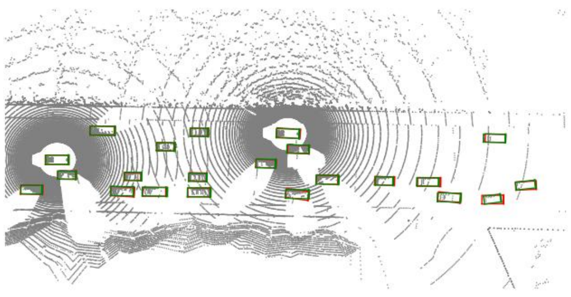
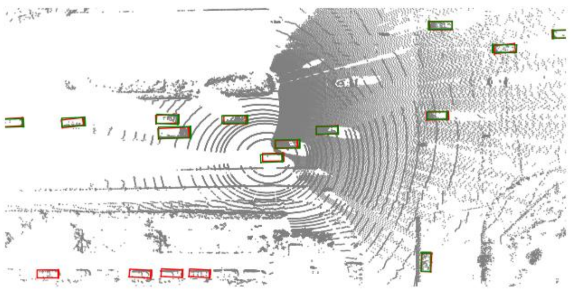
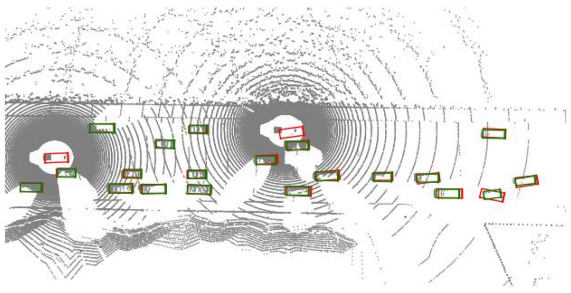
SparseAlign: A Fully Sparse Framework for Cooperative Object Detection

Supplementary Material

A. Qualitative results


(a) OPV2V


(b) DAIR-V2X


(c) OPV2Vt

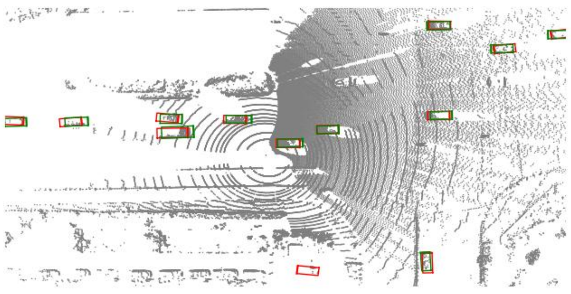

(d) DAIR-V2Xt

Figure 7. Qualitative results of COOD and TA-COOD. GT: green BBox, detection: red BBox.

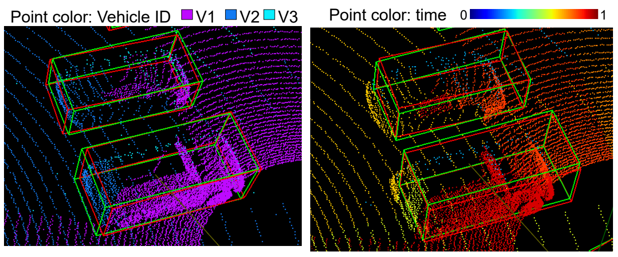


Figure 8. Qualitative result COOD (left) vs. TA-COOD (right).

Figure 7 presents qualitative results for a sample frame from each dataset. The COOD results in Figs. 7a and 7b demonstrate that our framework, SparseAlign, successfully detects most vehicles with high overlap with the ground truth. In contrast, TA-COOD poses a greater challenge, leading to less accurate detections, as shown in Figs. 7c and 7d. However, the model performs better on OPV2Vt than on DAIR-V2Xt. We attribute this to OPV2Vt being a simulated dataset with more precise ground truth, enabling more accurate learning of fine-grained temporal context.

Figure 8 illustrates the differences between COOD and TA-COOD. On the left, point clouds scanned by three IAs are visualized in different colors. The ground truth BBoxes (green) are perfectly aligned with the synchronized scanned points, allowing the model to focus solely on learning the geometric structure of the point clouds for accurate vehicle detection. However, in TA-COOD, a more realistic setting, LiDAR points are typically scanned at different time points rather than being synchronized. In the right image, these asynchronized points are color-coded from blue (earlier scans) to red (later scans). Additionally, the ground truth bounding boxes are aligned to a future time frame rather than the exact geometry of the scanned points. This ground truth encourages the model to leverage temporal context to accurately predict vehicle positions in the near future, compensating for location errors caused by communication latency.

B. Free Space Augmentation (FSA)

Free space augmentation (FSA) was first introduced in GevBEV[42] for BEV map-view semantic segmentation. Typically, point cloud data only capture reflections from obstacle surfaces. For example, in Fig. 9, points A and B represent reflections from the ground surface. With only these measurements, we can confirm the presence of obstacles at these locations but have no information about the space between them. However, distinguishing between empty and unknown spaces is crucial in many applications. For instance, autonomous vehicles can safely navigate through

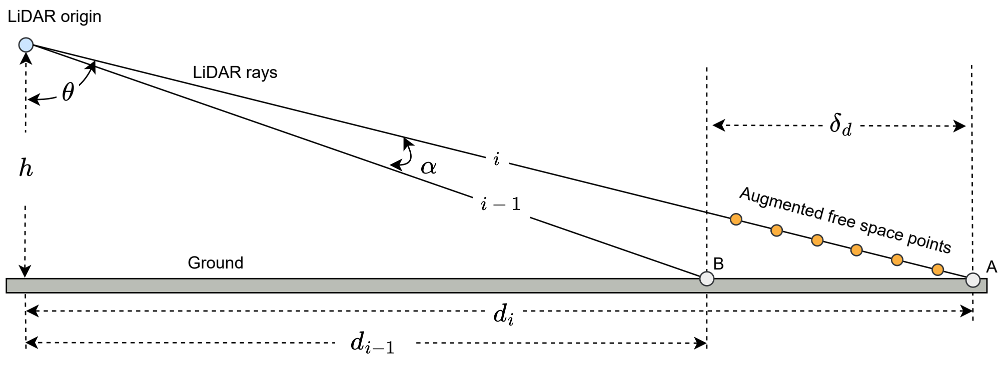


Figure 9. Free space augmentation.

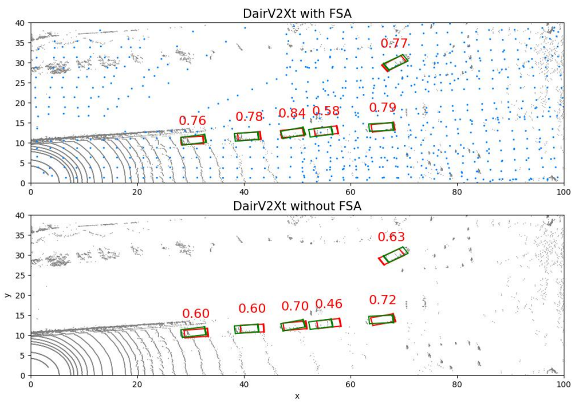


Figure 10. SparseAlign performance with and without FSA. The FSA points are in blue. Red texts are IoUs between the detected (red) and the ground-truth (green) BBoxes.

known empty spaces but must avoid unknown areas where no measurements exist.

Each LiDAR ray provides information not only through its reflection point but also along its entire path, which consists of free space points indicating empty space. Augmenting sparse point clouds with additional points along LiDAR rays helps fill in these gaps. However, adding points along the entire ray path would be computationally prohibitive. Instead, as illustrated by the yellow points in Fig. 9, it is sufficient to augment only the free space points between adjacent laser beams.

Mathematically, given the i-th LiDAR ray with ground distance d_i, LiDAR height relative to the ground h, and the inclination angle difference α between two adjacent rays (e.g., θ for the i-th ray), the gap distance between adjacent rays is calculated as:

$$\delta_d = d_i - d_{i-1} = d_i - h \cdot \tan\left(\arctan\frac{d_i}{h} - \alpha\right) \quad (10)$$

Within this gap distance, we evenly distribute free space points to augment the original point cloud. These points not only convey information about empty spaces but also enhance connectivity between disjoint LiDAR scan rings. This, in turn, improves the convolutional layers' ability to learn spatial context over a larger receptive field.

Figure 10 compares SparseAlign's performance with and without FSA. The results show that FSA slightly improves detection accuracy. Additionally, the augmented free space points (blue) incorporate timestamps computed based on their angles in the polar coordinate system. This enhances temporal context learning, particularly in distant regions where scan observations are sparse.

C. Dilation convolution

In this paper, we demonstrated that *SUNet* effectively mitigates ICF and CFM issues by expanding the receptive field through CEC layers. One might argue that dilated convolutional layers could serve a similar purpose. To investigate this, we conducted an additional experiment, modifying the dilation size of the first convolution layer in each *SUNet* block from one to two. The results indicate that, in this case, dilated convolutions fail to effectively expand the receptive field or mitigate the ICF issue. Instead, they degrade local feature learning.

Dilation size	OPV2V		DAIR-V2X	
	AP0.5	AP0.7	AP0.5	AP0.7
2	0.890	0.827	0.707	0.637
1	0.924	0.885	0.773	0.638

Table 7. Comparison of normal sparse convolutions and dilation convolutions

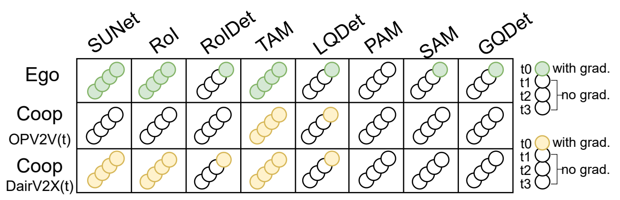


Figure 11. Schedules for gradient calculation.

D. Gradients calculation schedule for efficient training

To optimize training efficiency, we schedule gradient calculations only for essential modules, as illustrated in Fig. 11. $t0$ is the newest frame and $t1$ to $t3$ is the historical frames. For the ego vehicle, we compute gradients for all modules in the latest data frame, except for *PAM*, which is trained independently. Additionally, *SUNet*, *RoI*, and *TAM* compute gradients across all frames to facilitate temporal feature learning.

In the OPV2V and OPV2Vt datasets, each IA is equipped with identical range-view sensors. Consequently, we disable gradient calculations for the first three modules in cooperative IAs, as their data exhibit the same feature patterns as those of the ego IA.

In contrast, in the DAIR-V2X and DAIR-V2Xt datasets, cooperative IAs are infrastructure-based, featuring different sensors and viewing angles from vehicles. To ensure effective learning from infrastructure data, gradient calculations remain enabled for *SUNet*, *RoI*, and *TAM* in cooperative agents.

The final two modules pertain to the fusion process, which occurs exclusively in the ego vehicle. Therefore, no gradient calculations are performed on the cooperative side. Notably, we enable gradients for only one cooperative IA when necessary, irrespective of the total number of IAs.

E. Average precision on 3D metric

	TransIFF[4]	SparseAlign
OPV2V	-	0.922/0.816
DAIR-V2X	0.596/0.460	0.727/0.352
OPV2Vt	-	0.898/0.703
DAIR-V2Xt	-	0.698/0.237

Table 8. AP 3D at IoU thresholds of 0.5/0.7.

In the paper, we used BEV IoU thresholds to calculate the average precision. Here, we also report results based on 3D IoU thresholds for convenient comparison with other methods that use 3D AP as an evaluation metric, such as TransIFF[4]. The results in Tab. 8 indicate that our SparseAlign achieves higher AP at a 3D IoU threshold of 0.5 but lower AP at 0.7. We attribute this to the fact that the three alignment modules in SparseAlign consider only the x and y coordinates in geometric operations during the fusion process, without explicitly accounting for the z-axis.