Where2comm: Communication-Efficient Collaborative Perception via Spatial Confidence Maps

Yue Hu **Shaoheng Fang** **Zixing Lei**
Cooperative Medianet Innovation Center, Shanghai Jiao Tong University
{18671129361, shfang, chezacarss}@sjtu.edu.cn

Yiqi Zhong
University of Southern California
yiqizhon@usc.edu

Siheng Chen*
Shanghai Jiao Tong University, Shanghai AI Laboratory
sihengc@sjtu.edu.cn

Abstract

Multi-agent collaborative perception could significantly upgrade the perception performance by enabling agents to share complementary information with each other through communication. It inevitably results in a fundamental trade-off between perception performance and communication bandwidth. To tackle this bottleneck issue, we propose a spatial confidence map, which reflects the spatial heterogeneity of perceptual information. It empowers agents to only share spatially sparse, yet perceptually critical information, contributing to where to communicate. Based on this novel spatial confidence map, we propose `Where2comm`, a communication-efficient collaborative perception framework. `Where2comm` has two distinct advantages: i) it considers pragmatic compression and uses less communication to achieve higher perception performance by focusing on perceptually critical areas; and ii) it can handle varying communication bandwidth by dynamically adjusting spatial areas involved in communication. To evaluate `Where2comm`, we consider 3D object detection in both real-world and simulation scenarios with two modalities (camera/LiDAR) and two agent types (cars/drones) on four datasets: OPV2V, V2X-Sim, DAIR-V2X, and our original CoPerception-UAVs. `Where2comm` consistently outperforms previous methods; for example, it achieves more than $100,000\times$ lower communication volume and still outperforms DiscoNet and V2X-ViT on OPV2V. Our code is available at `https://github.com/MediaBrain-SJTU/where2comm`.

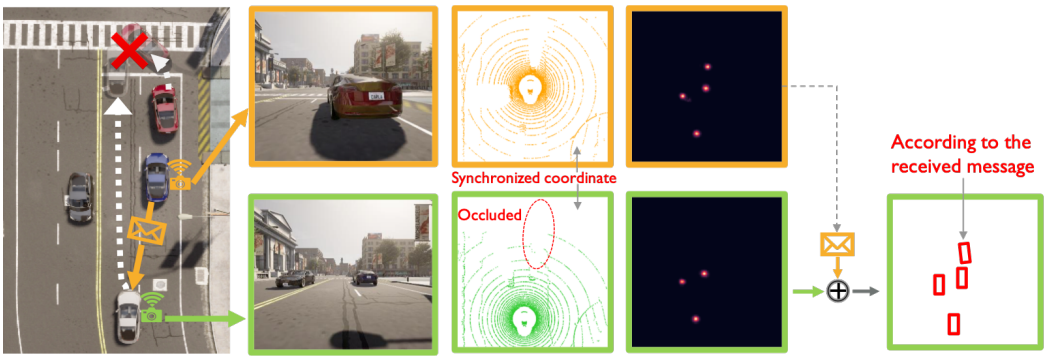
1 Introduction

Collaborative perception enables multiple agents to share complementary perceptual information with each other, promoting more holistic perception. It provides a new direction to fundamentally overcome a number of inevitable limitations of single-agent perception, such as occlusion and long-range issues. Related methods and systems are desperately needed in a broad range of real-world applications, such as vehicle-to-everything-communication-aided autonomous driving [1–3], multi-robot warehouse automation system [4, 5] and multi-UAVs (unmanned aerial vehicles) for search and rescue [6–8]. To realize collaborative perception, recent works have contributed high-quality datasets [9–11] and effective collaboration methods [12, 13, 2, 14–19].

In this emerging field, the current biggest challenge is how to optimize the trade-off between perception performance and communication bandwidth. Communication systems in real-world scenarios are always constrained that they can hardly afford huge communication consumption in real-time, such as passing complete raw observations or a large volume of features. Therefore,

*Corresponding author

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) Scene in BEV (b) Image view (c) LiDAR view (d) Confidence in BEV (e) Detection in BEV

Figure 1: Collaborative perception could contribute to safety-critical scenarios, where the white car and the red car may collide due to occlusion. This collision could be avoided when the blue car can share a message about the red car's position. Such a message is spatially sparse, yet perceptually critical. Considering the precious communication bandwidth, each agent needs to speak to the point!

we cannot solely promote the perception performance without evaluating the expense of every bit of precious communication bandwidth. To achieve a better performance and bandwidth trade-off, previous works put forth solutions from several perspectives. For example, When2com [12] considers a handshake mechanism which selects the most relevant collaborators; V2VNet [1] considers end-to-end-learning-based source coding; and DiscoNet [2] uses 1D convolution to compress message. However, all previous works make a plausible assumption: once two agents collaborate, they are obligated to share perceptual information of all spatial areas *equally*. This unnecessary assumption can hugely waste the bandwidth as a large proportion of spatial areas may contain irrelevant information for perception task. Figure 1 illustrates such a spatial heterogeneity of perceptual information.

To fill this gap, we consider a novel spatial-confidence-aware communication strategy. The core idea is to enable a spatial confidence map for each agent, where each element reflects the perceptually critical level of a corresponding spatial area. Based on this map, agents decide which spatial area (where) to communicate about. That is, each agent offers spatially sparse, yet critical features to support other agents, and meanwhile requests complementary information from others through multi-round communication to perform efficient and mutually beneficial collaboration.

Following this strategy, we propose `Where2comm`, a novel communication-efficient multi-agent collaborative perception framework with the guidance of spatial confidence maps; see Fig. 2. `Where2comm` includes three key modules: i) a spatial confidence generator, which produces a spatial confidence map to indicate perceptually critical areas; ii) a spatial confidence-aware communication module, which leverages the spatial confidence map to decide *where* to communicate via novel message packing, and *who* to communicate via novel communication graph construction; and iii) a spatial confidence-aware message fusion module, which uses novel confidence-aware multi-head attention to fuse all messages received from other agents, upgrading the feature map for each agent.

`Where2comm` has two distinct advantages. First, it promotes pragmatic compression at the feature level and uses less communication to achieve higher perception performance by focusing on perceptually critical areas. Second, it adapts to various communication bandwidths and communication rounds, while previous models only handle one predefined communication bandwidth and a fixed number of communication rounds. To evaluate `Where2comm`, we consider the collaborative 3D object detection task on four datasets: DAIR-V2X [11], V2X-Sim [9], OPV2V [10] and our original dataset CoPerception-UAVs. Our experiments cover both real-world and simulation scenarios, two types of agents (cars and drones) and sensors (LiDAR and cameras). Results show that i) the proposed `Where2comm` consistently and significantly outperforms previous works in the performance-bandwidth trade-off across multiple datasets and modalities; and ii) `Where2comm` achieves better trade-off when the communication round increases.

2 Related Works

Multi-agent communication. The communication strategy in multi-agent systems has been widely studied [20]. Early works [21–23] often use predefined protocols or heuristics to decide how agents communicate with each other. However, it is difficult to generalize those methods to complex tasks. Recent works, thus, explore learning-based methods for complex scenarios. For example,

Table 1: Major components comparisons of collaborative perception systems.

Method	Venue	Message packing	Communication graph construction	Message fusion
When2com [12]	CVPR 2020	Full feature map	Handshake-based sparse graph	Attention per-agent
V2VNet [1]	ECCV 2020	Full feature map	Fully connected graph	Average per-agent
DiscoNet [2]	NeurIPS 2021	Full feature map	Fully connected graph	MLP-based attention per-location
V2X-ViT [26]	ECCV 2022	Full feature map	Fully connected graph	Self-attention per-location
Where2comm	NeurIPS 2022	*Confidence*-aware sparse feature map + request map	*Confidence*-aware sparse graph	*Confidence*-aware multi-head attention per-location

CommNet [24] learns continuous communication in the multi-agent system. Vain [25] adopts the attention mechanism to help agents selectively fuse the information from others. Most of these previous works consider decision-making tasks and adopt reinforcement learning due to the lack of explicit supervision. In this work, we focus on the perception task. Based on direct perception supervision, we apply supervised learning to optimize the communication strategy in both trade-off perception ability and communication cost.

Collaborative perception. As a recent application of multi-agent communication systems to perception tasks, collaborative perception is still immature. To support this area of research, there is a surge of high-quality datasets (e.g., V2X-Sim [9], OpenV2V [10], Comap[27] and DAIR-V2X[11]), as well as collaboration methods aimed for better performance-bandwidth trade-off (see comparisons in Table 1). When2com [12] proposes a handshake communication mechanism to decide *when* to communicate and create sparse communication graph. V2VNet [1] proposes multi-round message passing based on graph neural networks to achieve better perception and prediction performance. DiscoNet [2] adopts knowledge distillation to take the advantage of both early and intermediate collaboration. OPV2V [10] proposes a graph-based attentive intermediate fusion to improve perception performances. V2X-ViT [26] introduces a novel heterogeneous multi-agent attention module to fuse information across heterogeneous agents. In this work, we leverage the proposed spatial confidence map to promote more compact messages, more sparse communication graphs, and more comprehensive fusion, resulting in efficient and effective collaboration.

3 Problem Formulation

Consider N agents in the scene. Let $\mathcal{X}_i$ and $\mathcal{Y}_i$ be the observation and the perception supervision of the ith agent, respectively. The objective of collaborative perception is to achieve the maximized perception performance of all agents as a function of the total communication budge B and communication round K; that is,

$$\xi_\Phi(B,K) = \arg\max_{\theta,\mathcal{P}} \sum_{i=1}^{N} g\left(\Phi_\theta\left(\mathcal{X}_i, \{\mathcal{P}_{i\to j}^{(K)}\}_{j=1}^{N}\right), \mathcal{Y}_i\right), \ \text{ s.t. } \sum_{k=1}^{K}\sum_{i=1}^{N} |\mathcal{P}_{i\to j}^{(k)}| \le B,$$

where $g(\cdot,\cdot)$ is the perception evaluation metric, Φ is the perception network with trainable parameter θ, and $\mathcal{P}_{i\to j}^{(k)}$ is the message transmitted from the ith agent to the jth agent at the kth communication round. Note that i) when $B = K = 0$, there is no collaboration and $\xi_\Phi(0,0)$ reflects the single-agent perception performance; ii) through optimizing the communication strategy and the network parameter, collaborative perception should perform well consistently at any communication bandwidth or round; and iii) we consider multi-round communication, where each agent serves as both a supporter (offering message to help others) and a requester (requesting messages from others).

In this work, we consider the perception task of 3D object detection and present three contributions: i) we make communication more efficient by designing compact messages and sparse communication graphs; ii) we boost the perception performance by implementing more comprehensive message fusion; iii) we enable the overall system to adapt to varying communication conditions by dynamically adjusting where and who to communicate.

4 `Where2comm`: Spatial Confidence-Aware Collaborative Perception System

This section presents `Where2comm`, a multi-round, multi-modality, multi-agent collaborative perception framework based on a spatial-confidence-aware communication strategy; see the overview in Fig. 2. `Where2comm` includes an observation encoder, a spatial confidence generator, the spatial confidence-aware communication module, the spatial confidence-aware message fusion module and a detection decoder. Among five modules, the proposed spatial confidence generator generates the spatial confidence map. Based on this spatial confidence map, the proposed spatial confidence-aware communication generates compact messages and sparse communication graphs to save communication bandwidth; and the proposed spatial confidence-aware message fusion module leverages

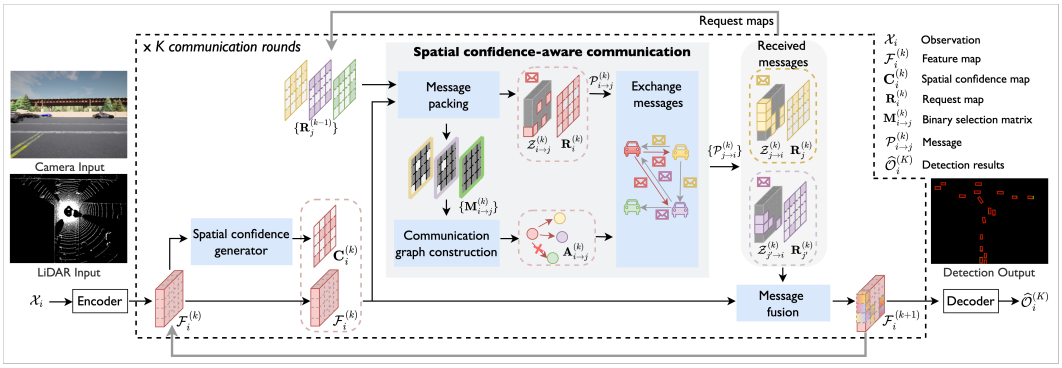


Figure 2: System overview. In `Where2comm`, spatial confidence generator enables the awareness of spatial heterogeneous of perceptual information, spatial confidence-aware communication enables efficient communication, and spatial confidence-aware message fusion boosts the performance.

informative spatial confidence priors to achieve better aggregation; also see an algorithmic summary in Algorithm 1 and the optimization-oriented design rationale in Section 7.3 in Appendix.

4.1 Observation encoder

The observation encoder extracts feature maps from the sensor data. `Where2comm` accepts single/multi-modality inputs, such as RGB images and 3D point clouds. This work adopts the feature representations in bird's eye view (BEV), where all agents project their individual perceptual information to the same global coordinate system, avoiding complex coordinate transformations and supporting better shared cross-agent collaboration. For the ith agent, given its input $\mathcal{X}_i$, the feature map is $\mathcal{F}_i^{(0)} = \Phi_{\text{enc}}(\mathcal{X}_i) \in \mathbb{R}^{H\times W\times D}$, where $\Phi_{\text{enc}}(\cdot)$ is the encoder, the superscript 0 reflects that the feature is obtained before communication and H, W, D are its height, weight and channel. All agents share the same BEV coordinate system. For the image input, $\Phi_{\text{enc}}(\cdot)$ is followed by a warping function that transforms the extracted feature from front-view to BEV. For 3D point cloud input, we discretize 3D points as a BEV map and $\Phi_{\text{enc}}(\cdot)$ extracts features in BEV. The extracted feature map is output to the spatial confidence generator and the message fusion module.

4.2 Spatial confidence generator

The spatial confidence generator generates a spatial confidence map from the feature map of each agent. The spatial confidence map reflects the perceptually critical level of various spatial areas. Intuitively, for object detection task, the areas that contain objects are more critical than background areas. During collaboration, areas with objects could help recover the miss-detected objects due to the limited view; and background areas could be omitted to save the precious bandwidth. So we represent the spatial confidence map with the detection confidence map, where the area with high perceptually critical level is the area that contains an object with a high confidence score.

To implement, we use a detection decoder structure to produce the detection confidence map. Given the feature map at the kth communication round, $\mathcal{F}_i^{(k)}$, the corresponding spatial confidence map is

$$\mathbf{C}_i^{(k)} = \Phi_{\text{generator}}(\mathcal{F}_i^{(k)}) \in [0,1]^{H\times W}, \tag{1}$$

where the generator $\Phi_{\text{generator}}(\cdot)$ follows a detection decoder. Since we consider multi-round collaboration, `Where2comm` iteratively updates the feature map by aggregating information from other agents. Once $\mathcal{F}_i^{(k)}$ is obtained, (1) is triggered to reflect the perceptually critical level at each spatial location. The proposed spatial confidence map answers a crucial question that was ignored by previous works: for each agent, information at which spatial area is worth sharing with others. By answering this, it provides a solid base for efficient communication and effective message fusion.

4.3 Spatial confidence-aware communication

With the guidance of spatial confidence maps, the proposed communication module packs compact messages with spatially sparse feature maps and transmits messages through a sparsely-connected communication graph. Most existing collaboration perception systems [1, 2, 26] considers full feature maps in the messages and fully-connected communication graphs. To reduce the communication bandwidth without affecting perception, we leverage the spatial confidence map to select the most

informative spatial areas in the feature map (where to communicate) and decide the most beneficial collaboration partners (who to communicate).

Message packing. Message packing determines what information should be included in the to-be-sent message. The proposed message includes: i) a request map that indicates at which spatial areas the agent needs to know more; and ii) a spatially sparse, yet perceptually critical feature map.

The request map of the ith agent is $\mathbf{R}_i^{(k)} = 1 - \mathbf{C}_i^{(k)} \in \mathbb{R}^{H \times W}$, negatively correlated with the spatial confidence map. The intuition is, for the locations with low confidence score, an agent is hard to tell if there is really no objects or it is just caused by the limited information (e.g. occlusion). Thus, the low confidence score indicates there could be missing information at that location. Requesting information at these locations from other agents could improve the current agent's detection accuracy.

The spatially sparse feature map are selected based on each agent's spatial confidence map and the received request maps from others. Specifically, a binary selection matrix is used to represent each location is selected or not, where 1 denotes selected, and 0 elsewhere. For the message sent from the ith agent to the jth agent at the kth communication round, the binary selection matrix is

$$\mathbf{M}_{i \to j}^{(k)} = \begin{cases} \Phi_{\text{select}}(\mathbf{C}_i^{(k)}) \in \{0,1\}^{H \times W}, & k = 0; \\ \Phi_{\text{select}}(\mathbf{C}_i^{(k)} \odot \mathbf{R}_j^{(k-1)}), \in \{0,1\}^{H \times W}, & k > 0; \end{cases} \tag{2}$$

where $\odot$ is the element-wise multiplication, $\mathbf{R}_j^{(k-1)}$ is the request map from the jth agent received at the previous round, $\Phi_{\text{select}}(\cdot)$ is the selection function which targets to select the most critical areas conditioned on the input matrix, which represents the critical level at the certain spatial location. We implement $\Phi_{\text{select}}(\cdot)$ by selecting the locations where the largest elements at in the given input matrix conditioned on the bandwidth limit; optionally, a Gaussian filter could be applied to filter out the outliers and introduce some context. In the initial communication round, each agent selects the most critical areas from its own perspective as the request maps from other agents are not available yet; in the subsequent rounds, each agent also takes the partner's request into account, enabling more targeted communication. Then, the selected feature map is obtained as $\mathcal{Z}_{i \to j}^{(k)} = \mathbf{M}_{i \to j}^{(k)} \odot \mathcal{F}_i^{(k)} \in \mathbb{R}^{H \times W \times D}$, which provides spatially sparse, yet perceptually critical information.

Overall, the message sent from the ith agent to the jth agent at the kth communication round is $\mathcal{P}_{i \to j}^{(k)} = (\mathbf{R}_i^{(k)}, \mathcal{Z}_{i \to j}^{(k)})$. Note that i) $\mathbf{R}_i^{(k)}$ provides spatial priors to request complementary information for the ith agent's need in the next round; the feature map $\mathcal{Z}_{i \to j}^{(k)}$ provides supportive information for the ith agent's need in the this round. They together enable mutually beneficial collaboration; ii) since $\mathcal{Z}_{i \to j}^{(k)}$ is sparse, we only transmit non-zero features and corresponding indices, leading to low communication cost; and iii) the sparsity of $\mathcal{Z}_{i \to j}^{(k)}$ is determined by the binary selection matrix, which dynamically allocates the communication budget at various spatial areas based on their perceptual critical level, adapting to various communication conditions.

Communication graph construction. Communication graph construction targets to identify when and who to communicate to avoid unnecessary communication that wastes the bandwidth. Most previous works [1, 2, 10] consider fully-connected communication graphs. When2com [12] proposes a handshake mechanism, which uses similar global features to match partners. This is hard to interpret because two agents, which have similar global features, do not necessarily need information from each other. Different from all previous works, we provide an explicit design rationale: the necessity of communication between the ith and the jth agents is simply measured by the overlap between the information that the ith agent has and the information that the jth agent needs. With the help of the spatial confidence map and the request map, we construct a more interpretable communication graph.

For the initial communication round, every agent in the system is not aware of other agents yet. To activate the collaboration, we construct a fully-connected communication graph. Every agent will broadcast its message to the rest of the system. For the subsequent communication rounds, we examine if the communication between agent i and agent j is necessary based on the maximum value of the binary selection matrix $\mathbf{M}_{i \to j}^{(k)}$, i.e. if there is at least one patch is activated, then we regard the connection is necessary. Formally, let $\mathbf{A}^{(k)}$ be the adjacency matrix of the communication graph at the kth communication round, whose (i, j)th element is

$$\mathbf{A}_{i,j}^{(k)} = \begin{cases} 1, & k = 0; \\ \max_{h \in \{0,1,..,H-1\}, w \in \{0,1,...,W-1\}} \left(\mathbf{M}_{i \to j}^{(k)}\right)_{h,w} \in \{0,1\}, & k > 0; \end{cases}$$

where h, w index the spatial area, reflecting message passing from the ith agent to the jth agent. Given this sparse communication graph, agents can exchange messages with selected partners.

4.4 Spatial confidence-aware message fusion

Spatial confidence-aware message fusion targets to augment the feature of each agent by aggregating the received messages from the other agents. To achieve this, we adopt a transformer architecture, which leverages multi-head attention to fuse the corresponding features from multiple agents at each individual spatial location. The key technical design is to include the spatial confidence maps of all the agents to promote cross-agent attention learning. The intuition is that, the spatial confidence map could explicitly reflect the perceptually critical level, providing a useful prior for attention learning.

Specifically, for the ith agent, after receiving the jth agent's message $\mathcal{P}_{j\to i}^{(k)}$, it could unpack to retrieve the feature map $\mathcal{Z}_{j\to i}^{(k)}$ and the spatial confidence map $\mathbf{C}_j^{(k)} = 1 - \mathbf{R}_j^{(k)}$. We also include the ego feature map in fusion and denote $\mathcal{Z}_{i\to i}^{(k)} = \mathcal{F}_i^{(k)}$ to make the formulation simple and consistent, where $\mathcal{Z}_{i\to i}^{(k)}$ might not be sparse. To fuse the features from the jth agent at the kth communication round, the cross-agent/ego attention weight for the ith agent is

$$\mathbf{W}_{j\to i}^{(k)} = \mathrm{MHA_W}\left(\mathcal{F}_i^{(k)}, \mathcal{Z}_{j\to i}^{(k)}, \mathcal{Z}_{j\to i}^{(k)}\right) \odot \mathbf{C}_j^{(k)} \in \mathbb{R}^{H\times W}, \tag{3}$$

where $\mathrm{MHA_W}(\cdot)$ is a multi-head attention applied at each individual spatial location, which outputs the scaled dot-product attention weight. Note that i) the proposed spatial confidence maps contributes to the attention weight, as the features with higher perceptually critical level are more preferred in the feature aggregation; ii) the cross-agent attention weight models the collaboration strength with a $H \times W$ spatial resolution, leading to more flexible information fusion at various spatial regions. Then, the feature map of the ith agent after fusing the messages in the kth communication round is

$$\mathcal{F}_i^{(k+1)} = \mathrm{FFN}\left(\sum_{j\in\mathcal{N}_i \bigcup \{i\}} \mathbf{W}_{j\to i}^{(k)} \odot \mathcal{Z}_{j\to i}^{(k)}\right) \in \mathbb{R}^{H\times W\times D},$$

where $\mathrm{FFN}(\cdot)$ is the feed-forward network and $\mathcal{N}_i$ is the neighbors of the ith agent defined in the communication graph $\mathbf{A}^{(k)}$. The fused feature $\mathcal{F}_i^{(k+1)}$ would serve as the ith agent's feature in the $(k+1)$th round. In the final round, we output $\mathcal{F}_i^{(k+1)}$ to the detection decoder to generate detections.

Sensor positional encoding. Sensor positional encoding represents the physical distance between each agent's sensor and its observation. It adopts a standard positional encoding function conditioned on the sensing distance and feature dimension. The features are summed up with the positional encoding of each location before inputting to the transformer.

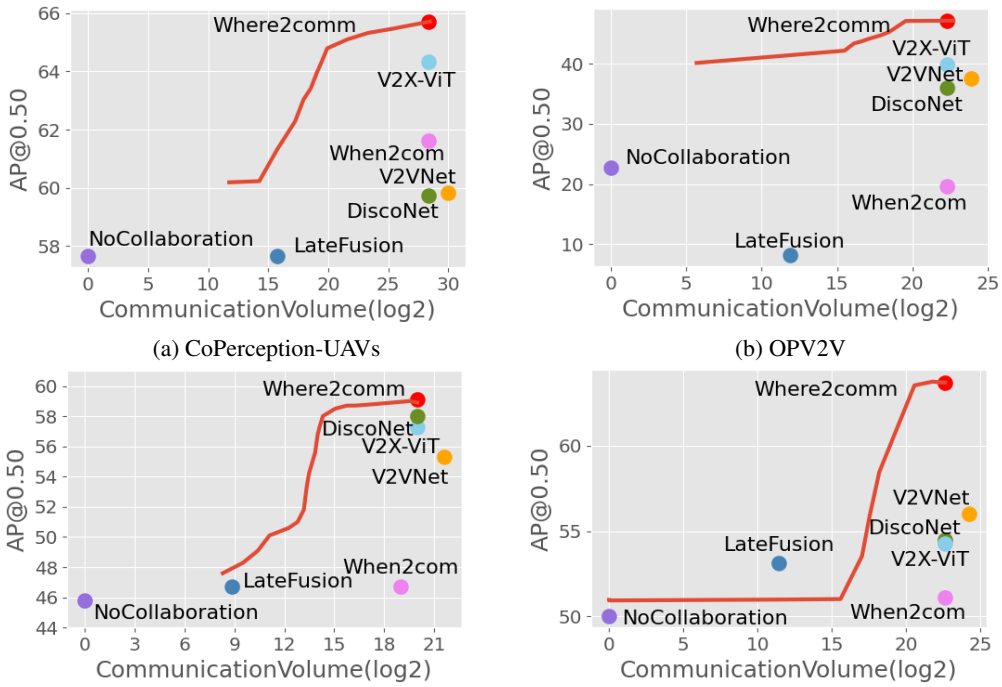
Compared to existing fusion modules that do not use attention mechanism [1] or only use agent-level attentions [12], the per-location attention mechanism adopted by the proposed fusion emphasizes the location-specific feature interactions. It makes the feature fusion more targeted. Compared to the methods that also use the per-location attention-based fusion module[2, 10, 26], the proposed fusion module leverages multi-head attention with two extra priors, including spatial confidence map and sensing distances. Both assist attention learning to prefer high quality and critical features.

4.5 Detection decoder

The detection decoder decodes features into objects, including class and regression output. Given the feature map at the kth communication round $\mathcal{F}_i^{(k)}$, the detection decoder $\Phi_{\mathrm{dec}}(\cdot)$ generate the detections of ith agent by $\widehat{\mathcal{O}}_i^{(k)} = \Phi_{\mathrm{dec}}(\mathcal{F}_i^{(k)}) \in \mathbb{R}^{H\times W\times 7}$, where each location of $\widehat{\mathcal{O}}_i^{(k)}$ represents a rotated box with class $(c, x, y, h, w, \cos\alpha, \sin\alpha)$, denoting class confidence, position, size and angle. The objects are the final output of the proposed collaborative perception system. Note that $\widehat{\mathcal{O}}_i^{(0)}$ denotes the detections without collaboration.

4.6 Training details and loss functions

To train the overall system, we supervise two tasks: spatial confidence generation and object detection at each round. As mentioned before, the functionality of the spatial confidence generator is the same as the classification in the detection decoder. To promote parameter efficiency, our spatial confidence generator reuses the parameters of the detection decoder. For the multi-round settings, each round is



(a) CoPerception-UAVs (b) OPV2V

(c) V2X-Sim (d) DAIR-V2X

Figure 3: `Where2comm` achieves consistently superior performance-bandwidth trade-off on all the three collaborative perception datasets, e.g, `Where2comm` achieves *5,000* times less communication volume and still outperforms When2com on CoPerception-UAVs dataset. The entire red curve comes from a single `Where2comm` model evaluated at varying bandwidths.

supervised with one detection loss, the overall loss is $L = \sum_{k=0}^{K} \sum_{i}^{N} L_{\text{det}} \left(\widehat{\mathcal{O}}_i^{(k)}, \mathcal{O}_i \right)$, where $\mathcal{O}_i$ is the ith agent's ground-truth objects, L_{det} is the detection loss [28].

Training strategy for multi-round setting. To adapt to multi-round communication and dynamic bandwidth, we train the model under various communication settings with curriculum learning strategy [29]. We first gradually increase the communication bandwidth and round; and then, randomly sample bandwidth and round to promote robustness. Through this training strategy, a single model can perform well at various communication conditions.

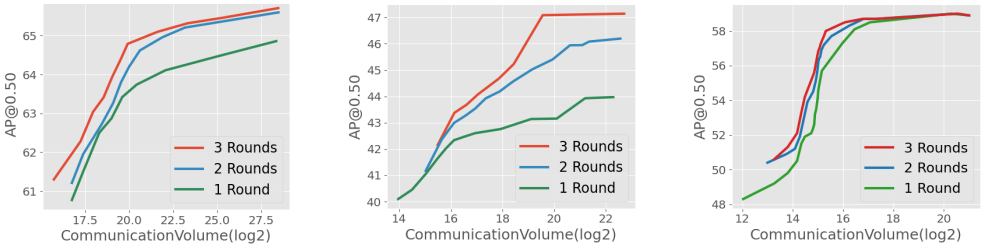
5 Experimental Results

Our experiments covers four datasets, both real-world and simulation scenarios, two types of agents (cars and drones) and two types of sensors (LiDAR and cameras). Specifically, we conduct camera-only 3D object detection in the setting of V2X-communication aided autonomous driving on OPV2V dataset [10], camera-only 3D object detection in the setting of drone swarm on the proposed CoPerception-UAVs dataset, and LiDAR-based 3D object detection on DAIR-V2X dataset [11] and V2X-Sim dataset [9]. The detection results are evaluated by Average Precision (AP) at Intersection-over-Union (IoU) threshold of 0.50 and 0.70. The communication results count the message size by byte in log scale with base 2. To compare communication results straightforward and fair, we do not consider any extra data/feature/model compression.

5.1 Datasets and experimental settings

OPV2V. OPV2V [10] is a vehicle-to-vehicle collaborative perception dataset, co-simulated by OpenCDA [10] and Carla [30]. It includes 12K frames of 3D point clouds and RGB images with 230K annotated 3D boxes. The perception range is 40m$\times$40m. For camera-only 3D object detection task on OPV2V, we implement the detector following CADDN [31]. The input front-view image size is $(416, 160)$. The front-view input feature map is transformed to BEV with resolution 0.5m/pixel.

V2X-Sim. V2X-Sim [9] is a vehicle-to-everything collaborative perception dataset, co-simulated by SUMO [32] and Carla, including 10K frames of 3D LiDAR point clouds and 501K 3D boxes.



(a) CoPerception-UAVs (b) OPV2V (c) V2X-Sim

Figure 4: More communication rounds continuously improve performance-bandwidth trade-off.

The perception range is 64m×64m. For LiDAR-based 3D object detection task, our detector follows MotionNet [33]. We discretize 3D points into a BEV map with size $(256, 256, 13)$ and the resolution is 0.4m/pixel in length and width, 0.25m in height.

CoPerception-UAVs. To enrich the collaborative perception datasets, we consider the swarm of unmanned aerial vehicles (UAV) and propose a UAV-swarm-based collaborative perception dataset: CoPerception-UAVs, co-simulated by AirSim [34] and Carla [30], including 131.9K aerial images and 1.94M 3D boxes. The perception range is 200m×350m. For the camera-only 3D object detection task on CoPerception-UAVs, our detector follows DVDET [8]. The input aerial image size is $(800, 450)$. The aerial-view input feature map is transformed to BEV with the resolution of 0.25m/pixel, and the size is $(192, 352)$; see more details in Appendix.

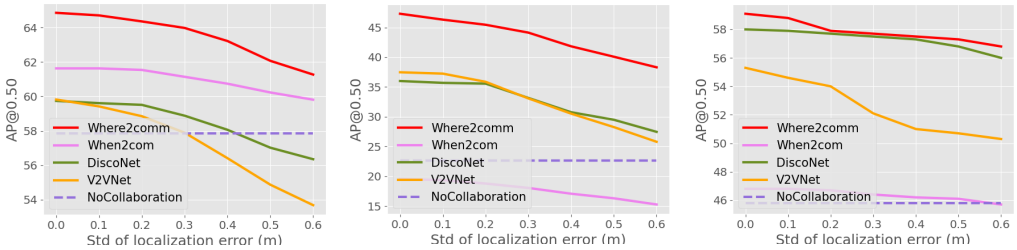
DAIR-V2X. DAIR-V2X [11] is the only public **real-world** collaborative perception dataset. Each sample contains two agents: a vehicle and an infrastructure, with 3D annotations. The perception range is 201.6m×80m. Originally DAIR-V2X does not label objects outside the camera's view, we relabel all objects to cover 360-degree detection range. We complement several intermediate fusion-based baselines on DAIR-V2X to comprehensively validate our method on real data. For LiDAR-based 3D object detection task, our detector follows PointPillar [35]. We represent the field of view into a BEV map with size $(200, 504, 64)$ and the resolution is 0.4m/pixel in length and width.

5.2 Quantitative evaluation

Benchmark comparison. Fig. 3 compares the proposed `Where2comm` with the previous methods in terms of the trade-off between detection performance (AP@IoU=0.50) and communication bandwidth; also see exact values in Table 3 of Appendix. We consider single-agent detection without collaboration ($\widehat{\mathcal{O}}_i^{(0)}$), When2com [12], V2VNet [1], DiscoNet [2], V2X-ViT [26] and late fusion, where agents directly exchange the detected 3D boxes. The red curve comes from a single `Where2comm` model evaluated at varying bandwidths. We see that the proposed `Where2comm`: i) achieves a far-more superior perception-communication trade-off across all the communication bandwidth choices and various collaborative perception tasks, including camera-only 3D object detection from aerial view and car front view, and LiDAR-based 3D object detection; ii) achieves significant improvements over previous state-of-the-arts on both real-world (DAIR-V2X) and simulation scenarios, improves the SOTA performance by 7.7% on DAIR-V2X, 6.62% on CoPerception-UAVs, 25.81% on OPV2V, 1.9% on V2X-Sim; iii) achieves the same detection performance of previous state-of-the-arts with extremely less communication volume: 5128 times less on CoPerception-UAVs, more than 100K times less on OPV2V, 55 times less on V2X-Sim, 105 times less on DAIR-V2X.

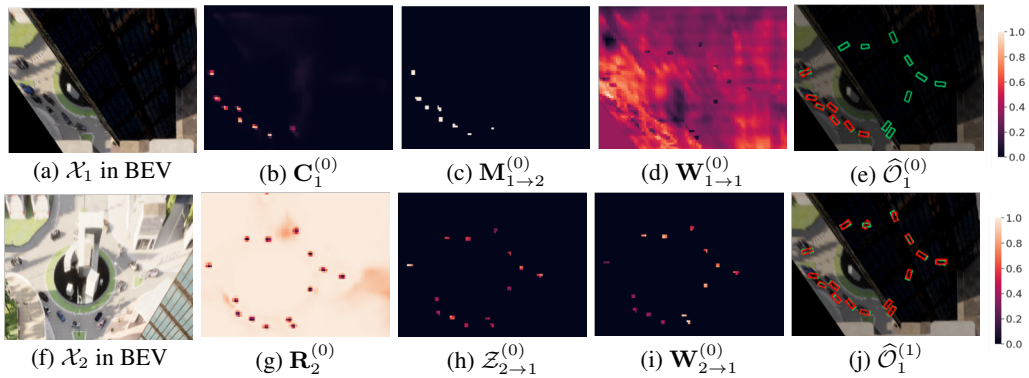
Multi-round evaluation. Fig. 4 presents the performances of `Where2comm` at communication rounds ranging from 1 to 3. Each curve comes from a single `Where2comm` model with a certain communication round evaluated at varying bandwidths. Results show that 1 communication round is good, more rounds are even better. Multi-round communication steadily improves the performance-bandwidth trade-off across all three datasets, reflecting its effectiveness and robustness. This encourages the agents to actively collaborate without worrying the performance degradation. This also validates that `Where2comm` can well work at various communication bandwidths and rounds.

Robustness to localization noise. We follow the localization noise setting in V2VNet and V2X-ViT (Gaussian noise with a mean of 0m and a standard deviation of 0m-0.6m) and conduct experiments on all the three datasets to validate the robustness against realistic localization noise. *Where2comm* is more robust to the localization noise than previous SOTAs. Fig. 5 shows the detection performances as a function of localization noise level in CoPerception-UAVs, OPV2V and V2X-Sim datasets, respectively We see: i) overall the collaborative perception performance degrades with the increasing



(a) CoPerception-UAVs (b) OPV2V (c) V2X-Sim

Figure 5: Robustness to localization error. Gaussian noise with zero mean and varying std is introduced. *Where2comm* consistently outperforms previous SOTAs and No Collaboration.



(a) $\mathcal{X}_1$ in BEV (b) $\mathbf{C}_1^{(0)}$ (c) $\mathbf{M}_{1\to2}^{(0)}$ (d) $\mathbf{W}_{1\to1}^{(0)}$ (e) $\widehat{\mathcal{O}}_1^{(0)}$

(f) $\mathcal{X}_2$ in BEV (g) $\mathbf{R}_2^{(0)}$ (h) $\mathcal{Z}_{2\to1}^{(0)}$ (i) $\mathbf{W}_{2\to1}^{(0)}$ (j) $\widehat{\mathcal{O}}_1^{(1)}$

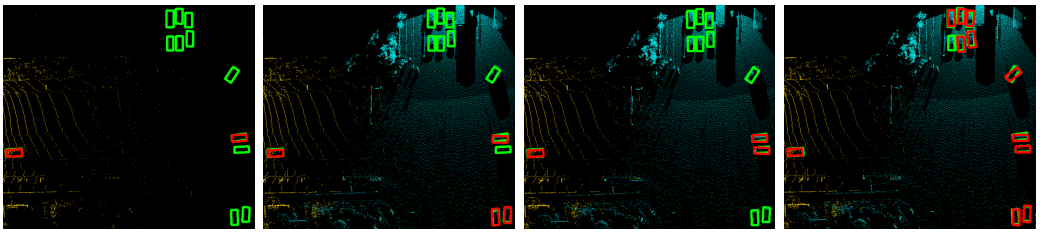
Figure 6: Visualization of collaboration between Drone 1 and Drone 2 on CoPerception-UAVs dataset, including spatial confidence map ($\mathbf{C}_1^{(0)}$), selection matrix ($\mathbf{M}_{1\to2}^{(0)}$), message ($\{\mathbf{R}_2^{(0)}, \mathcal{Z}_{2\to1}^{(0)}\}$) in the communication module, attention weight in the fusion module ($\mathbf{W}_{1\to1}^{(0)}$,$\mathbf{W}_{2\to1}^{(0)}$), and Drone 1's detection results before ($\widehat{\mathcal{O}}_1^{(0)}$) and after ($\widehat{\mathcal{O}}_1^{(1)}$) collaboration. Green and red boxes denote ground-truth and detection, respectively. The objects occluded by a tall building can be detected through transmitting spatially sparse, yet perceptually critical message.

localization noise, while *where2comm* outperforms previous SOTAs (When2com, V2VNet,DiscoNet) under all the localization noise. ii) *where2comm* keeps being superior to *No Collaboration* while V2VNet fails when noise is over 0.4m and DiscoNet fails when noise is over 0.5m on CoPerception-UAVs. The reasons are: i) the powerful transformer architecture in fusion module attentively select the most suitable collaborative feature; ii) the spatial confidence map helps filter out noisy features, these two designs work together to mitigate noise localization distortion effects.

5.3 Qualitative evaluation

Visualization of spatial confidence map. Fig. 6 illustrates how `Where2comm` is empowered by the proposed spatial confidence map. In the scene, Drone 1's view is occluded by a tall building. With Drone 2's help, Drone 1 is able to detect through occlusion. Fig. 6 (a-d) shows Drone 1's observation, spatial confidence map (1), binary selection matrix (2), and ego attention weight (3). Fig. 6 (f-h) shows Drone 2's observation and message sent to Drone 1, including the request map (opposite of confidence map) and the sparse feature map, achieving efficient communication. Fig. 6 (i) shows the attention weight for Drone 1 to fuse Drone 2's messages, which is sparse, yet highlights the objects' positions. Fig. 6 (e) and (j) compares the detection results before and after the collaboration with Drone 2. We see that the proposed spatial confidence map contributes to spatially sparse, yet perceptually critical message, which effectively helps Drone 1 detect occluded objects.

Visualization of detection results. Fig. 7 shows that compared to *No Collaboration*, *When2com* and *DiscoNet*, `Where2comm` is able to achieves more complete and accurate detection results. The reason is that *When2com* employs a scalar to denote the agent-to-agent attention, which cannot distinguish which spatial area is more informative; *DiscoNet* employs a MLP-based fusion weight learning, which cannot well capture the complex collaboration attention; while `Where2comm` can zoom in to critical spatial areas in a cell-level resolution and leverage the spatial confidence map and sensing distances as priors to achieve more comprehensive fusion.



(a) No Collaboration (b) When2com (c) DiscoNet (d) Where2comm

Figure 7: `Where2comm` qualitatively outperforms When2com and DiscoNet in DAIR-V2X dataset. Green and red boxes denote ground-truth and detection, respectively. Yellow and blue denote the point clouds collected from vehicle and infrastructure, respectively.

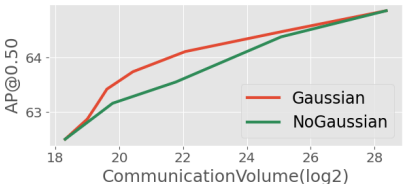


Figure 8: Selection matrix ablation study. Applying Gaussian filter improves performance.

Table 2: Fusion component ablation study. Multi-head attention (MHA), sensor positional encoding (SPE) and spatial confidence map (SCM) all improves the performances. Results are reported in AP@0.50/AP@0.70.

MHA	SPE	SCM	OPV2V	CoPerception-UAVs	V2X-Sim
			34.96/13.92	63.48/44.23	51.2/45.7
✓			38.75/13.28	63.99/44.46	57.3/50.8
✓	✓		39.82/16.43	64.34/46.86	59.1/52.0
✓	✓	✓	**47.30/19.30**	**64.83/47.62**	**59.1/52.2**

5.4 Ablation studies

Effect of Gaussian filter in perceptually critical area selection. Fig. 8 compares two versions of the selection matrix (2) with and without Gaussian filter. We see that applying Gaussian filter improves the overall performance. The reason is that: i) Gaussian filter could help filter out the outliers in the input map, selecting more robust critical regions; ii) it considers the context, benefiting the independent feature selection at each certain location by providing more information.

Effect of components in spatial confidence-aware message fusion. Tab. 2 assesses the effectiveness of the proposed fusion with two priors. We see that: i) per-location multi-head attention (MHA) outperforms the vanilla attention by 10.84% on OPV2V on AP@0.50, because MHA leverages information from multiple heads, better capturing cross-agent attention; and ii) As two informative priors, both sensing position encoding (SPE) and spatial confidence map (SCM) can consistently improve the performance. Especially, the version with all three designs improves the detection performance by 22.06% on OPV2V on AP@0.50.

6 Conclusion and limitation

We propose `Where2comm`, a novel communication-efficient collaborative perception framework. The core idea is to exploit a spatial confidence map at each agent to promote pragmatic compression, assisting agents to decide what to communicate with whom, and whose information to aggregate. Each agent offers spatially sparse, yet perceptually critical features to support other agents; meanwhile, requests complementary information from others in multi-round communication. Comprehensive experiments covering multi-type agents and multi-modality inputs show that `Where2comm` achieves far superior trade-off between perception performance and communication bandwidth.

Limitation and future work. The current work focuses on perceptually critical spatial areas. In future, we plan to expand a similar idea to the temporal dimension and determine critical time stamps. More cost will be reduced by exploring when to communicate. We also expect that more methods on pragmatic compression and emergent communication could be applied to collaborative perception.

Acknowledgment. This research is partially supported by the National Key R&D Program of China under Grant 2021ZD0112801, National Natural Science Foundation of China under Grant 62171276, the Science and Technology Commission of Shanghai Municipal under Grant 21511100900, CCF-DiDi GAIA Research Collaboration Plan 202112 and CALT Grant 2021-01.

References

[1] Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *European Conference on Computer Vision*, pages 605–621. Springer, 2020.

[2] Yiming Li, Shunli Ren, Pengxiang Wu, Siheng Chen, Chen Feng, and Wenjun Zhang. Learning distilled collaboration graph for multi-agent perception. *Advances in Neural Information Processing Systems*, 34, 2021.

[3] Siheng Chen, Baoan Liu, Chen Feng, Carlos Vallespi-Gonzalez, and Carl K. Wellington. 3d point cloud processing and learning for autonomous driving: Impacting map creation, localization, and perception. *IEEE Signal Processing Magazine*, 38:68–86, 2021.

[4] Zhi Li, Ali Vatankhah Barenji, Jiazhi Jiang, Ray Y Zhong, and Gangyan Xu. A mechanism for scheduling multi robot intelligent warehouse system face with dynamic demand. *Journal of Intelligent Manufacturing*, 31(2):469–480, 2020.

[5] Michela Zaccaria, Mikhail Giorgini, Riccardo Monica, and Jacopo Aleotti. Multi-robot multiple camera people detection and tracking in automated warehouses. In *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, pages 1–6. IEEE, 2021.

[6] Jürgen Scherer, Saeed Yahyanejad, Samira Hayat, Evsen Yanmaz, Torsten Andre, Asif Khan, Vladimir Vukadinovic, Christian Bettstetter, Hermann Hellwagner, and Bernhard Rinner. An autonomous multi-uav system for search and rescue. In *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pages 33–38, 2015.

[7] Ebtehal Turki Alotaibi, Shahad Saleh Alqefari, and Anis Koubaa. Lsar: Multi-uav collaboration for search and rescue missions. *IEEE Access*, 7:55817–55832, 2019.

[8] Yue Hu, Shaoheng Fang, Weidi Xie, and Siheng Chen. Aerial monocular 3d object detection. *arXiv preprint arXiv:2208.03974*, 2022.

[9] Yiming Li, Ziyan An, Zixun Wang, Yiqi Zhong, Siheng Chen, and Chen Feng. V2X-Sim: A virtual collaborative perception dataset for autonomous driving. *IEEE Robotics and Automation Letters*, 7, 2022.

[10] 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. *ICRA*, 2022.

[11] 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.

[12] 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.

[13] 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.

[14] Yang Zhou, Jiuhong Xiao, Yue Zhou, and Giuseppe Loianno. Multi-robot collaborative perception with graph neural networks. *IEEE Robotics and Automation Letters*, 2022.

[15] Eduardo Arnold, Mehrdad Dianati, and Robert de Temple. Cooperative perception for 3d object detection in driving scenarios using infrastructure sensors. *IEEE Transactions on Intelligent Transportation Systems*, 23:1852–1864, 2022.

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

[17] Yiming Li, Juexiao Zhang, Dekun Ma, Yue Wang, and Chen Feng. Multi-robot scene completion: Towards task-agnostic collaborative perception. In *Conference on Robot Learning (CoRL)*. PMLR, 2022.

[18] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. CoBEVT: Cooperative bird's eye view semantic segmentation with sparse transformers. *CoRL*, 2022.

[19] Sanbao Su, Yiming Li, Sihong He, Songyang Han, Chen Feng, Caiwen Ding, and Fei Miao. Uncertainty quantification of collaborative detection for self-driving, 2022.

[20] Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. *ICLR*, 2019.

[21] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pages 330–337, 1993.

[22] Faisal Qureshi and Demetri Terzopoulos. Smart camera networks in virtual reality. *Proceedings of the IEEE*, 96(10):1640–1656, 2008.

[23] Yiming Li, Bir Bhanu, and Wei Lin. Auction protocol for camera active control. In *2010 IEEE International Conference on Image Processing*, pages 4325–4328. IEEE, 2010.

[24] Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29, 2016.

[25] Yedid Hoshen. Vain: Attentional multi-agent predictive modeling. *Advances in Neural Information Processing Systems*, 30, 2017.

[26] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2X-ViT: Vehicle-to-everything cooperative perception with vision transformer. *ECCV*, 2022.

[27] Y Yuan and M Sester. Comap: A synthetic dataset for collective multi-agent perception of autonomous driving. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:255–263, 2021.

[28] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019.

[29] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[30] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[31] Cody Reading, Ali Harakeh, Julia Chae, and Steven L. Waslander. Categorical depth distribution network for monocular 3d object detection. *CVPR*, 2021.

[32] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of sumo-simulation of urban mobility. *International journal on advances in systems and measurements*, 5(3&4), 2012.

[33] Pengxiang Wu, Siheng Chen, and Dimitris N. Metaxas. Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11382–11392, 2020.

[34] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, pages 621–635. Springer, 2018.

[35] 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, 2019.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[37] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2018.

[38] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. *ICLR*, 2021.

[39] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 5.1 and the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5.1 and the supplemental material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We have not repeated experiments many times to get error bars since experiments of 3d object detection on large scale datasets is time-consuming.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 5.1 and the supplemental material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 5.1.
   (b) Did you mention the license of the assets? [Yes] See Section 5.1.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See Section 5.1 and the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We generate the data using the open-sourced tool and the owners consent to all the public use for research purposes.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] Our data is synthesized using the open-sourced tool, so there is no real-world personally identifiable information, nor offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Algorithm 1 Multi-round spatial confidence-aware collaborative perception system

Define N as the number of agents , K as communication round
# Initialization
for $i = 1, 2, \ldots, N$, **do**
  $\mathcal{F}_i^{(0)} = \Phi_{\text{enc}}(\mathcal{X}_i) \in \mathbb{R}^{H \times W \times D}$ ▷ Extract intermediate feature
end for
for $k = 0, 1, \ldots, K-1$, **do**
  for $i = 1, 2, \ldots, N$, **do** # Each agent is computing individually
    $\mathbf{C}_i^{(k)} = \Phi_{\text{generator}}(\mathcal{F}_i^{(k)}) \in \mathbb{R}^{H \times W}$ ▷ Generate spatial confidence map
    for $j = 1, 2, \ldots, N$, **do**
      # Message packing
      $\mathbf{R}_i^{(k)} = 1 - \mathbf{C}_i^{(k)} \in \mathbb{R}^{H \times W}$ ▷ Pack request map
      if $k = 0$ **then**
        $\mathbf{M}_{i \to j}^{(k)} = \Phi_{\text{select}}(\mathbf{C}_i^{(k)}) \in \{0, 1\}^{H \times W}$ ▷ Select critical areas
      else
        $\mathbf{M}_{i \to j}^{(k)} = \Phi_{\text{select}}(\mathbf{C}_i^{(k)} \odot \mathbf{R}_j^{(k-1)}) \in \{0, 1\}^{H \times W}$ ▷ Select requested areas
      end if
      $\mathcal{Z}_{i \to j}^{(k)} = \mathbf{M}_{i \to j}^{(k)} \odot \mathcal{F}_i^{(k)} \in \mathbb{R}^{H \times W \times D}$ ▷ Pack spatially sparse features
      # Communication graph learning
      if $k = 0$ **then**
        $\mathbf{A}_{i \to j}^{(k)} = 1$ ▷ Broadcast critical features and request
      else
        $\mathbf{A}_{i \to j}^{(k)} = \max_{h,w} \left(\mathbf{M}_{i \to j}^{(k)}\right)_{h,w} \in \{0, 1\}$ ▷ Communicate only when necessary
      end if
    end for
    # Communication
    Send $\mathcal{P}_{i \to j} = \left(\mathcal{Z}_{i \to j}^{(k)}, \mathbf{R}_i^{(k)}\right)$ to other agents
    Receive $\{\mathcal{P}_{j \to i} = \left(\mathcal{Z}_{j \to i}^{(k)}, \mathbf{R}_j^{(k)}\right), j \neq i\}$ from other agents
    # Message fusion
    $\mathcal{F}_i^{(k+1)} = f_{\text{fuse}}\left(\mathcal{F}_i^{(k)}, \{(\mathcal{Z}_{j \to i}^{(k)}, \mathbf{R}_j^{(k)}), j = 1, 2, ..., N\}\right) \in \mathbb{R}^{H \times W \times D}$
  end for
  Store $\mathcal{F}_i^{(k+1)}$ and $\{\mathbf{R}_j^{(k)}, j \neq i\}$ for the next round
end for
$\mathcal{O}_i^{(K)} = \Phi_{\text{dec}}(\mathcal{F}_i^{(K)})$ ▷ Output the final detections

7 Appendix

7.1 Highlights of our contribution

To sum up, our contributions are:

• We propose a novel fine-grained spatial-aware communication strategy, where each agent can decide where to communicate and pack messages only related to the most perceptually critical spatial areas. This strategy not only enables more precise support for other agents, but also more targeted request from other agents in multi-round communication.

• We propose `Where2comm`, a novel collaborative perception framework based on the spatial-aware communication strategy. With the guidance of the proposed spatial confidence map, `Where2comm` leverages novel message packing and communication graph learning to achieve lower communication bandwidth, and adopts confidence-aware multi-head attention to reach better perception performance.

• We conduct extensive experiments to validate `Where2comm` achieves state-of-the-art performance-bandwidth trade-off on multiple challenging real/simulated datasets across views and modalities.

7.2 Detailed information about the system pipeline

Alg. 1 presents the pipeline of our multi-round spatial confidence-aware collaborative perception system.

7.3 Detailed information about the optimization problem of collaborative perception

The constrained optimization in Sec.3 is the mathematical formation of collaborative perception. It is hard to obtain the global optimum due to hard constrains and non-differentialability of binary variables. Therefore, the proposed Where2comm essentially introduces an auxiliary variable and decomposes the original problem into two sub-optimization problems, each one of which is easy to solve.

To understand the details, let us consider a setting of fixed communication bandwidth and communication round, $K = 1, B = [B_1]$. Then, the optimization is

$$\max_{\theta,\mathcal{P}} \sum_{i=1}^{N} g\left(\Phi_\theta\left(\mathcal{X}_i, \{\mathcal{P}_{i\to j}\}_{j=1}^{N}\right), \mathcal{Y}_i\right), \ \text{ s.t. } \sum_{i,j=1}^{N} |\mathcal{P}_{i\to j}| \leq B_1.$$

Since there is only one round, we do not consider the request map, then, the message sent from the ith agent to the jth agent is $\mathcal{P}_{i\to j} = \mathcal{Z}_{i\to j} = \mathbf{M}_{i\to j} \odot \mathcal{F}_i$, whose spatial sparsity is determined by the binary selection mask $\mathbf{M}_{i\to j}$. Note that $\mathbf{M}_{i\to j}$ determines where to communicate, and is the key of the proposed Where2comm. Then, the original optimization is equivalent to

$$\max_{\theta,\mathbf{M}} \sum_{i=1}^{N} g\left(\Phi_\theta\left(\mathcal{X}_i, \{\mathbf{M}_{i\to j}\}_{j=1}^{N}\right), \mathcal{Y}_i\right), \ \text{ s.t. } \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} |\mathbf{M}_{i\to j}| \leq b_1, \mathbf{M}_{i\to j} \in \{0,1\}^{H\times W}, \tag{4}$$

where $\mathcal{F}_i$ can attribute to the network $\Phi_\theta(\cdot)$ and input data $\mathcal{X}_i$ and $b_1 = B_1/D$ with D the channel number of $\mathcal{F}_i$. Due to the binary constrains, it is hard to optimize (4) directly. Instead, we decompose (4) into two sub-optimization problems and optimize the binary selection matrix $\mathbf{M}_{i\to j}$ and the network parameters θ once at a time: i) obtain a feasible binary selection matrix $\mathbf{M}_{i\to j}$ by optimizing a proxy constrained problem; ii) given the feasible binary selection matrix $\mathbf{M}_{i\to j}$, optimize the perception network parameter θ. The constraint is satisfied in i) and the perception goal is achieved in ii). Specifically, two sub-optimization problems are

• **Obtain a feasible binary selection matrix $\mathbf{M}_{i\to j}$.** This essentially optimizes where to allocate the communication bandwidth. Intuitively, the spatial confidence reflects the perceptually critical level, so that those spatial regions with higher spatial confidence will provide more critical information to help the partners and should have a higher priority be selected.

Following this spirit, we consider a proxy constrained problem as follows,

$$\max_{\mathbf{M}} \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} \mathbf{M}_{i\to j} \odot \mathbf{C}_i, \ \text{ s.t. } \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} |\mathbf{M}_{i\to j}| \leq b_1, \mathbf{M}_{i\to j} \in \{0,1\}^{H\times W},$$

where $\mathbf{C}_i$ is the spatial confidence map. Note that i) even this optimization problem has hard constraints and non-differentialability of binary variables, it has an analytical solution that naturally satisfies all the constraints in (4); and ii) even we cannot solve the original objective, this proxy objective still carries the similar idea to promote better, yet more compact perception. This solution is obtained by selecting those spatial regions whose corresponding elements in $\mathbf{M}$ rank top-b_1. The detailed steps of **selection function** are: i) arrange the elements in the input matrix in descending order; ii) given the communication budget constrain, decide the total number (b_1) of communication regions; iii) set the spatial regions of $\mathbf{M}$, where elements rank in top-b_1 as the 1 and 0 verses.

• **Given the feasible binary selection matrix, optimize the network parameter θ.** This essentially optimizes the perception performance. The sub-problem is

$$\max_{\theta} \sum_{i=1}^{N} g\left(\Phi_\theta\left(\mathcal{X}_i, \{\mathbf{M}_{i\to j}\}_{j=1}^{N}\right), \mathcal{Y}_i\right).$$

This can be solved by standard **supervised learning**. For example, the perception evaluation metric $g(\cdot)$ can be evaluated by the detection loss calculated between detections and the ground-truth and the

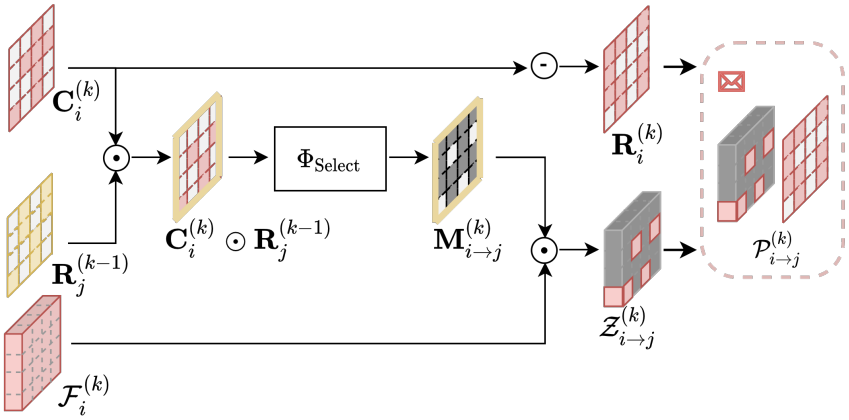


Figure 9: Spatial confidence-aware message packing module. $\odot$ denotes point-wise multiplication, $\ominus$ denotes point-wise minus by a matrix with the same shape as the input and filled with 1. Best viewed in color. Grey denotes the location being filled with zeros for the binary selection matrix $\mathbf{M}_{i\to j}^{(k)}$ and the feature map $\mathcal{Z}_{i\to j}^{(k)}$.

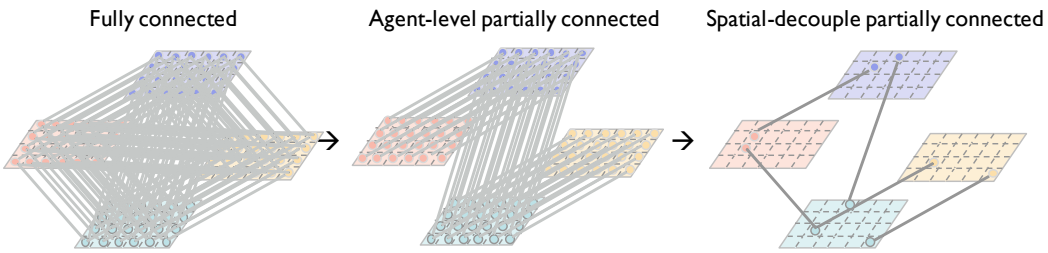


Figure 10: Spatial confidence-aware communication graph construction module. We spatially decouple the full feature map, and could flexibly involve the informative spatial areas in the communication. This *Spatial-decouple partially connected* communication could further flexibly prune irrelevant connections per-location and is more bandwidth-efficient.

detection loss is optimized with an Adam optimizer. We thus get the optimized perception network parameter θ. Note that this sub-problem does not involve any constraints and is thus easy to optimize.

7.4 Detailed information about the module design

Observation encoder. Here we elaborate on the warping functions for the monocular camera, where the depth is unknown and estimated. Instead of directly projecting 2D features to flat ground space, we first lift them to 3D voxel space and then collapse them to the BEV. This design considers all the possible depths/altitudes, introducing flexibility in the projection, and mitigating the distortion effect caused by information loss in imaging. The detailed steps are: **1)** Categorical Depth Distribution Network (CaDDN [31]), which is a recent and effective method to warp image feature to BEV feature, is applied to estimate the depth distribution for each image feature point. **2)** Each feature point is wrapped from the 2D image space to the 3D physic space according to the known camera parameters. **3)** The 3D voxel features are flattened to BEV features. Briefly, the warping function is unfolded as follows: for each image feature point locates at (u, v), given the estimated categorical depth d_i, and the known camera projection matrix $\mathbf{P} \in \mathbb{R}^{3\times 4}$, 3D physical space coordinates $[x, y, z]^T$ is calculated conditioned on the image feature coordinates $[u, v, d_i]^T$ based on the projection function: $[u, v, d_i]^T = \mathbf{P} \cdot [x, y, z, 1]^T$.

Spatial confidence-aware message packing. Fig. 9 presents the detail about the spatial confidence-aware message packing module. For the message from agent i to agent j at kth communication round, the module takes the spatial confidence map $\mathbf{C}_i^{(k)}$ of agent i and the request map $\mathbf{R}_j^{(k-1)}$ of agent j

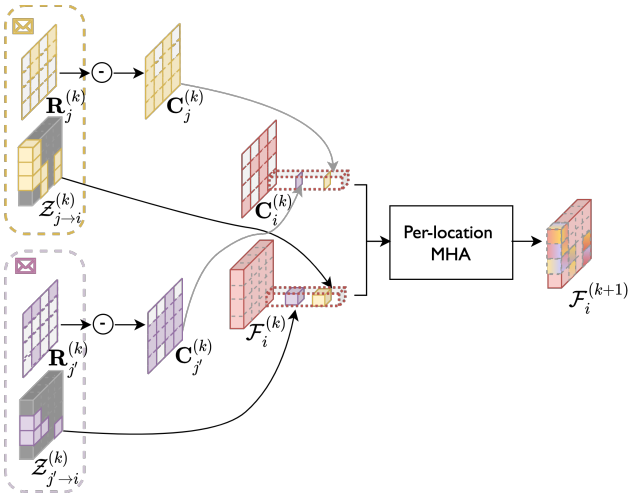


Figure 11: Spatial confidence-aware message fusion module. Each agent attentively augments the features with the received messages at each location. And the per-location multi-head attention are separately operated at each location in parallel, it takes the features and the corresponding confidence scores as input, and outputs the augmented features.

as input, and outputs the message $\mathcal{P}_{i\to j}^{(k)}$ including the masked feature map $\mathcal{Z}_{i\to j}^{(k)}$ and the request map of agent i.

Spatial confidence-aware communication graph construction. Fig. 10 presents the comparisons on the communication graph with previous works. *Fully connected* versus *agent-level partially connected* versus ours *spatial-decouple partially connected* communication. *Fully connected* communication results in a large amount of bandwidth usage, growing on the order of $O(N^2)$, where N is the number of agents in a network. *Agent-level partially connected* communication prune irrelevant connections between agents while may erroneously sever the information connection. *Spatial-decouple partially connected* communication could further flexibly prune irrelevant connections per-location and can substantially reduce the overall network complexity.

Spatial confidence-aware message fusion. Fig. 11 presents the detail about the spatial confidence-aware message fusion module. Given the received messages $\{\mathcal{P}_{j\to i}^{(k)}, j \in \mathcal{N}_i\}$, each agent i attentively augments the features with the received messages at each location. And the request map $\mathbf{R}_j^{(k)}$ in the received message is firstly decoded to the confidence map $\mathbf{C}_j^{(k)}$ via a point-wise minus. Then the per-location multi-head attention are separately operated at each location in parallel, it takes the features and the corresponding confidence scores as input, and outputs the augmented features.

Sensor positional encoding. Sensor positional encoding is conditioned on the physical distance between the known sensor coordinates and each BEV gird's coordinate in the 3D physic space. It is introduced to provide spatial prior, as the smaller the sensing distance is, the clear the observation would be. Mathematically, similar to the position encoding in [36], our sensor positional encoding is given by $SPE_{(dis,2p)} = sin(dis/10000^{2p/D})$, $SPE_{(dis,2p+1)} = cos(dis/10000^{2p/D})$ where dis is the physical distance, p is the dimension, D is the total channel dimension of the BEV feature map, sin and cos denote the sine and cosine functions.

7.5 Detailed information about experimental settings

Implementation details. For camera-only 3D object detection task on OPV2V, we implement the detector following CADDN [31]. The model is trained 100 epoch with initial learning rate of 1e-3, and decay by 0.1 at epoch 80. For LiDAR-based 3D object detection task, our detector follows MotionNet [33]. We train 120 epoch with learning rate 1e-3. For the camera-only 3D object detection task on CoPerception-UAVs, our detector follows the CenterNet [28] with DLA-34 [37] backbone. The model is trained 140 epoch with learning rate 5e-4.

Table 3: Overall performance on CoPerception-UAVs, OPV2V, V2X-Sim and DAIR-V2X. Comm denotes the communication volume calculated with Equation (5).

Dataset	CoPerception-UAVs		OPV2V		V2X-Sim1.0		DAIR-V2X	
Method/Metric	Comm	AP@0.50/0.70	Comm	AP@0.50/0.70	Comm	AP@0.50	Comm	AP@0.50/0.70
No Collaboration	0.00	57.67/29.52	0.00	22.65/9.09	0.00	45.80	0.00	50.03/43.57
Late Fusion	15.77	53.12/37.88	11.87	8.24/3.84	8.83	46.70	11.45	53.12/37.88
When2com	28.37	61.63/33.55	22.28	19.69/8.29	20.00	46.70	22.62	51.12/36.17
V2VNet	29.95	59.82/33.14	23.87	37.47/14.67	21.58	55.30	24.21	56.01/42.25
V2X-ViT	28.37	59.12/41.57	22.28	39.82/16.43	20.00	57.30	22.62	54.26/43.35
DiscoNet	28.37	59.74/29.71	22.28	36.00/12.50	20.00	58.00	22.62	54.29/44.88
	11.76	60.19/34.94	5.67	40.11/15.36	6.70	47.60	11.40	50.98/39.11
	14.27	60.23/34.93	15.49	42.15/16.09	8.29	49.10	15.58	51.01/39.10
	15.73	61.30/35.29	16.13	43.37/16.84	9.52	50.60	17.03	53.53/40.70
	17.96	63.04/36.10	17.04	44.07/17.15	10.41	51.80	17.53	55.84/42.44
Where2comm	19.04	63.94/37.16	17.86	44.68/17.77	11.10	54.20	18.19	58.46/44.46
	21.62	65.10/38.98	18.43	45.23/18.02	12.27	56.60	20.56	63.54/48.78
	23.33	65.32/39.25	18.92	46.04/18.23	12.80	57.00	21.78	63.76/48.94
	25.31	65.46/39.27	18.92	46.04/18.23	13.98	58.90	22.35	63.71/48.89
	28.48	65.71/39.38	22.71	47.14/19.07	20.00	59.10	22.62	63.71/48.93

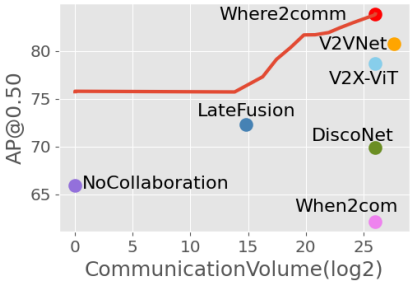


Figure 12: Where2comm achieves consistently superior performance-bandwidth trade-off on V2X-Sim2.0 [9].

Table 4: Overall performance on V2X-Sim2.0 [9]. Comm denotes the communication volume calculated with Equation(5). Metric AP@(0.50/0.70) is used.

Method/Metric	Comm	AP@0.50/0.70	Method	Comm	AP@0.50/0.70
No Collaboration	0.00	65.93/51.79	**Where 2comm**	13.84	75.72/65.13
Late Fusion	14.84	72.33/62.12		17.47	79.14/67.05
When2com	26.04	62.15/49.42		19.84	81.69/70.79
V2VNet	27.62	80.80/71.22		21.98	81.94/72.10
V2X-ViT	26.04	78.73/63.17		24.13	82.99/73.05
DiscoNet	26.04	69.73/55.12		25.93	83.77/74.09

Inference strategy in multi-round setting. For the single-round communication, all the communication budget are used in this broadcast communication round. For the two-round communication, a small bandwidth (about 20%) is allocated to activate the collaboration; for the next round, the remained relatively large (about 80%) bandwidth is allocated to transmit the targeted information to meet agents' request. For more than two rounds communication setting, we strategically allocate communication budget across multiple communication rounds. For the initial broadcast round, a small bandwidth (about 20%) is allocated to activate the collaboration; for the next round, a relatively large (about 60%) bandwidth is allocated to transmit the targeted information to meet agents' request; then, the bandwidth is gradually reduced, accounting for the communication degradation with the increasing rounds.

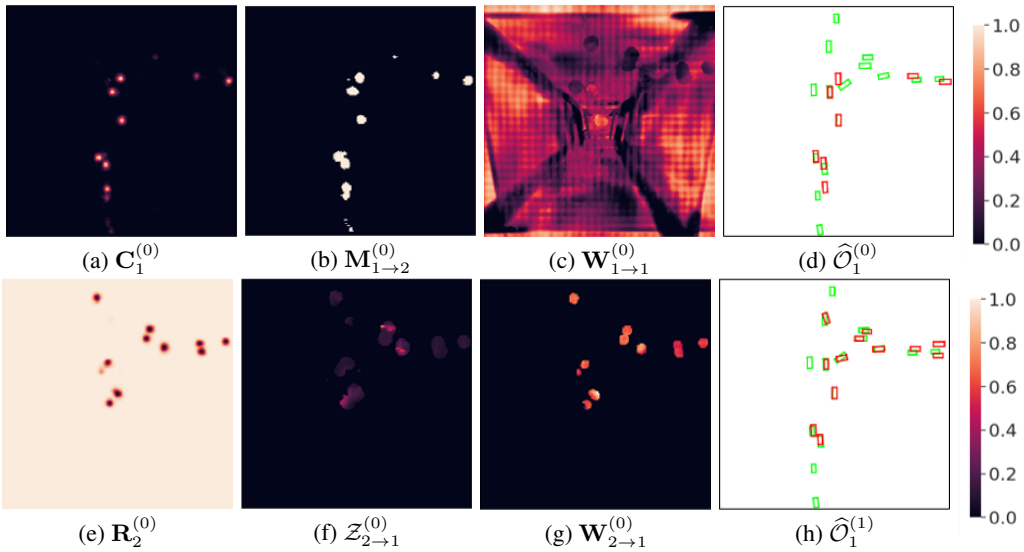
Communication volume. Our communication volume is the same as DiscoNet [2], the only difference is that our log base is 2, while it is 10, so our number is about 3.32 times theirs. The base 2 is chosen to align with the metric bit/byte, this is, communication volume counts the message size by byte in log scale with base 2. Mathematically for the selected sparse feature map $\mathcal{Z}_{i\to j}^{(k)} = \mathbf{M}_{i\to j}^{(k)} \odot \mathcal{F}_i^{(k)} \in \mathbb{R}^{H\times W\times D}$, the communication volume is

$$\log_2\left(|\mathbf{M}_{i\to j}^{(k)}| \times D \times 32/8\right), \tag{5}$$

where $|\cdot|$ denotes the L0 norm counting the non-zero elements in the binary selection matrix, this is, the total spatial girds need to be transmitted, and for each feature point D denotes the channel dimension, 32 is multiplied as float32 data type is used to represent each number, 8 is divided as the metric byte is used.

7.6 Benchmarks

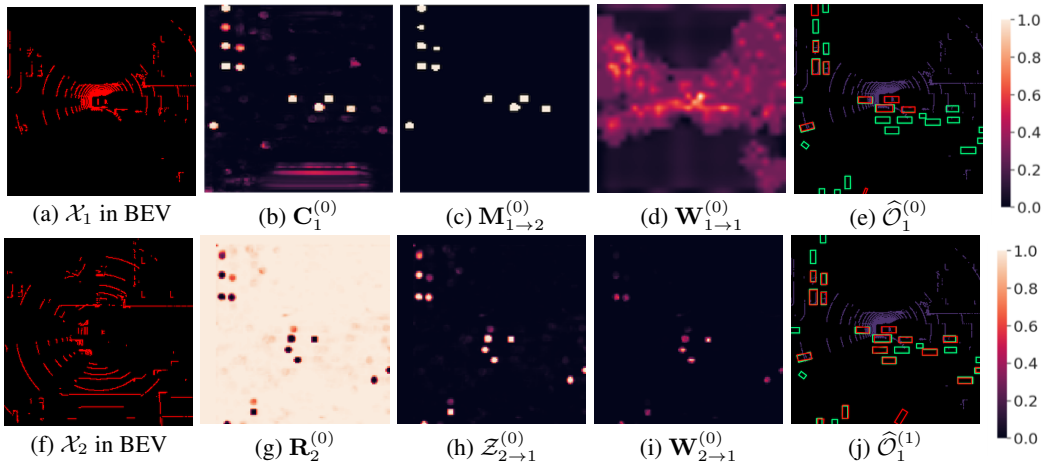
We conduct extensive experiments on all the available collaborative perception benchmarks. Tab. 3 presents the overall performance on the four datasets, CoPerception-UAVs, OPV2V [10], V2X-



(a) $\mathbf{C}_1^{(0)}$ (b) $\mathbf{M}_{1\to2}^{(0)}$ (c) $\mathbf{W}_{1\to1}^{(0)}$ (d) $\widehat{\mathcal{O}}_1^{(0)}$

(e) $\mathbf{R}_2^{(0)}$ (f) $\mathcal{Z}_{2\to1}^{(0)}$ (g) $\mathbf{W}_{2\to1}^{(0)}$ (h) $\widehat{\mathcal{O}}_1^{(1)}$

Figure 13: Visualization of collaboration between Vehicle 1 and Vehicle 2 on OPV2V dataset, including spatial confidence map ($\mathbf{C}_1^{(0)}$), selection matrix ($\mathbf{M}_{1\to2}^{(0)}$), message ($\{\mathbf{R}_2^{(0)}, \mathcal{Z}_{2\to1}^{(0)}\}$) in the communication module, attention weight in the fusion module ($\mathbf{W}_{1\to1}^{(0)}$,$\mathbf{W}_{2\to1}^{(0)}$), and Vehicle 1's detection results before ($\widehat{\mathcal{O}}_1^{(0)}$) and after ($\widehat{\mathcal{O}}_1^{(1)}$) collaboration. Green and red boxes denote ground-truth and detection, respectively. The objects occluded can be detected through transmitting spatially sparse, yet perceptually critical message.



(a) $\mathcal{X}_1$ in BEV (b) $\mathbf{C}_1^{(0)}$ (c) $\mathbf{M}_{1\to2}^{(0)}$ (d) $\mathbf{W}_{1\to1}^{(0)}$ (e) $\widehat{\mathcal{O}}_1^{(0)}$

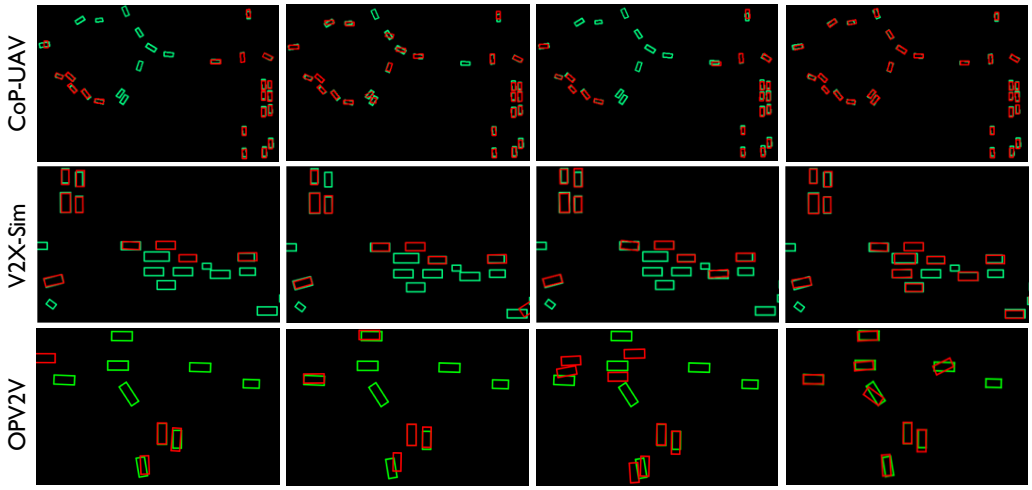
(f) $\mathcal{X}_2$ in BEV (g) $\mathbf{R}_2^{(0)}$ (h) $\mathcal{Z}_{2\to1}^{(0)}$ (i) $\mathbf{W}_{2\to1}^{(0)}$ (j) $\widehat{\mathcal{O}}_1^{(1)}$

Figure 14: Visualization of collaboration between Vehicle 1 and Vehicle 2 on V2X-Sim dataset, including spatial confidence map ($\mathbf{C}_1^{(0)}$), selection matrix ($\mathbf{M}_{1\to2}^{(0)}$), message ($\{\mathbf{R}_2^{(0)}, \mathcal{Z}_{2\to1}^{(0)}\}$) in the communication module, attention weight in the fusion module ($\mathbf{W}_{1\to1}^{(0)}$,$\mathbf{W}_{2\to1}^{(0)}$), and Drone 1's detection results before ($\widehat{\mathcal{O}}_1^{(0)}$) and after ($\widehat{\mathcal{O}}_1^{(1)}$) collaboration. Green and red boxes denote ground-truth and detection, respectively. The objects occluded by a tall building can be detected through transmitting spatially sparse, yet perceptually critical message.

Sim1.0 [2] and DAIR-V2X [11]. And we further benchmark the updated V2X-Sim2.0 [9] in Fig. 12 and Tab. 4. For this LiDAR-based 3D object detection task, our detector follows PointPillar [35]. We see that `where2comm` consistently achieves significant improvements over previous methods on all the benchmarks.

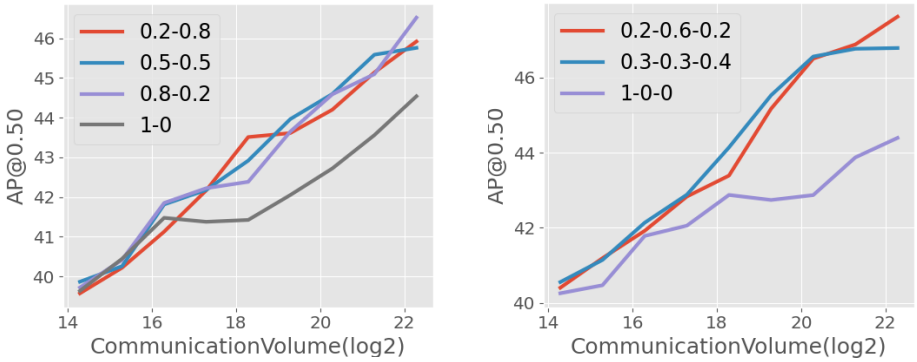
7.7 Visualization

Visualization of collaboration in OPV2V and V2X-Sim. Fig. 13 and Fig. 14 illustrates how `Where2comm` is empowered by the proposed spatial confidence map on OPV2V and V2X-Sim dataset. In the scene, with Vehicle 2's help, Vehicle 1 is able to detect the missed objects in the



(a) No Collaboration (b) When2com (c) DiscoNet (d) Where2comm

Figure 15: Where2comm qualitatively outperforms the state-of-the-art methods in CoPerception-UAVs, V2X-Sim and OPV2V datasets. Green and red boxes denote ground-truth and detection, respectively.



(a) 2-rounds communication. (b) 3-rounds communication.

Figure 16: Bandwidth allocation ablation study in multi-round communication. (a-b) shows the perception performance and communication bandwidth trade-offs for 2- and 3-round communication using different bandwidth allocation strategies on the OPV2V dataset. The legend shows the bandwidth ratio from the initial communication round to the entire communication round. Allocating more bandwidth in the second and subsequent communication rounds achieves a better performance-bandwidth trade-off than allocating all bandwidth in the initial communication round.

single view. Fig. 13 (a-d) shows Vehicle 1's spatial confidence map, binary selection matrix, ego attention weight, and the detection results by its own observation. Fig. 13 (e-f) shows Vehicle 2's message sent to Drone 1, including the request map (opposite of confidence map) and the sparse feature map, achieving efficient communication. Fig. 13 (g) shows the attention weight for Vehicle 1 to fuse Vehicle 2's messages, which is sparse, yet highlights the objects' positions. Fig. 13 (d) and (h) compares the detection results before and after the collaboration with Vehicle 2. We see that the proposed spatial confidence map contributes to spatially sparse, yet perceptually critical message, which effectively helps Vehicle 1 detect occluded objects.

Visualization of detection results. Fig. 15 shows that Where2comm qualitatively outperforms the state-of-the-art methods in CoPerception-UAVs, V2X-Sim and OPV2V datasets.

7.8 Ablation on bandwidth allocation

Fig. 16 shows the bandwidth allocation ablation study in multi-round communication setting. We see that allocating more bandwidth in the second and subsequent communication rounds achieves a better

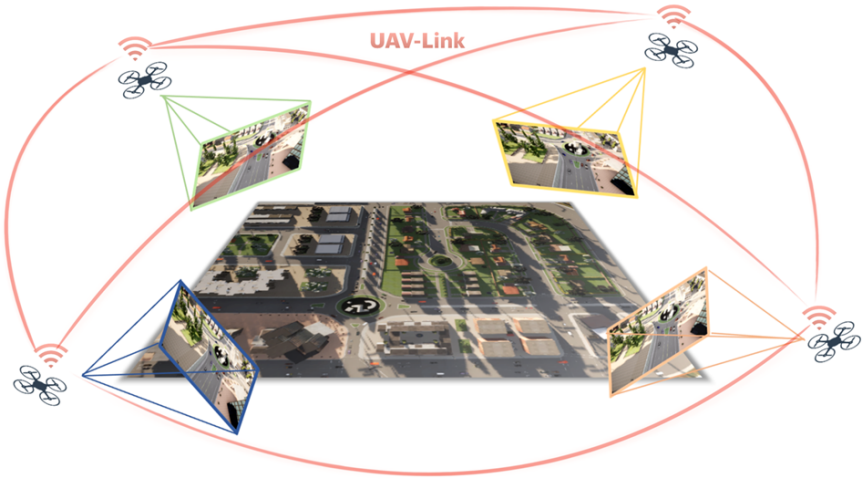


Figure 17: As an important component of the UAV swarm, collaborative perception could fundamentally resolve various reception-field restrictions in the traditional single-agent perception.

performance-bandwidth trade-off than allocating all bandwidth in the initial communication round, and the gain is stable for different bandwidth allocation strategies. The reason is that multi-round communication employs a request map in the second and subsequent communication rounds to denote the spatial area where each agent needs more information, which enables more targeted and efficient communication.

7.9 Discussion on the realistic limitations

There are many challenges in a collaborative perception system. In this work, we focus on the biggest 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 [13, 12, 1, 2]. Because 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. *Where2comm* 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 done.
• For other realistic communication issues such as **latency**, *where2comm* communicates strategically when necessary, rather than all the time or everywhere, to reduce the possibility of encountering communication problems. In addition, a prediction module could be integrated to estimate the missed or delayed frames according to the historically received frames. And by focusing on the informative spatial regions, *where2comm* can reduce the estimation difficulty.
• For the **time synchronisation** issue, by using the powerful transformer architecture-based fusion module, *where2comm* can attentively augment the features with the received asynchronous features from other agents. In addition, *where2comm* can introduce positional encoding conditioned on delay time and easily extend to global multi-head attention to further reduce the effects of time synchronization.
• For the **noisy localization** issue, *where2comm* exchanges the intermediate features among agents, which has a relatively low spatial resolution, thus is relatively robust to noisy pose. In addition, *where2comm* can easily extend to a deformable transformer architecture like [38] to further alleviate the feature distortion caused by the noisy localization.
• For the **attack** issue, by focusing on specific spatial regions and attentively fusing the received features from other agents, *where2comm* is relatively less likely to be attacked.
• For the **data availability**, *where2comm* works on both RGB and point cloud modalities, and is sensor friendly, so it can be deployed on cheap camera sensors and lidar sensors.

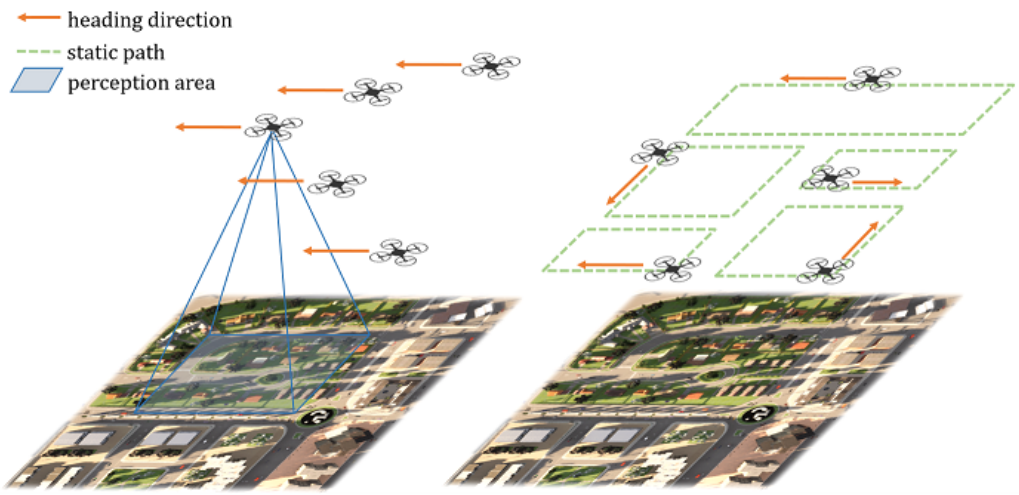


Figure 18: Two types of UAV swarm formation. The left shows the discipline formation mode, where the swarm keeps a static array and the right shows the dynamic formation mode, where each UAV navigates independently in the scene.

7.10 CoPerception-UAVs dataset details

CoPerception-UAVs dataset collects data from drones, see Fig. 17. As the rapid development of an unmanned aerial vehicle (UAV) significantly enhances human's ability to perceive the world from an aerial perspective. UAV-based systems have been widely used in numerous applications, including search and rescue, security and surveillance, photography, geographical mapping, as well as traffic monitoring. Through collaboration, UAV swarm can further distribute multiple tasks and achieve higher flexibility, stronger robustness, and a larger perception range, leading to significant advantages in harsh and complex environments. Unfortunately, collaborative perception mainly focuses on the vehicles and ignores the UAV literature. To **provide more diverse views and challenging benchmark for the collaborative perception community**, here we present the first comprehensive large-scale collaborative perception dataset for UAV swarm so far.

Since building a dataset in the real world is too expensive and laborious, in this initial version, we consider a virtual dataset based on the co-simulation of AirSim [34] and Carla [30], where AirSim simulates the UAV swarms and Carla simulates the complex background scenes and dynamic foreground objects. In the simulation, we consider that the UAV swarm is flying over diverse simulated scenes at various altitudes. Each UAV has a sensing device to collect RGB images, a computation device to perceive the environment with a perception model, and a communication device to transmit perception information among UAVs. In this setting, the UAV swarm is able to achieve 2D/3D object detection, pixel-wise or bird's-eye-view (BEV) semantic segmentation in a collaborative manner. Our dataset consists of 131.9k synchronous images collected from 5 coordinated UAVs flying at 3 altitudes over 3 simulated towns with 2 types of swarm formation. To enable the model training and testing, each image is fully annotated with the pixel-wise semantic segmentation labels, 2D bounding boxes of vehicles, as well as 3D bounding boxes on the ground and the semantic mask from BEV view. This benchmark can enable the evaluation of collaborative perception methods on the important perception tasks: 2D/3D/BEV object detection and semantic segmentation. The dataset details as unfolded as follows.

Data collection. Our proposed dataset is collected by the co-simulation of CARLA [30] and AirSim[34] (both under MIT license). We use CARLA to generate complex simulation scenes and traffic flow; and use AirSim to simulate UAV swarm flying in the scene. The flight route of UAVs is controlled by AirSim and sample data are collected randomly at about 4-second intervals.

Map creation. The simulation scenes, including the road layout, static objects, and traffic flow, are created based on CARLA [30] simulation. We take three open-source maps (*town4* to *town6*) provided by CARLA as the basic road layouts, which are the three largest maps in scale. To increase the complexity and diversity of the scenes and make the perception tasks more challenging, we customize the original maps, adding and replacing various buildings, vegetation, roadblocks, barriers, and other static objects with various assets provided by CARLA.

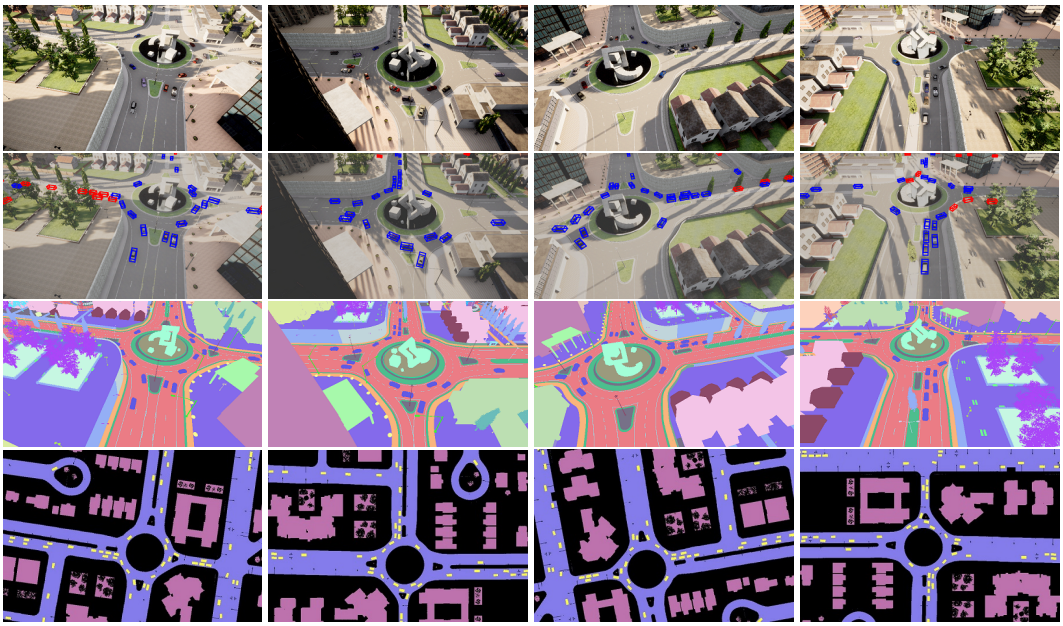


Figure 19: Data and annotations of one sample. From top to bottom: RGB image, image with 3D bounding boxes, image with semantic labels, and BEV map with semantic labels. From left to right are from different cameras equipped on four UAVs.

Traffic flow creation. Moving vehicles in the scene are managed through CARLA. Hundreds of vehicles are spawned in each scene by script *spawn_npc.py* provided by CARLA. The initial location and motion trajectory of each vehicle is determined by the map's road layout.

Sensor setup. Each UAV is equipped with 5 RGB cameras in 5 directions and 5 semantic cameras collecting semantic ground truth for RGB cameras. The cameras include a bird's eye view camera and four cameras facing forward, backward, right, and left with a pitch degree of $-45°$. Each camera has an FoV of $90°$ and the resolution is 800×450. On each UAV, all the cameras are fixed and their internal relative position and rotation degree are invariable. The translation (x, y, z) and rotation (w, x, y, z in quaternion) of each camera in both global and ego coordinates are recorded during data collection. With such a sensor setting, a UAV at the height of $40m$ can mostly cover an area of $200m \times 200m$.

Formation flying. The UAV swarm moves and executes tasks in the three-dimensional space, where the situation could be much more complex than those of vehicles or roadside units. In our proposed dataset, we take into consideration two main factors that may affect the perception and collaboration patterns of UAV swarms: flight formation and altitude. Each UAV swarm consists of 5 UAVs. We arrange two types of formation modes for a UAV swarm: discipline mode, where all 5 UAVs keeps a consistent and relatively static array, and dynamic mode, where each UAV navigates independently in the scene; see Fig. 18. The former simulates the situation where the swarm of UAVs is executing a same specific task such as exploring an unknown area, search and rescue; while the latter simulates the monitoring and patrolling tasks in the city.

Fully-annotated data are provided in our proposed dataset, including synchronous images with pixel-wise semantic labels, 2D & 3D bounding boxes of vehicles, and BEV semantic map; see 19.

Camera data. We collect synchronous images from all cameras on 5 UAVs, which is 25 images in a sample. Camera intrinsics and extrinsics in global coordinate are provided to support coordinate transformation across various UAVs. In total, 123.8K images are collected for the discipline swarm mode and 8.1K for the dynamic swarm mode.

Bounding boxes. During data collection, 3D bounding boxes of vehicles are recorded at the same moment with images, 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.

Then we provide 2D bounding boxes by projecting the 3D bounding boxes to the image perspective plane of each camera, resulting in 1.94M 3D bounding boxes and 3.6M 2D bounding boxes in total.

Data usage. In total, CoPerception-UAVs has 131.9K aerial images and 1.94M 3D boxes. We randomly split the samples into train/validation/test, resulting 91,175/19,500/20,250 images, and 1,316,536/303,888/319,576 3D bounding boxes. The dataset is organized in a similar way with the widely-used autonomous driving dataset, nuScenes [39]; so it can be used directly with the well-established nuScenes-devkit.