means more attention ego vehicle pays. Generally, the color of infrastructure attention maps is brighter than others, especially for the occluded regions of other agents, indicating the more importance ego vehicle assigns to the infrastructure. This observation agrees with our intuition that the sensor observation of infrastructure has fewer occlusions, which leads to better feature representations.

#### D.5 Explanation on effects of transmission size

Here we provide more explanations of the data transmission size experiment in our paper. Different fusion strategies usually have distinct bandwidth requirements *e.g.*, early fusion requires large bandwidth to transmit raw data, whereas late fusion only delivers minimal size of data. This communication volume will significantly influence the time delay, thus we need to simulate a more realistic time delay setting to study the effects of transmission size.

Following [32], we decompose the total time delay into two parts: i) the data transmission time  $t_c$  during broadcasting, ii) the idle time  $t_i$  caused by the lack of synchronization between the perception system and communication system. The total time delay is calculated as

$$t_{total} = t_c + t_i \quad (25)$$

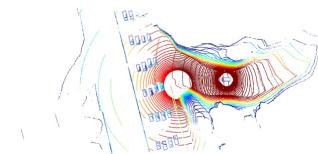
As mentioned in the paper, the data transmission time has

$$t_c = f_s/v \quad (26)$$

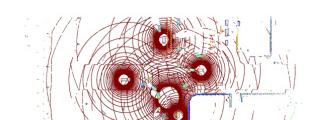
where  $f_s$  is the data size and  $v$  is the transmission rate. Idle time  $t_i$  can be further decoupled into the idle time on the sender side and the time on the receiver side *i.e.*,  $t_i = t_{i,1} + t_{i,2}$ . For  $t_{i,1}$ , the worst case in terms of delay happens when the communication system just misses a perception cycle and needs to wait for the next round. Similarly, for  $t_{i,2}$ , the worst case occurs when new data is received just after a new cycle of the perception system has started. Assume both perception system and communication system have the same rate of 10Hz, then  $0\text{ ms} < t_i < 200\text{ ms}$ . We employ a uniform distribution  $\mathcal{U}(0, 200)$  to model this uncertainty. In summary, we use the following equation to mimic the real-world time delay.

$$t_c = f_s/v + \mathcal{U}(0, 200) \quad (27)$$

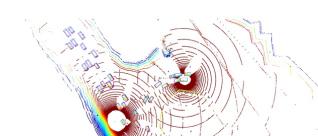
which captures the influence of transmission size and asynchrony-caused uncertainty. In practice, we sample the time delay according to Eq. 26 and discretize it to the observed timestamps, which are discrete in a 10Hz update system.



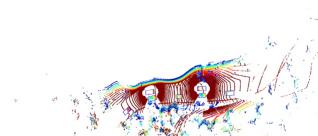
(a) Entrance ramp



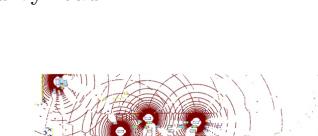
(b) Interaction



(c) Mid-block



(d) Rural curvy road



(e) Urban curvy street

Fig. 10: **Data samples of 5 different roadway types.** Left is the snapshot of simulation and right is the corresponding aggregated LiDAR point clouds from multiple agents.

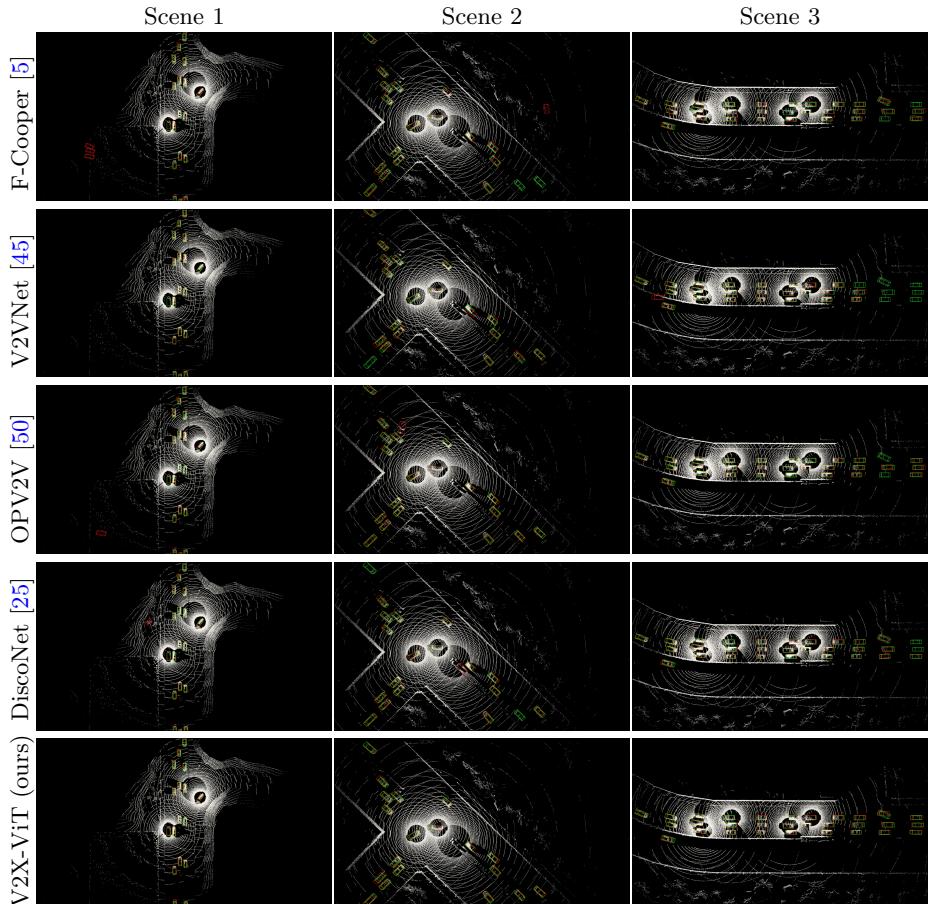


Fig. 11: **Qualitative comparison on scenarios 1-3.** Green and red 3D bounding boxes represent the ground truth and prediction respectively. Our method yields more accurate detection results.

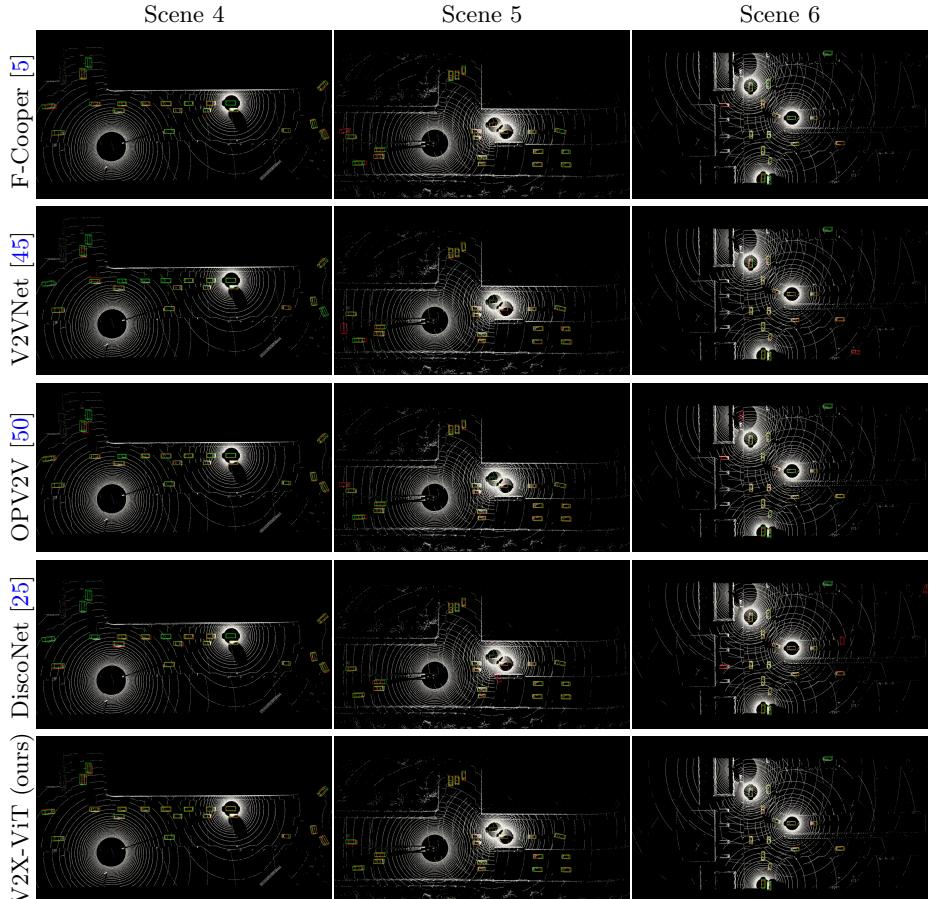
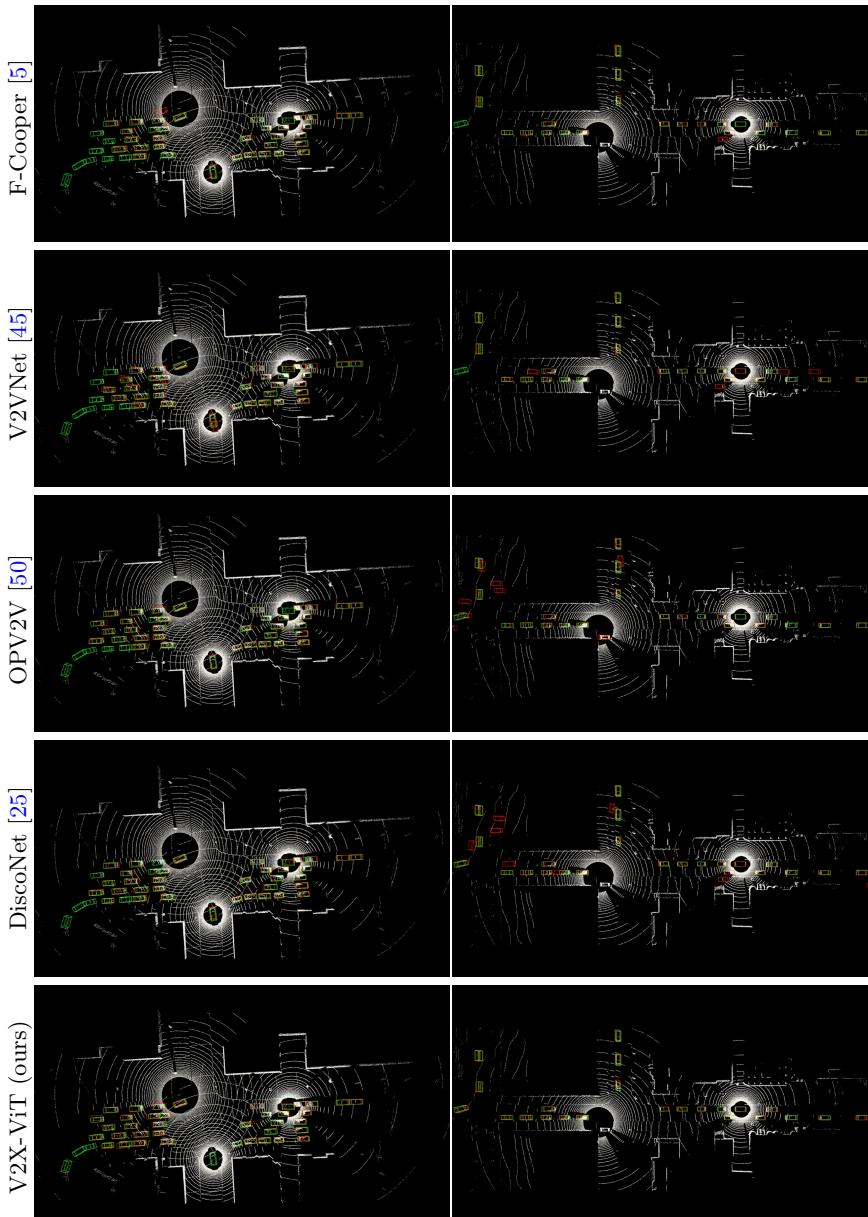


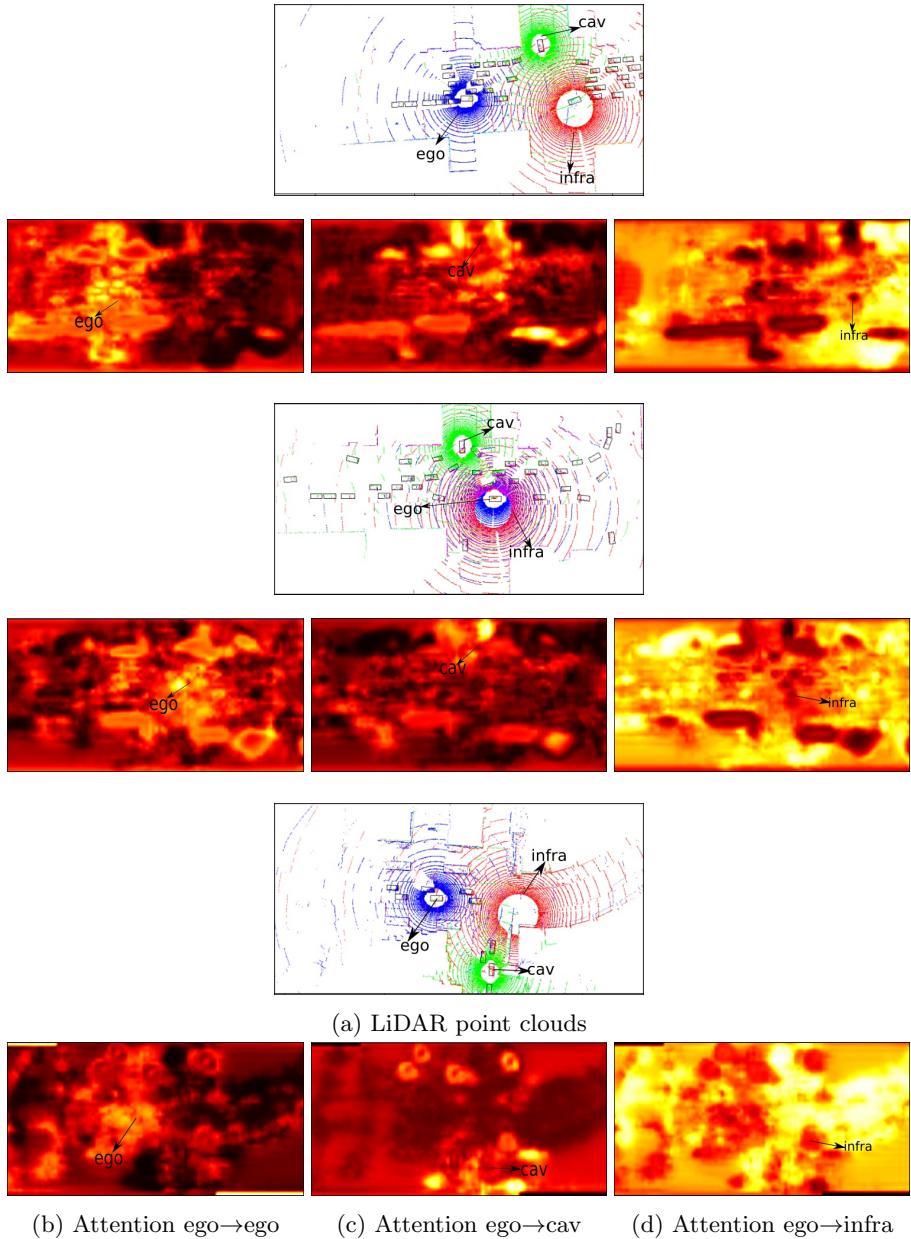
Fig. 12: Qualitative comparison on scenarios 4-6. Green and red 3D bounding boxes represent the ground truth and prediction respectively.

## Scene 7



## Scene 8

Fig. 13: **Qualitative comparison on scenarios 7-8.** Green and red 3D bounding boxes represent the ground truth and prediction respectively.



**Fig. 14: Additional attention map visualizations on 3 different scenes.** V2X-ViT learned to pay more attention to infra features on occluded areas from AV's perspectives, thus yielding more robust detection under occlusions.

## References

1. Rt3000. <https://www.oxts.com/products/rt3000-v3>, accessed: 2021-11-11 [9](#), [11](#)
2. Institut for AI Industry Research (AIR), T.U.: Vehicle-infrastructure cooperative autonomous driving: Dair-v2x dataset (2021) [9](#)
3. Arena, F., Pau, G.: An overview of vehicular communications. Future Internet **11**(2), 27 (2019) [12](#)
4. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016) [19](#)
5. Chen, Q., Ma, X., Tang, S., Guo, J., Yang, Q., Fu, S.: 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. pp. 88–100 (2019) [2](#), [4](#), [10](#), [11](#), [21](#), [24](#), [25](#), [26](#)
6. Chen, Q., Tang, S., Yang, Q., Fu, S.: Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). pp. 514–524. OPTorganization (2019) [2](#), [3](#)
7. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C.: Twins: Revisiting the design of spatial attention in vision transformers. arXiv preprint arXiv:2104.13840 **1**(2), 3 (2021) [4](#)
8. Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B.: Cswin transformer: A general vision transformer backbone with cross-shaped windows. arXiv preprint arXiv:2107.00652 (2021) [4](#), [18](#)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) [4](#), [18](#), [19](#)
10. Dosovitskiy, A., Ros, G., Codella, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning. pp. 1–16 (2017) [3](#), [9](#)
11. El Madawi, K., Rashed, H., El Sallab, A., Nasr, O., Kamel, H., Yogamani, S.: Rgb and lidar fusion based 3d semantic segmentation for autonomous driving. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). pp. 7–12. OPTorganization (2019) [1](#)
12. Fan, X., Zhou, Z., Shi, P., Xin, Y., Zhou, X.: Rafm: Recurrent atrous feature modulation for accurate monocular depth estimating. IEEE Signal Processing Letters pp. 1–5 (2022). <https://doi.org/10.1109/LSP.2022.3189597> [1](#)
13. Fan, Z., Song, Z., Liu, H., Lu, Z., He, J., Du, X.: Svt-net: Super light-weight sparse voxel transformer for large scale place recognition. AAAI (2022) [4](#)
14. Fan, Z., Zhu, Y., He, Y., Sun, Q., Liu, H., He, J.: Deep learning on monocular object pose detection and tracking: A comprehensive overview. ACM Computing Surveys (CSUR) (2021) [1](#)
15. Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. In: International Conference on Machine Learning. pp. 1319–1327. PMLR (2013) [4](#)
16. Guo, Y., Ma, J., Leslie, E., Huang, Z.: Evaluating the effectiveness of integrated connected automated vehicle applications applied to freeway managed lanes. IEEE Transactions on Intelligent Transportation Systems (2020) [20](#)
17. Hu, H., Zhang, Z., Xie, Z., Lin, S.: Local relation networks for image recognition. In: ICCV. pp. 3464–3473 (2019) [8](#), [17](#)

18. Hu, Z., Dong, Y., Wang, K., Sun, Y.: Heterogeneous graph transformer. In: Proceedings of The Web Conference 2020. pp. 2704–2710 (2020) [5](#), [9](#)
19. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: NeurIPS (2015) [6](#), [16](#)
20. Kenney, J.B.: Dedicated short-range communications (dsrc) standards in the united states. Proceedings of the IEEE **99**(7), 1162–1182 (2011) [10](#)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) [10](#)
22. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: CVPR. pp. 12697–12705 (2019) [1](#), [4](#), [5](#), [19](#)
23. Lei, Z., Ren, S., Hu, Y., Zhang, W., Chen, S.: Latency-aware collaborative perception. arXiv preprint arXiv:2207.08560 (2022) [2](#)
24. Li, Y., Ma, D., An, Z., Wang, Z., Zhong, Y., Chen, S., Feng, C.: V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving. IEEE Robotics and Automation Letters (2022) [9](#)
25. Li, Y., Ren, S., Wu, P., Chen, S., Feng, C., Zhang, W.: Learning distilled collaboration graph for multi-agent perception. NeurIPS **34** (2021) [4](#), [10](#), [11](#), [13](#), [21](#), [24](#), [25](#), [26](#)
26. Li, Y., Zhuang, Y., Hu, X., Gao, Z., Hu, J., Chen, L., He, Z., Pei, L., Chen, K., Wang, M., et al.: Toward location-enabled iot (le-iot): Iot positioning techniques, error sources, and error mitigation. IEEE Internet of Things Journal **8**(6), 4035–4062 (2020) [9](#), [11](#)
27. Liang, M., Yang, B., Wang, S., Urtasun, R.: Deep continuous fusion for multi-sensor 3d object detection. In: ECCV. pp. 641–656 (2018) [1](#)
28. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. pp. 2980–2988 (2017) [6](#)
29. Liu, Y.C., Tian, J., Ma, C.Y., Glaser, N., Kuo, C.W., Kira, Z.: Who2com: Collaborative perception via learnable handshake communication. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 6876–6883. IEEE (2020) [15](#)
30. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021) [4](#), [8](#), [17](#), [18](#), [19](#)
31. Mo, Y., Zhang, P., Chen, Z., Ran, B.: A method of vehicle-infrastructure cooperative perception based vehicle state information fusion using improved kalman filter. Multimedia Tools and Applications pp. 1–18 (2021) [4](#)
32. Rauch, A., Klanner, F., Dietmayer, K.: Analysis of v2x communication parameters for the development of a fusion architecture for cooperative perception systems. In: 2011 IEEE Intelligent Vehicles Symposium (IV). pp. 685–690. OPTorganization (2011) [12](#), [22](#)
33. Rauch, A., Klanner, F., Rasshofer, R., Dietmayer, K.: Car2x-based perception in a high-level fusion architecture for cooperative perception systems. In: 2012 IEEE Intelligent Vehicles Symposium. pp. 270–275. OPTorganization (2012) [4](#)
34. Rawashdeh, Z.Y., Wang, Z.: Collaborative automated driving: A machine learning-based method to enhance the accuracy of shared information. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). pp. 3961–3966. OPTorganization (2018) [4](#)
35. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: CVPR. pp. 10529–10538 (2020) [4](#)

36. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: CVPR. pp. 770–779 (2019) [4](#)
37. Treml, M., Arjona-Medina, J., Unterthiner, T., Durgesh, R., Friedmann, F., Schubert, P., Mayr, A., Heusel, M., Hofmarcher, M., Widrich, M., et al.: Speeding up semantic segmentation for autonomous driving. In: NeurIPS workshop MLITS (2016) [1](#)
38. Tsukada, M., Oi, T., Ito, A., Hirata, M., Esaki, H.: Autoc2x: Open-source software to realize v2x cooperative perception among autonomous vehicles. In: 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall). pp. 1–6. OPTorganization (2020) [11](#)
39. Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., Li, Y.: Maxim: Multi-axis mlp for image processing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5769–5780 (2022) [4](#)
40. Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., Li, Y.: Maxvit: Multi-axis vision transformer. arXiv preprint arXiv:2204.01697 (2022) [4](#)
41. Vadivelu, N., Ren, M., Tu, J., Wang, J., Urtasun, R.: Learning to communicate and correct pose errors. arXiv preprint arXiv:2011.05289 (2020) [4](#)
42. Vaswani, A., Ramachandran, P., Srinivas, A., Parmar, N., Hechtman, B., Shlens, J.: Scaling local self-attention for parameter efficient visual backbones. In: CVPR. pp. 12894–12904 (2021) [4](#)
43. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS. pp. 5998–6008 (2017) [4](#)
44. Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A., Chen, L.C.: Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In: European Conference on Computer Vision. pp. 108–126. Springer (2020) [18](#)
45. Wang, T.H., Manivasagam, S., Liang, M., Yang, B., Zeng, W., Urtasun, R.: V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In: ECCV. pp. 605–621. Springer (2020) [2](#), [4](#), [10](#), [11](#), [13](#), [21](#), [24](#), [25](#), [26](#)
46. Wang, Z., Cun, X., Bao, J., Liu, J.: Uformer: A general u-shaped transformer for image restoration. arXiv preprint arXiv:2106.03106 (2021) [4](#)
47. Xia, X., Hang, P., Xu, N., Huang, Y., Xiong, L., Yu, Z.: Advancing estimation accuracy of sideslip angle by fusing vehicle kinematics and dynamics information with fuzzy logic. IEEE Transactions on Vehicular Technology (2021) [9](#), [11](#)
48. Xu, R., Guo, Y., Han, X., Xia, X., Xiang, H., Ma, J.: Opencda: an open cooperative driving automation framework integrated with co-simulation. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). pp. 1155–1162. OPTorganization (2021) [9](#)
49. Xu, R., Tu, Z., Xiang, H., Shao, W., Zhou, B., Ma, J.: Cobevt: Cooperative bird’s eye view semantic segmentation with sparse transformers. arXiv preprint arXiv:2207.02202 (2022) [4](#)
50. Xu, R., Xiang, H., Xia, X., Han, X., Liu, J., Ma, J.: Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. arXiv preprint arXiv:2109.07644 (2021) [2](#), [4](#), [5](#), [9](#), [10](#), [11](#), [13](#), [15](#), [21](#), [24](#), [25](#), [26](#)
51. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018) [4](#)
52. Yang, B., Luo, W., Urtasun, R.: Pixor: Real-time 3d object detection from point clouds. In: CVPR. pp. 7652–7660 (2018) [4](#)
53. Yang, Z., Sun, Y., Liu, S., Shen, X., Jia, J.: Std: Sparse-to-dense 3d object detector for point cloud. In: CVPR. pp. 1951–1960 (2019) [4](#)

54. Yuan, Y., Cheng, H., Sester, M.: Keypoints-based deep feature fusion for cooperative vehicle detection of autonomous driving. *IEEE Robotics and Automation Letters* **7**(2), 3054–3061 (2022) [2](#)
55. Zelin, Z., Ze, W., Yueqing, Z., Boxun, L., Jiaya, J.: Tracking objects as pixel-wise distributions. *arXiv preprint arXiv:2207.05518* (2022) [1](#)
56. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R., et al.: Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955* (2020) [8](#), [17](#)
57. Zhang, Z., Fisac, J.F.: Safe occlusion-aware autonomous driving via game-theoretic active perception. *arXiv preprint arXiv:2105.08169* (2021) [1](#)
58. Zhao, X., Mu, K., Hui, F., Prehofer, C.: A cooperative vehicle-infrastructure based urban driving environment perception method using a ds theory-based credibility map. *Optik* **138**, 407–415 (2017) [4](#)
59. Zhong, Y., Zhu, M., Peng, H.: Vin: Voxel-based implicit network for joint 3d object detection and segmentation for lidars. *arXiv preprint arXiv:2107.02980* (2021) [4](#)
60. Zhou, Y., Tuzel, O.: Voxelnets: End-to-end learning for point cloud based 3d object detection. In: *CVPR*. pp. 4490–4499 (2018) [4](#)
61. Zhou, Z., Fan, X., Shi, P., Xin, Y.: R-msfm: Recurrent multi-scale feature modulation for monocular depth estimating. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 12777–12786 (2021) [1](#)