Communication-Efficient Collaborative Perception via Information Filling with Codebook

Supplementary Material

Here, we start with module design details and give out more implementation details, then introduce the dataset details, including generation and qualitative samples, and the exact values for the benchmarks.

1. Detailed Information About Module Designs

1.1. Information-filling-driven message selection

Solution. Alg. 1 presents the solution of our information-filling-driven message selection, this is,

$$\{\mathbf{M}^*_{i\to j}\}_{i,j} = \underset{\mathbf{M}}{\operatorname{argmax}} \sum_{j=1}^{N} f_{\min}\left(\mathbf{C}_j + \sum_{i=1,i\neq j}^{N} \mathbf{M}_{i\to j}\odot \mathbf{C}_i, u\right), \tag{1a}$$

$$\text{where} \sum_{i,j=1,j\neq i}^{N} \mathbf{M}_{i\to j} \leq b, \mathbf{M}_{i\to j} \in \{0,1\}^{H\times W}. \tag{1b}$$

Here, $\mathbf{M}_{i\to j} \in \{0,1\}^{H\times W}$ is the binary selection matrix supported on the BEV map. Each element in the matrix indicates whether Agent i should send the information to Agent j at a specific spatial location (1 for sending information, and 0 for not sending). $\odot$ denotes element-wise multiplication, and the scalar u is a hyper-parameter to reflect the upper bound of information demand. The function $f_{\min}(\cdot,\cdot)$ computes the element-wise minimum between a matrix and a scalar.

Despite the hard constraints and non-differentiability of binary variables in this proxy-constrained optimization problem, it possesses an analytical solution. We tackle this by splitting the optimization into two sub-problems: i) optimizing the maximization in Equation (1a) without the constraint in Equation (1b) and removing the indifferentiable thresholding function $f_{\min}(\cdot)$; ii) addressing the equivalent maximization problem of Equation (1a) while considering the constraint in Equation (1b).

• The first sub-problem involves unconstrained maximization optimization, which is given by

$$\{\overline{\mathbf{M}}_{i\to j}\}_{i,j} = \underset{\mathbf{M}}{\operatorname{argmax}} \sum_{j=1}^{N} f_{\min}\left(\mathbf{C}_j + \sum_{i=1,i\neq j}^{N} \mathbf{M}_{i\to j}\odot \mathbf{C}_i, u\right), \tag{2a}$$

$$\text{where } \mathbf{M}_{i\to j} \in \{0,1\}^{H\times W}. \tag{2b}$$

This involves selecting the highest-scoring regions to meet the information demand, excluding unnecessary information for each sender-receiver pair, resulting in $\overline{\mathbf{M}}_{i\to j}$. Steps include: a) Sorting scores from all collaborators in descending order for each spatial location; b) Accumulating these scores until reaching the information demand threshold and disregarding the rest, refining the subset.

By doing so, we can remove the indiscernible cutoff by using the optimized matrix $\overline{\mathbf{M}}_{i\to j}$ to focus on required information scores, this is,

$$f_{\min}\left(\mathbf{C}_j + \sum_{i=1,i\neq j}^{N} \mathbf{M}_{i\to j}\odot \mathbf{C}_i, u\right) = \mathbf{C}_j + \overline{\mathbf{M}}_{i\to j}\odot \mathbf{C}_i, \tag{3a}$$

$$= \mathbf{C}_j + \mathbf{C}_{i\to j}. \tag{3b}$$

• The second sub-problem is a proxy-constrained maximization optimization without an indiscernible cutoff. By substituting Equation (3b) into Equation (1a), we get an equivalent formulation of the original optimization in Equation (1a),

$$\{\mathbf{M}^*_{i\to j}\}_{i,j} = \underset{\mathbf{M}}{\operatorname{argmax}} \sum_{j=1}^{N} \sum_{i=1,i\neq j}^{N} \mathbf{M}_{i\to j}\odot \mathbf{C}_{i\to j}, \tag{4a}$$

$$\text{where} \sum_{i,j=1,j\neq i}^{N} \mathbf{M}_{i\to j} \leq b, \mathbf{M}_{i\to j} \in \{0,1\}^{H\times W}. \tag{4b}$$

This optimization has an analytical solution, which involves selecting top-b ranked spatial regions based on elements in $\mathbf{M}$. The steps are: c) Resorting all retained scores across spatial regions in descending order; d) Forming $\mathbf{M}^*$ by marking top-b elements in this list as 1, others as 0.

Note that, information demand is fulfilled in b), communication constraint is met in d), and maximization is achieved through prioritization in a) and c). Collectively, these steps yield an optimal solution for the constrained optimization problem in Equation (1a) and Equation (1b).

Computation cost. Step c is the most computationally demanding, involving the sorting of all necessary spatial regions to meet the information demand. However, in our scenario, the precise order is irrelevant; we only need to identify the top-b elements from m spatial region candidates, resulting in a computational cost of $O(\log(m))$. By concentrating on the highly sparse foreground areas, we significantly lower this cost to a negligible level, thus allowing each agent to offer more focused support to others at minimal expense.

1.2. Codebook-based message representation

Extensibility for new heterogeneous agents. The codebook representation creates a common feature space that enables the integration of new heterogeneous agents. In the training phase, perceptual features from all agents, whether equipped with camera or LiDAR sensors, are collected in F for codebook training. This process benefits from joint

Algorithm 1 Information-filling-driven Message Selection

Require: Spatial information score maps $\{\mathbf{C}_i\}_{i=1}^{N}$ of N agents with dimensions (H, W), information demand u, communication budget b.

Ensure: Selection matrices $\{\mathbf{M}_{i\to j}\}_{j=1,j\neq i}^{N}$ for each agent pair (i, j).

# Select the required information to fulfill the receiver's information demand from high-scoring senders per-location
# Initialization
for all $i \in \{1, \ldots, N\}, j \in \{1, \ldots, N\}$ **do**
    $\overline{\mathbf{M}}_{i\to j} = \mathbf{0} \in \{0, 1\}^{H\times W}$
end for
for all $j \in \{1, \ldots, N\}$ **do** ▷ Receiver
    for $h \in \{0, \ldots, H-1\}, w \in \{0, \ldots, W-1\}$ **do** ▷ Per-location
        # Step a: Prioritize senders with higher scores
        $R = f_{\text{rank}}(\{\mathbf{C}_i[h, w]\}_{i=1,i\neq j}^{N})$ ▷ Senders
        $s = \mathbf{C}_j[h, w]$ ▷ The receiver's initial information amount
        # Step b: Exclude information over information demand
        $A = []$ ▷ The selected senders per-location
        for each $\mathbf{C}_i[h, w]$ in R **do**
            if $s \leq u$ **then** ▷ Check the whether the information demand is reached
                # Select sender
                $s = s + \mathbf{C}_i[h, w]$
                Append i to A
            else
                # Stop selection once demand is met
                break
            end if
        end for
        # Select the required regions whose accumulated information below information demand
        for all $i \in A$ **do**
            $\overline{\mathbf{M}}_{i\to j}[h, w] = 1$
        end for
    end for
    # Exclude information over demand
    for all $i \in \{1, \ldots, N\} \setminus \{j\}$ **do**
        $\mathbf{C}_{i\to j} = \mathbf{C}_i \odot \overline{\mathbf{M}}_{i\to j}$
    end for
end for
# Select the most beneficial information within the communication budget among all the needed spatial regions
# Initialization
for all $i \in \{1, \ldots, N\}, j \in \{1, \ldots, N\}$ **do**
    $\mathbf{M}_{i\to j} = \mathbf{0} \in \{0, 1\}^{H\times W}$
end for
 # Step c: Prioritize information with higher scores
$R \leftarrow f_{\text{rank}}(\{\mathbf{C}_{i\to j}\}_{i,j=1,i\neq j}^{N})$ ▷ All the required spatial regions between all the sender-receiver pairs
 # Step d: Exclude information over communication budget
for all $j \in \{1, \ldots, N\}$ **do** ▷ Receiver
    for all $i \in \{1, \ldots, N\} \setminus \{j\}$ **do** ▷ Sender
        for $h \in \{0, \ldots, H-1\}, w \in \{0, \ldots, W-1\}$ **do** ▷ Per-location
            if $\mathbf{C}_{i\to j}[h, w]$ is in top-b of R **then**
                $\mathbf{M}_{i\to j}[h, w] = 1$
            end if
        end for
    end for
end for
return $\{\mathbf{M}_{j\to i}\}_{i,j=1,i\neq j}^{N}$

supervision using diverse inputs, enhancing learning efficiency and ensuring that critical perceptual information is retained. As a result, the optimized task-adaptive codebook $\mathbf{D}^*$ encapsulates the essential features from various modalities. During inference, all agents utilize this optimal codebook $\mathbf{D}^*$ directly.

Adaptability for codebook configuration. The codebook's configuration is highly adaptable, allowing for adjustments in both the size of the codebook n_L and the number of codes n_R utilized for representing the input vector. During training, we vary the code quantity from 1 to n_R, enabling the optimized codebook to accommodate different configurations and communication budgets during inference.

During training, especially with an increased number of codes, the representation comprises combinations of multiple codes. Consequently, task-driven codebook learning entails the aggregation of these codes for feature approximation at each spatial location. This process is defined as follows,

$$\mathbf{D}^* = \arg\min_{\mathbf{D}} \sum_{\mathcal{F}\in F} \sum_{h,w} \min_{\{\ell_r\}_{r=1}^{n_R}} \left(\Psi(\mathbf{F}_d) + \left\| \mathcal{F}_{[h,w]} - \mathbf{F}_d \right\|_2^2 \right), \tag{5a}$$

$$\text{where } \mathbf{F}_d = \sum_{\ell_r \in \mathcal{L}_{n_R}} \mathbf{D}_{[\ell_r]}, \mathcal{L}_{n_R} = \{\ell_r\}_{r=1}^{n_R}. \tag{5b}$$

Here, $\mathbf{F}_d$ is a combination of codes $\{\mathbf{D}_{[\ell_r]}\}_{r=1}^{n_R}$, and $\Psi(\cdot)$ measures the detection performance achieved by replacing $\mathcal{F}_{[h,w]}$ with $\mathbf{F}_d$. The code index set $\mathcal{L}_{n_R} = \{\ell_r\}_{r=1}^{n_R}$ is selected to minimize the reconstruction error in a greedy way. As n_r ranges from 1 to n_R, the optimization of the code index is carried out as follows

$$l_r^* = \min_{\mathcal{L}_{n_{r-1}} \cup \{l_r\}} \left\| \mathcal{F}_{[h,w]} - \sum_{\ell_k \in \mathcal{L}_{n_{r-1}} \cup \{l_r\}} \mathbf{D}_{[\ell_k]} \right\|_2^2. \tag{6}$$

In this process, the code index l_r^* is determined by minimizing the reconstruction error, which is the L2 norm of the difference between the feature vector $\mathcal{F}_{[h,w]}$ and the sum of the selected codes from the codebook $\mathbf{D}$. The selection at each step involves the union of the set $\mathcal{L}_{n_{r-1}}$, representing the previously selected indices, and the new index l_r^*.

During inference, each agent leverages the optimized codebook $\mathbf{D}$ to convert the selected sparse feature map $\mathcal{Z}_{i\to j}$ into code indices $\mathcal{I}_{i\to j}$. At each Bird's Eye View (BEV) location (h, w), given a code quantity of n_r, the code index is obtained as follows,

$$(\mathcal{I}_{i\to j})_{[h,w]} = \arg\min_{\mathcal{L}_{n_r}} \left\| (\mathcal{Z}_{i\to j})_{[h,w]} - \sum_{\ell_k \in \mathcal{L}_{n_r}} \mathbf{D}_{[\ell_k]} \right\|_2^2. \tag{7}$$

This method aligns with Equation (6) to create the optimized set of code indices $\mathcal{L}_{n_r} = \{\ell_k\}_{k=1}^{n_r}$. The value of n_r, ranging from 1 to n_R, allows the codebook to be flexible for different configurations and communication requirements during deployment.

2. Additional Experimental Results

2.1. Robustness assessment under more metrics

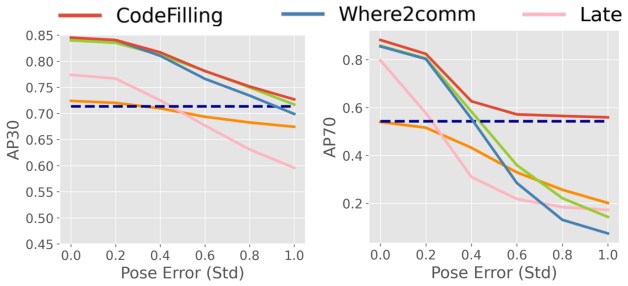
We validate the robustness against pose error and communication latency on both OPV2VH+ and DAIR-V2X under camera-only and heterogeneous settings. The pose error setting follows CoAlign [8] using Gaussian noise with a mean of 0m and standard deviations ranging from 0m to 1.0m. The latency setting follows SyncNet [5], varying from 0ms to 500ms. Figs. 1 and 2 show the detection performances as a function of pose error and latency, respectively in terms of AP30 and AP70. We see: i) while perception performance generally declines with increasing levels of pose error and latency, `CodeFilling` consistently outperforms baselines under all imperfect conditions; ii) `CodeFilling` consistently surpasses No Collaboration, whereas baselines fail when pose error exceeds 0.4m and latency surpasses 100ms. In `CodeFilling`, setting a lower information demand u in situations with pose errors and latency issues allows each agent to collect less misleading collaborative information, thereby at least maintaining their individual performance.

2.2. Discussion on the realistic limitations

There are many challenges in a collaborative perception system. In this work, we focus on the bottleneck challenge in current collaborative perception systems; that is, the trade-off between communication bandwidth and perception performance. This challenge has been actively addressed in previous works [2, 3, 6, 7]. Collaborative perception is enabled and also severely limited by the communication capacity, which is critically reflected in the highly dynamic and limited bandwidth in real-world communication systems. `CodeFilling` flexibly adapts to various communication bandwidths, achieving superior performance-bandwidth trade-off.

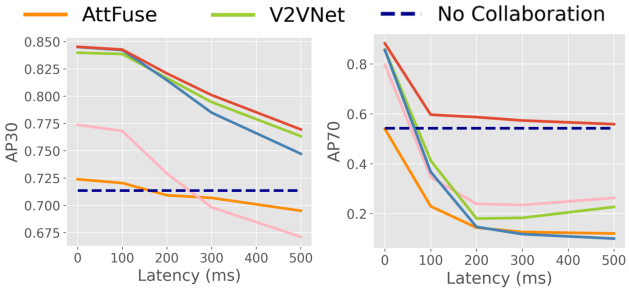
Here we further discuss other realistic limitations, assess the robustness of our system, and future improvements to be made.

- For other realistic communication issues such as **latency**, **time synchronization**, **pose error**, **attack**, `CodeFilling` communicates strategically when necessary, rather than all the time or everywhere, to reduce the possibility of encountering communication problems. And `CodeFilling` can set a lower information demand u in situations with these issues, which allows each agent to collect less misleading collaborative information, thereby at least maintaining their individual performance.
- For the **data availability**, `CodeFilling` works on both RGB and point cloud modalities and is sensor-friendly, so it can be deployed on cheap camera sensors and lidar sensors. And it accommodates heterogeneous settings where agents with different equipment can also collaborate with



(b) DAIR-V2X (a) OPV2VH+

Figure 1. `CodeFilling` is robust to pose error issue.



(b) DAIR-V2X (a) OPV2VH+

Figure 2. `CodeFilling` is robust to communication latency issue.

each other.

2.3. Experimental settings

In our system, for LiDAR sensor inputs, we adopt the PointPillar detector [4], while for camera inputs, we follow the CaDDN [9], utilizing 50 depth categories with linearly increasing spacing. To enhance learning effectiveness, we train all models in a heterogeneous setting. Consequently, in the inference phase, this model becomes versatile and applicable in homogeneous and heterogeneous settings, including camera-only, LiDAR-only, and heterogeneous setups.

For the training strategy, we initially pre-train the single-agent detector without a codebook for 30 epochs, starting with a learning rate of 2e-3 and reducing it by a factor of 0.1 at the 20th epoch. This phase establishes a robust perceptual feature space. Subsequently, we train the entire collaborative perception model for 20 epochs, integrating both codebook reconstruction and perception losses. This dual supervision not only boosts learning efficiency but also ensures the codebook retains essential perceptual features, enabling a lossless performance for the perceptual task.

3. OPV2VH+ Dataset

Data generation. We extend the original OPV2V [11] with more collaborative agents (10), and extend the OPV2V+ [3] with more modalities. Our OPV2VH+ is co-simulated by OpenCDA [10] and CARLA [1]. OpenCDA provides the driving scenarios that ensure the agents drive smoothly and safely, including the vehicle's initial location and moving speed. CARLA provides the maps, and weather and controls the movements of the agents. We replay the simulation logs of OPV2V and equip more vehicles with LiDAR, camera and depth sensors. Figure. 3 shows the LiDAR and four RGB/depth camera views (front, left, right, back) of the same agent. Figure. 4 and Figure. 5 show a randomly selected data sample with 10 collaborative agents, the collected LiDAR and front view images in the same timestamp.
Data collection. We collect synchronous images from all 4 cameras, 4 depth sensors, and 1 LiDAR sensor on all the collaborative vehicles in a sample. LiDAR extrinsic, camera/depth sensor intrinsics, and extrinsic in global coordinates are provided to support coordinate transformation across various collaborative vehicles. During data collection, 3D bounding boxes of vehicles in the scene are recorded at the same moment with sensor inputs, including location (x, y, z), rotation (w, x, y, z in quaternion) in the global coordinate, and their length, width, and height. The location (x, y, z) is the center of the bounding box. In total, 10,416 samples, 10,4160 point cloud sweeps, 416,640 RGB/depth images, and 482,037 3D bounding boxes are collected.

Data usage. We randomly split the samples into train/validation/test, resulting 6736/1980/1700 samples, 67,360/19,800/17,000 LiDAR sweeps, 269,400/79,200/68,000 images, and 333,543/75,289/73,205 3D bounding boxes. The dataset is organized in a similar way to the OPV2V [11] and OPV2V+ [3] dataset; so it can be used directly with the original dataset processing tool-kits.

4. Benchmarks

We conduct extensive experiments on all two widely used collaborative perception benchmarks covering three types of collaboration settings: i) all the collaborative agents use cameras, ii) all the collaborative agents use LiDARs, and iii) the collaborative agents randomly use camera or LiDAR. Regarding the heterogeneous setup, agents are randomly assigned either LiDAR or camera, resulting in a balanced 1:1 ratio of agents across the different modalities.

Tab. 2 presents the overall performance on the real-world dataset, DAIR-V2X [12], and the extended simulation dataset OPV2VH+. We see that `CodeFilling` consistently achieves significant improvements over previous methods on all the benchmarks.

Tab. 4 and Tab. 6 presents the overall performance under realistic issues on the real-world dataset, DAIR-V2X [12], and the extended simulation dataset OPV2VH+. We see that `CodeFilling` is more robust to the pose error and communication latency issues.

Table 1. Overall performance on DAIR-V2X. The communication cost is denoted as B.

Dataset	DAIR-V2X					
Setting	LiDAR		Camera		Heterogeneous	
Method/Metric	B	AP@30/50	B	AP@30/50	B	AP@30/50
No Collaboration	0.00	71.35/67.27	0.00	5.65/1.93	0.00	5.54/1.92
Late	19.43	77.40/69.54	19.43	15.82/6.59	19.43	40.82/25.14
AttFuse	22.62	72.38/64.83	22.62	2.63/0.63	22.62	12.93/3.71
DiscoNet	22.62	82.16/78.60	22.62	6.43/1.78	22.62	28.88/16.15
V2VNet	22.62	83.98/79.28	22.62	19.81/7.38	22.62	47.14/28.47
HMViT	22.62	77.09/69.92	22.62	6.65/1.40	22.62	39.78/20.83
Where2comm	0.00	71.35/67.27	0.00	5.65/1.93	0.00	5.54/ 1.92
	13.85	82.39/77.07	13.70	18.47/7.36	13.88	38.73/22.30
	14.94	83.73/78.48	15.59	19.59/7.86	14.95	43.39/26.29
	19.42	84.50/79.24	21.59	21.90/8.32	19.40	47.47/29.73
	22.62	84.50/79.39	22.62	21.96/8.34	22.62	47.47/30.00
CodeFilling	0.00	71.35/67.27	0.00	5.65/1.93	0.00	5.54/ 1.92
	4.96	79.80/75.60	4.88	15.30/6.03	5.00	33.83/19.19
	6.98	82.52/77.73	6.86	18.30/7.10	6.09	39.45/23.72
	8.09	84.00/79.39	8.68	19.63/7.83	8.11	42.61/25.43
	12.12	84.47/79.91	14.41	22.01/8.50	12.09	46.89/28.69
	15.62	84.52/79.99	15.62	22.22/8.51	15.62	47.51/29.14
	22.26	**84.52/79.99**	22.26	**22.22/8.51**	22.26	**47.51/30.00**

Table 2. Overall performance on OPV2VH+. The communication cost is denoted as B.

Dataset	OPV2VH+					
Setting	LiDAR		Camera		Heterogeneous	
Method/Metric	B	AP@50/70	B	AP@50/70	B	AP@50/70
No Collaboration	0.00	68.83/54.27	0.00	15.43/4.97	0.00	44.20/30.57
Late	23.08	86.57/79.51	23.08	51.37/27.71	23.08	78.55/66.02
AttFuse	26.27	76.55/53.95	26.27	28.08/10.18	26.27	63.21/43.51
DiscoNet	26.27	80.13/61.15	26.27	35.16/12.42	26.27	67.59/47.17
V2VNet	26.27	90.06/85.73	26.27	64.18/43.40	26.27	87.96/80.13
HMViT	26.27	89.55/80.79	26.27	57.05/26.19	26.27	86.89/71.61
Where2comm	0.00	68.54/54.04	0.00	15.29/ 4.92	0.00	43.75/30.27
	17.25	85.39/79.44	16.49	50.16/25.94	16.90	77.16/63.60
	18.77	89.71/85.14	18.57	58.99/38.75	18.66	85.33/78.39
	22.63	90.76/85.36	25.81	65.05/47.66	25.01	88.19/83.31
	26.27	90.77/85.53	26.27	67.28/48.90	26.27	88.42/79.63
CodeFilling	0.00	68.54/54.04	0.00	15.29/4.92	0.00	43.75/30.27
	8.10	85.65/75.14	6.26	35.40/13.69	6.41	65.15/46.61
	9.27	87.38/77.90	8.40	43.79/20.01	8.30	75.47/56.74
	13.22	89.90/85.79	12.29	55.11/31.59	12.21	82.90/71.88
	16.04	90.53/86.94	18.92	63.89/43.78	18.15	87.43/80.45
	19.51	90.66/86.99	19.51	66.39/47.77	19.51	88.05/80.98
	25.60	**90.82/88.19**	25.61	**67.39/51.16**	25.61	**88.58/83.65**

Table 3. Robustness to pose error on DAIR-V2X.

Dataset	DAIR-V2X				
Method/Metric	AP30/AP50↑				
Noise Level $\sigma_t/\sigma_r(m/^\circ)$	0.0/0.0	0.2/0.2	0.4/0.4	0.6/0.6	1.0/1.0
No Collaboration	71.35/67.27	71.35 /67.27	71.35 /67.27	71.35 /67.27	71.35 /67.27
Late	77.39/69.53	76.66 /67.04	72.49 /59.92	67.67 /54.89	59.57 /50.17
AttFuse	72.40/64.87	72.01 /64.22	70.96 /62.89	69.36 /61.82	67.43 /60.98
V2VNet	83.98/79.28	83.51 /77.13	81.19 /71.88	75.15 /67.92	71.69 /62.74
Where2comm	84.50/79.39	84.03 /77.12	81.02 /70.31	76.62 /64.92	69.89 /59.10
CodeFilling	**84.52/79.99**	**84.05 /78.05**	**81.69 /72.58**	**78.10 /68.47**	**72.68 /67.99**

Table 4. Robustness to pose error on OPV2VH+.

Dataset	OPV2VH+				
Method/Metric	AP50/AP70↑				
Noise Level $\sigma_t/\sigma_r(m/^\circ)$	0.0/0.0	0.2/0.2	0.4/0.4	0.6/0.6	1.0/1.0
No Collaboration	68.83 /54.27	68.83 /54.27	68.83 /54.27	68.83 /54.27	68.83 /54.27
Late	86.86 /79.70	85.24 /57.63	63.86 /31.02	44.13 /21.83	28.90 /17.26
AttFuse	76.56 /53.97	75.98 /51.58	72.95 /43.20	66.19 /32.93	49.08 /20.15
V2VNet	90.06 /85.76	89.74 /80.52	86.24 /58.30	74.63 /35.93	44.59 /14.34
Where2comm	90.77 /85.53	90.20 /80.23	83.33 /55.33	68.52 /28.48	29.07 /7.44
CodeFilling	**90.82 /88.19**	**90.29 /82.38**	**84.23 /62.61**	**74.28 /57.12**	**72.53 /55.90**

Table 5. Robustness to communication latency on DAIR-V2X.

Dataset	DAIR-V2X				
Method/Metric	AP30/AP50↑				
Latency Level (ms)	0	100	200	300	500
No Collaboration	71.35 /67.27	71.35 /67.27	71.35 /67.27		
Late	77.37 /69.53	76.81 /66.74	72.91 /61.85	69.82 /60.46	67.11 /60.18
AttFuse	72.39 /64.83	72.04 /64.03	70.92 /63.40	70.68 /63.35	69.50 /62.84
V2VNet	83.98 /79.28	83.86 /77.75	81.65 /74.52	79.46 /72.51	76.33 /70.57
Where2comm	84.50 /79.39	84.22 /77.34	81.46 /72.69	78.49 /70.62	74.72 /68.92
CodeFilling	**84.52 /79.99**	**84.27 /78.34**	**82.08 /74.73**	**80.10 /72.83**	**76.95 /71.50**

Table 6. Robustness to communication latency on OPV2VH+.

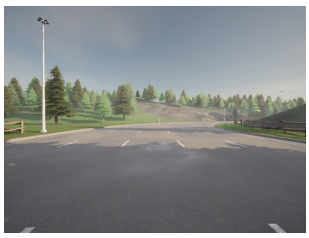
Dataset	OPV2VH+				
Method/Metric	AP50/AP70↑				
Latency Level (ms)	0	100	200	300	500
No Collaboration	68.83 /54.27	68.83 /54.27	68.83 /54.27	68.83 /54.27	68.83 /54.27
Late	86.87 /79.68	75.85 /34.62	39.17 /23.91	32.04 /23.46	34.18 /26.30
AttFuse	76.55 /53.95	64.90 /22.95	34.92 /14.38	24.75 /12.63	20.62 /12.03
V2VNet	90.07 /85.73	83.03 /41.23	46.97 /18.04	31.56 /18.30	29.17 /22.68
Where2comm	90.77 /85.53	83.31 /36.74	40.10 /14.69	21.73 /11.78	14.03 /10.00
CodeFilling	**90.82 /88.19**	**85.29 /59.63**	**75.16 /58.67**	**73.69 /57.31**	**71.53 /55.84**

(a) LiDAR

(b) Camera 0

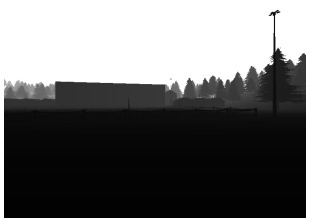
(c) Camera 1

(d) Camera 2

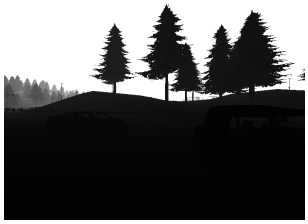


(e) Camera 3

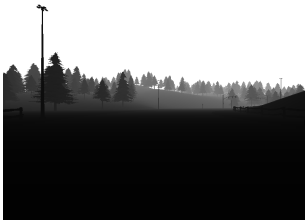
(f) Depth 0



(g) Depth 1



(h) Depth 2



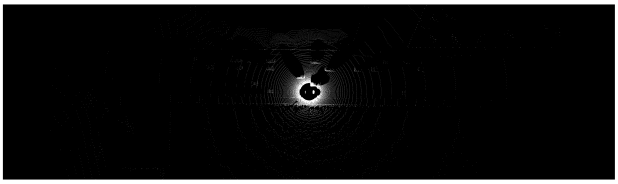
(i) Depth 3

Figure 3. Each agent is equipped with 1 LiDAR, 4 cameras, and 4 depth sensors in OPV2VH+.

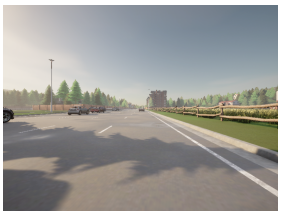
(a) Agent 0: LiDAR 0

(b) Agent 0: Camera 0

(c) Agent 0: Depth 0

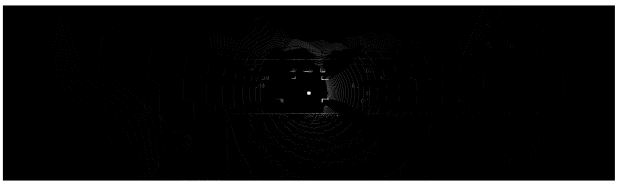


(d) Agent 1: LiDAR 0



(e) Agent 1: Camera 0

(f) Agent 1: Depth 0



(g) Agent 2: LiDAR 0

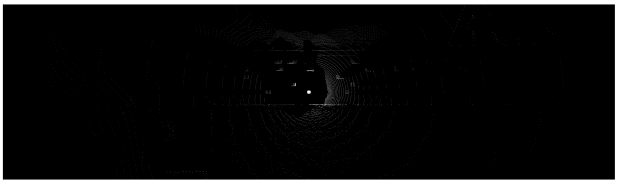
(h) Agent 2: Camera 0

(i) Agent 2: Depth 0

(j) Agent 3: LiDAR 0

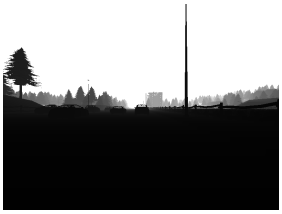
(k) Agent 3: Camera 0

(l) Agent 3: Depth 0



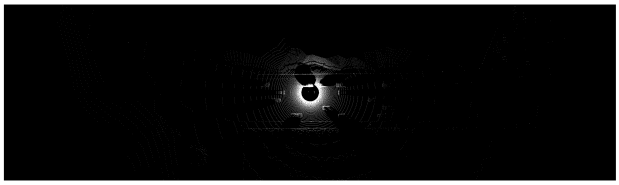
(m) Agent 4: LiDAR 0

(n) Agent 4: Camera 0



(o) Agent 4: Depth 0

Figure 4. Agents 0 through 4 in a data sample comprising 10 agents from the OPV2VH+ dataset.



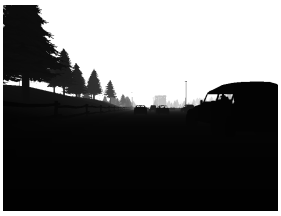
(a) Agent 5: LiDAR 0

(b) Agent 5: Camera 0

(c) Agent 5: Depth 0

(d) Agent 6: LiDAR 0

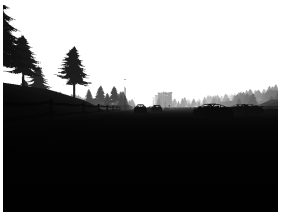
(e) Agent 6: Camera 0



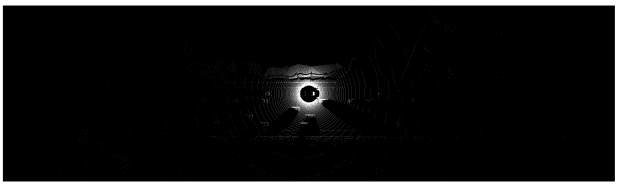
(f) Agent 6: Depth 0

(g) Agent 7: LiDAR 0

(h) Agent 7: Camera 0



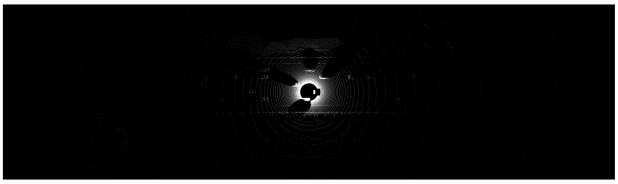
(i) Agent 7: Depth 0



(j) Agent 8: LiDAR 0

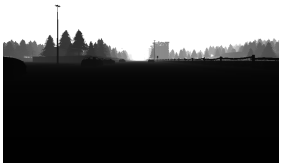
(k) Agent 8: Camera 0

(l) Agent 8: Depth 0



(m) Agent 9: LiDAR 0

(n) Agent 9: Camera 0



(o) Agent 9: Depth 0

Figure 5. Agents 5 through 9 in a data sample comprising 10 agents from the OPV2VH+ dataset.

References

[1] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio M. López, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on Robot Learning*, 2017. 4

[2] 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

[3] Yue Hu, Yifan Lu, Runsheng Xu, Weidi Xie, Siheng Chen, and Yanfeng Wang. Collaboration helps camera overtake lidar in 3d detection. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3, 4

[4] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12689–12697, 2018. 4

[5] Zixing Lei, Shunli Ren, Yue Hu, Wenjun Zhang, and Siheng Chen. Latency-aware collaborative perception. *ECCV*, 2022. 3

[6] Yen-Cheng Liu, Junjiao Tian, Nathaniel Glaser, and Zsolt Kira. When2com: Multi-agent perception via communication graph grouping. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 4106–4115, 2020. 3

[7] Yen-Cheng Liu, Junjiao Tian, Chih-Yao Ma, Nathan Glaser, Chia-Wen Kuo, and Zsolt Kira. Who2com: Collaborative perception via learnable handshake communication. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6876–6883. IEEE, 2020. 3

[8] Yifan Lu, Quanhao Li, Baoan Liu, Mehrdad Dianat, Chen Feng, Siheng Chen, and Yanfeng Wang. Robust collaborative 3d object detection in presence of pose errors. *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 3

[9] Cody Reading, Ali Harakeh, Julia Chae, and Steven L. Waslander. Categorical depth distribution network for monocular 3d object detection. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8551–8560, 2021. 4

[10] Runsheng Xu, Yi Guo, Xu Han, Xin Xia, Hao Xiang, and Jiaqi Ma. Opencda: An open cooperative driving automation framework integrated with co-simulation. *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1155–1162, 2021. 4

[11] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Liu, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. *2022 International Conference on Robotics and Automation (ICRA)*, pages 2583–2589, 2021. 4

[12] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, et al. DAIR-V2X: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. *In Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition (CVPR)*, 2022. 4