UMC: A Unified Bandwidth-efficient and Multi-resolution based Collaborative Perception Framework

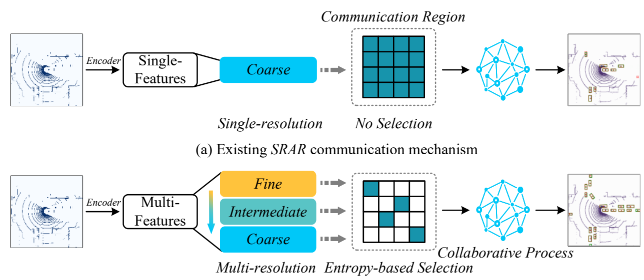
Tianhang Wang[1], Guang Chen[1,✉,*], Kai Chen[1], Zhengfa Liu[1], Bo Zhang[2], Alois Knoll[3], Changjun Jiang[1]
[1]Tongji University, [2]Shanghai Westwell Technology Co., Ltd, [3]Technische Universität München
{tianya_wang,guangchen,14sdck,1811466,cjjiang}@tongji.edu.cn
bob.zhang@westwell-lab.com knoll@in.tum.de

Abstract

*Multi-agent collaborative perception (MCP) has recently attracted much attention. It includes three key processes: communication for sharing, collaboration for integration, and reconstruction for different downstream tasks. Existing methods pursue designing the collaboration process alone, ignoring their intrinsic interactions and resulting in suboptimal performance. In contrast, we aim to propose a **U**nified **C**ollaborative perception framework named UMC, optimizing the communication, collaboration, and reconstruction processes with the **M**ulti-resolution technique. The communication introduces a novel trainable multi-resolution and selective-region (MRSR) mechanism, achieving higher quality and lower bandwidth. Then, a graph-based collaboration is proposed, conducting on each resolution to adapt the MRSR. Finally, the reconstruction integrates the multi-resolution collaborative features for downstream tasks. Since the general metric can not reflect the performance enhancement brought by MCP systematically, we introduce a brand-new evaluation metric that evaluates the MCP from different perspectives. To verify our algorithm, we conducted experiments on the V2X-Sim and OPV2V datasets. Our quantitative and qualitative experiments prove that the proposed UMC outperforms the state-of-the-art collaborative perception approaches.*

1. Introduction

Single-vehicle perception has made remarkable achievements in object detection[20, 30, 31], segmentation[28, 32], and other tasks with the advent of deep learning. However, single-vehicle perception often suffers from environmental conditions such as occlusion[38, 49] and severe weather[4, 16, 52], making accurate recognition challenging. To overcome these issues, several appealing studies have been devoted to collaborative perception[1, 2, 7, 41, 48], which take advantage of sharing the multiple-viewpoint of the same scene with the Vehicle-to-Vehicle(V2V) communication[3].



(a) Existing *SRAR* communication mechanism

(b) Our proposed *MRSR* communication mechanism

Figure 1. **The SRAR versus the proposed MRSR mechanism.** (a) The existing single-resolution and all-region (SRAR) method utilizes the coarse-grained feature with bandwidth inefficient all-region transmission. (b) The proposed MRSR mechanism incorporates bandwidth-efficient entropy-based selection with multi-resolution features.

To design a collaborative perception algorithm, current approaches[39, 23, 44, 18, 43] mainly focus on the collaboration process alone, aiming to design a high-performance collaboration strategy. Nevertheless, such a strategy ignores intrinsic interactions with two critical processes: communication[22, 12] and reconstruction[9]. This kind of collaboration strategy will inherently cause suboptimal performance because the quality of communication directly determines the performance of collaboration[18], and the reconstruction influences the quality of feature maps for downstream tasks. Meanwhile, the collaboration can also affect the design of communication and reconstruction. Hence, failing to optimize these three processes will degrade the system.

In this paper, to the best of our knowledge, we are the first to propose a unified collaborative perception framework that optimizes the communication, collaboration, and reconstruction processes with multi-resolution technique.

*Corresponding author: guangchen@tongji.edu.cn. Our code is available at:https://github.com/ispc-lab/UMC.

As for communication, as shown in Figure 1, we propose a novel multi-resolution and selective-region (MRSR) mechanism, which is different from the single-resolution and all-region (SRAR) mechanism that is used by existing collaborative algorithms[18, 44, 39]. In MRSR, we replace the coarse-grained feature in SRAR with multi-grain features, which reflect the scene from a global to a local perspective. The multi-grain features will have the complementary advantages of global structure information and local texture details. Meanwhile, to reduce the burden of bandwidth, unlike SRAR, which blindly transmits all regions, we implement a trainable and region-wise entropy-based communication selection (Entropy-CS) module that highlights the informative regions and selects appropriate regions for each level of resolution. The proposed MRSR is adaptive to real-time communication, reflecting dynamic connections among agents.

We propose a graph-based collaborative GRU(G-CGRU) module for MRSR's each level resolution feature in collaboration. Compared to the existing collaboration strategies that only focus on the present shared information without considering the previous state, we redesign the GRU-cell[5] to adapt the time series of collaborative perception, which models the continuity of vehicle movement. In G-CGRU, the collaboration depends not only on collaborators but also on the last moment information of the ego agent. Meanwhile, we propose matrix-valued gates in G-CGRU to ensure collaboration at a high spatial resolution and allow the agents to adaptively weight the informative regions.

To adapt the proposed multi-resolution collaboration mechanism for reconstruction, we propose a multi-grain feature enhancement (MGFE) module to strengthen the feature reconstruction process. In MGFE, the fine-grained collaborative feature maps will give direct guidance to the agents' feature maps via a global pooling operation, and the coarse-grained collaborative feature maps will again enhance the agents' feature maps from a global perspective. This design allows the agents to comprehensively reconstruct the feature maps from local and global viewpoints.

The general evaluation metric (*e.g., average precision*) can reflect the overall performance of multi-agent collaborative perception (MCP). However, the performance enhancement brought by the collaboration concept that MCP introduced cannot be reflected directly. To address this issue, we introduce a brand new evaluation metric that systematically evaluates the MCP from four aspects.

To validate the proposed framework, we conducted comprehensive experiments in 3D object detection on V2X-Sim[17] and OPV2V[45] datasets. Our proposed unified, bandwidth-efficient and multi-resolution based collaborative perception framework (UMC) achieves a better performance-bandwidth trade-off than the state-of-the-art SRAR-based collaboration methods. Our contributions are listed as follows:

- We present a unified, bandwidth-efficient framework (UMC) for collaborative perception, which optimizes the communication, collaboration, and reconstruction processes with the multi-resolution technique.
- We propose a novel multi-resolution and selective-region mechanism for communication and the graph-based collaborative GRU for each resolution collaboration and multi-grain feature enhancement module for reconstruction.
- We present a brand new evaluation metric for 3D object collaborative perception, which can evaluate the performance from different perspectives.

2. Related Work

2.1. V2X Collaborative Perception

F-Cooper[1] is the first to apply feature-level collaboration, which weights interaction information equally using a max-based function. When2com[22] employs an attention mechanism to build a bandwidth-efficient communication group. V2VNet[39] proposes a spatially-aware message transmission mechanism that assigns varying weights to the different agents. DiscoNet[18] adapts the knowledge distillation to enhance student model performance by constraining the teacher model, which is based on corresponding raw data collaboration. V2X-ViT[44] presents a unified V2X framework based on Transformer that takes into account the heterogeneity of V2X systems. Where2comm[12] takes the advantage of sparsity of foreground information in detection downstream task to save the communication bandwidth of V2X collaboration. CoBEVT[43] generates BEV map predictions with multi-camera perception by V2X.

2.2. 3D Perception for Autonomous Driving

Due to the declining price of commodity LiDAR sensors, 3D perception has become more prevalent in autonomous driving. VoxelNet[53] pioneered utilizing 3D object detection in autonomous driving. More and more advanced algorithms have been discovered. There are three branches of 3D perception input forms that have been widely adopted: point-based methods[28, 36, 29] directly process raw captured point; and voxel-based methods[53, 46, 13, 35] divide the points into a volumetric grid; and 2D projections based methods[37, 47, 14] compact the input by projecting points into images. In UMC, we utilize bird's eye view projection as input, which decreases the processing time and enables the use of 2D object detection networks.

2.3. Multi-resolution Image Inpainting

Multi-resolution is a widely used image inpainting[50, 27, 40] technique to combine low-resolution feature

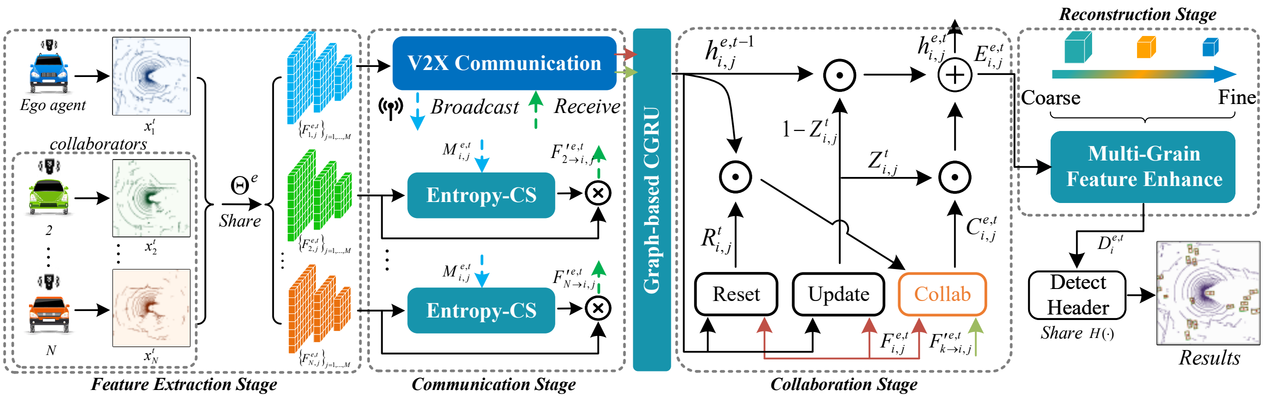


Figure 2. The overview of the proposed UMC framework. i) ***Feature extraction stage***, the agents obtain the $\boldsymbol{F}_i^{e,t}$ by the shared feature encoder Θ^e with the observation $\boldsymbol{x}_i^t$. ii) ***Communication stage***, the ego agent (in **blue**) will broadcast compact query matrix $\boldsymbol{M}_i^{e,t}$ in each resolution to collaborators by V2X communication, and the collaborators will compute the transmission map by entropy-CS module at local. Then the ego agents will receive the selected messages from the assisted collaborators. iii) ***Collaboration stage***, the ego agents will employ the G-CGRU module in each resolution for high efficient collaboration. iv) ***Reconstruction stage***, the MGFE module will reconstruct the ego agent's feature by multi-resolution collaborative feature maps for different downstream tasks.

maps that contain global structure information with high-resolution feature maps that are rich in local texture details, such as, Lin *et al*. [20] firstly introduce FPN to improve the scale invariance. PEN-Net[51] proposes the attention map mechanism, which learns from high-resolution feature maps to guide low-resolution features. Wang *et al*. [40] design a parallel multi-resolution fusion network that can effectively fuse structure and texture information. DFNet[11] employs U-Net[33] based multiple fusion blocks to implement multi-resolution constraints for inpainting. In UMC, we implement the multi-resolution technique for a new application scenario: multi-agent collaborative perception. The multi-grain collaborative feature makes it possible for the agents to reconstruct the occlusion area from both a local and a global point of view.

3. UMC

In this section, we present the technical details of the proposed UMC. The overall architecture of the proposed UMC is shown in Figure 2, which consists of three new modules, entropy-CS, G-CGRU, and the MGFE. Entropy-CS extends the traditional entropy theory to select the informative regions of observations, reducing the heavy bandwidth burden brought by multi-resolution. G-CGRU redesigns the GRU-Cell to adapt the multi-agent collaboration, which models the continuity of vehicle movement. MGFE introduces the multi-grain feature enhancement to strengthen the feature reconstruction process for different downstream tasks.

3.1. Problem Statement and Notation

We assume there are N collaborators with their corresponding observations $\boldsymbol{X}^t = \{x_n^t\}_{n=1,...,N}$ at time t. Among those collaborators, the feature set of the i-th agent is defined as $\boldsymbol{F}_i^{e,t} = \{\boldsymbol{F}_{i,j}^{e,t}\}_{j=1,...,M} \leftarrow \Theta^e(x_{i,t})$, where $\Theta(\cdot)$ is the feature encoder shared by all the collaborators and M represents the number of intermediate layers in $\Theta(\cdot)$. Note that because each agent has its unique pose $\boldsymbol{\xi}^t$, we need conduct feature alignment operation $\Lambda(\cdot,\cdot)$ across agents for subsequent collaboration process, *e.g.*, for the i-th agent, the transformed feature map from the k-th agent of the j-th intermediate layer is $\boldsymbol{F}_{k\to i,j}^{e,t} \leftarrow \Lambda(\boldsymbol{F}_{k,j}^{e,t}, \boldsymbol{\xi}_{k\to i}^t)$, where the $\boldsymbol{\xi}_{k\to i}^t$ is based on two ego poses $\boldsymbol{\xi}_k^t$ and $\boldsymbol{\xi}_i^t$. Now $\boldsymbol{F}_{k\to i,j}^{e,t}, \boldsymbol{F}_{i,j}^{e,t}$ are supported in the same coordinate system.

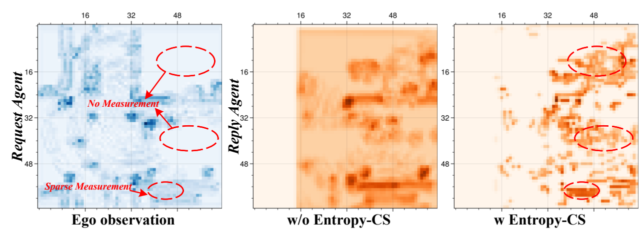


Figure 3. Illustration of entropy-based communication selection. The entropy-CS filters out the lower quality regions, achieving high efficient bandwidth usage.

3.2. Entropy-based Communication Selection

The intuition of entropy-CS is low computational complexity and high interpretability to filter out the lower-quality regions. Based on above considerations, we extend the traditional entropy theory[6] toward two-dimensional communication selection. Hence, the entropy-CS is non-parameter and high efficient to reduce the heavy bandwidth burden brought by multi-resolution feature information.

To further compress bandwidth requirements, the ego agent i and collaborator will first compress their encoder features $\boldsymbol{F}_{i,j}^{e,t}, \boldsymbol{F}_{k\to i,j}^{e,t} \in \mathbb{R}^{K,K,C}$ along channels into the query matrix $\boldsymbol{M}^{e,t}_{i,j}, \boldsymbol{M}^{e,t}_{k\to i,j} \in \mathbb{R}^{K,K}$ through a trainable 1×1 convolution kernel $\boldsymbol{W}_{1\times 1}$, which is supervised by the downstream loss function. Then, we construct the two-stage entropy-based selective-region communication mechanism:

Self-select stage We first let the k-th collaborator determine location $\boldsymbol{S}^{e,t}_{k\to k,j}$ itself, where potentially contains useful information for later collaboration:

$$\boldsymbol{S}^{e,t}_{k\to k,j} = \boldsymbol{\Gamma}(\Phi(\boldsymbol{M}^{e,t}_{k\to i,j}, \boldsymbol{M}^{e,t}_{k\to i,j}), \delta_s) \tag{1}$$

where the $\boldsymbol{\Gamma}(\cdot, \delta_s)$ is an element-wise function, zeroing out elements smaller than the values of top-$\delta_s\%$ in $\cdot$, and Φ is the entropy estimation function, which measures the correlation from $\boldsymbol{K} \in \mathbb{R}^{K,K}$ to $\boldsymbol{Q} \in \mathbb{R}^{K,K}$ at the location of $\boldsymbol{L}$. If $\boldsymbol{L}$ is not assigned, the Φ will compute all region of $\boldsymbol{K}, \boldsymbol{Q}$:

$$\begin{aligned} &\Phi(\boldsymbol{K}, \boldsymbol{Q}, \boldsymbol{L}) = \{p^{xy} * log(p^{xy})\}_{x,y\in \boldsymbol{L}}, \text{with} \\ &p^{xy} = \frac{1}{M*N}\sum_{j,k}^{M,N} \sigma(\boldsymbol{K}(x+j, y+k) - \boldsymbol{Q}(x,y)) \end{aligned} \tag{2}$$

where $\sigma(\cdot)$ is Sigmoid function and we set M, N to 3 in later experiments. Note that if $\|\boldsymbol{S}^{e,t}_{k\to k,j}\| \simeq 0$, represents that the k-th collaborator almost has no sufficient information. The entropy-CS will close the later cross-select stage, which can further reduce the bandwidth requirements.

Cross-select stage After the self-entropy selection stage, we thus derive the cross-entropy selection to obtain the communication location $\boldsymbol{S}^{e,t}_{k\to i,j}$ with the broadcast $\boldsymbol{M}^{e,t}_{i,j}$:

$$\boldsymbol{S}^{e,t}_{k\to i,j} = \boldsymbol{\Gamma}(\Phi(\boldsymbol{M}^{e,t}_{i,j}, \boldsymbol{M}^{e,t}_{k\to i,j}, \boldsymbol{S}^{e,t}_{k\to k,j}), \delta_c) \tag{3}$$

As demonstrated in Figure 3, the entries of the derived matrix $\boldsymbol{S}^{e,t}_{k\to i,j}$ indicate where to communicate, which requires about $\frac{1}{\delta_s\delta_c}$ bandwidth of existing SRAR-based methods[39, 18, 22]. However, the transmission content $\boldsymbol{T}^{e,t}_{k\to i,j} \leftarrow \boldsymbol{F}^{e,t}_{k\to i,j}[\boldsymbol{S}^{e,t}_{k\to i,j}]$ is sparse, with many positions being $\boldsymbol{0}$. This could potentially harm further collaboration process[24]. To compensate for this, after the ego agent receives, we spatially interpolate these positions from their neighbors across all channels at local, using a similar approach in [42]: $\boldsymbol{F}^{'e,t}_{k\to i,j} \leftarrow Interpolate(\boldsymbol{T}^{e,t}_{k\to i,j})$.

3.3. Graph-based Collaborative GRU

Once a requesting ego agent collects the information from its linked supporting collaborators, it is important to design how to integrate its local observation with the selected multi-resolution feature maps from supporters[18].

We observe that the detected agents in perception area change slowly over time in the urban low-speed collaborative scenes[45, 17]. Hence, the collaborative feature maps from last time are actually helpful to the present process.

However, the general RNN model[34, 10] can only process one-dimensional features, which makes it hard to preserve the spatial structure and local details of the agent's observation.

Based on the above considerations, we propose a novel graph-based collaborative GRU module named G-CGRU for each resolution collaboration process, as demonstrated in Figure 2. The key intuition of G-CGRU is that the collaboration depends not only on supporters but also on requesting the ego agent's previous information.

For the ego agent i of the j-th resolution intermediate feature maps, the inputs of G-CGRU are hidden states $\boldsymbol{h}^{e,t-1}_{i,j} \in \mathbb{R}^{K,K,C}$, the ego agent observation $\boldsymbol{F}^{e,t}_{i,j}$, and the supporters' selected feature maps $\{\boldsymbol{F}^{'e,t}_{k\to i,j}\}_{k\neq i}$. The module mainly has three key gates as follows:

Update and Reset gates The $Update$ gate is responsible for determining where the previous information needs to be passed along the next state and the $Reset$ is to decide how much of the past information is needed to neglect. The $Update$ and $Reset$ gates share the same structure but different weights. Here we take $Reset$ as an examples:

$$\begin{aligned} &\boldsymbol{W}_{ir} = \sigma(\boldsymbol{W}_{3\times 3} * ([\boldsymbol{h}^{e,t-1}_{i,j}; \boldsymbol{F}^{e,t}_{i,j}])) \\ &\boldsymbol{R}^{t}_{i,j} = \sigma(\boldsymbol{W}_{ir} \odot \hat{\boldsymbol{h}}^{e,t-1}_{i,j} + (1-\boldsymbol{W}_{ir}) \odot \boldsymbol{F}^{e,t}_{i,j}) \end{aligned} \tag{4}$$

where $\odot, \sigma(\cdot), [\cdot;\cdot]$ represent dot product, Sigmoid function, and concatenation operation along channel dimensions, respectively. $*$ indicate a 3×3 convolution operation. The $\boldsymbol{R}^{t}_{i,j} \in \mathbb{R}^{K,K,C}$ learns where the hidden features $\boldsymbol{h}^{e,t-1}_{i,j}$ are conducive to the present.

Collab gate Based on the above $Reset$ and $Update$ gates, we thus derive the $Collab$ gate. To make better collaborative feature integration, we construct a collaboration graph $\mathcal{G}^t_c(\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}})$ in $Collab$ gate, where node $\boldsymbol{\mathcal{V}} = \{\mathcal{V}_i\}_{i\in 1,\ldots,N}$ is the set of collaborative agents with the real-time pose information in the environment and $\boldsymbol{\mathcal{E}} = \{\boldsymbol{W}_{k\to i}\}_{i,k\in 1,\ldots,N, i\neq k}$ is the set of trainable edge matrix weights between agents and models the collaboration strength between two agents. Let $\mathcal{C}_{\mathcal{G}^t_c}(\cdot)$ be the collaboration process defined in the $Collab$ module's graph $\mathcal{G}^t_c$. The j-th resolution enhanced maps of ego i agent after collaboration are $\boldsymbol{E}^{e,t}_{i,j} \leftarrow \mathcal{C}_{\mathcal{G}^t_c}(\boldsymbol{h}^{e,t-1}_{i,j}, \boldsymbol{F}^{'e,t}_{k\to i,j}, \boldsymbol{F}^{e,t}_{i,j})$. This process has two stages: message attention (**S1**) and message aggregation (**S2**).

In the *message attention stage* (**S1**), each agent determines the matrix-valued edge weights, which reflect the strength from one agent to another at each individual cell. To determine the edge weights, we firstly get the conductive history information from hidden features by $Reset$ gates through $\hat{\boldsymbol{h}}^{e,t-1}_{i,j} \leftarrow \boldsymbol{h}^{e,t-1}_{i,j} \odot \boldsymbol{R}^{t}_{i,j}$. Then, we utilize the edge encode Π to correlate the history information, the feature map from another agent and ego feature map; that is, the matrix-value edge weight from k-th agent to the i-th agent is $\boldsymbol{W}_{k\to i} = \Pi(\hat{\boldsymbol{h}}^{e,t-1}_{i,j}, \boldsymbol{F}^{'e,t}_{k\to i,j}, \boldsymbol{F}^{e,t}_{i,j}) \in \mathbb{R}^{K,K}$, where Π concatenates three feature maps along the channel dimension and then utilizes four 1×1 convolutional layers to gradually reduce the number of channels from $3C$ to 1. Also, to normalize the edge weights across different agents, we implement a softmax operation on each cell of the feature map. Compared with previous work[22, 12, 44] generally consider time-discrete edge weight to reflect the static collaboration strength between two agents; while we consider time-continuous edge weight $\boldsymbol{W}_{k\to i}$. which dynamically models the collaboration strength from the k-th agent to the i-th agent from $t-1$ to t. In the *message aggregation stage* (**S2**), each agent aggregates the feature maps from collaborators based on the normalized matrix-valued edge weights, the updated feature map $\boldsymbol{C}^{e,t}_{i,j}$ is utilized by $\sum_{k=1}^{N}\boldsymbol{W}_{k\to i}\circ \boldsymbol{F}'^{e,t}_{k\to i,j}$, where $\circ$ represents the dot production broadcasting along the channel dimension. Finally, the collaborative map is $\boldsymbol{E}^{e,t}_{i,j}=\boldsymbol{Z}^{t}_{i,j}\odot\boldsymbol{C}^{e,t}_{i,j}+(1-\boldsymbol{Z}^{t}_{i,j})\odot\boldsymbol{h}^{e,t-1}_{i,j}$. Note that the $\boldsymbol{Z}^{t}_{i,j}$ is generated by the $Update$ gate.

3.4. Multi-grain Feature Enhancement Module

The final ego agent i collaborative feature maps $\{\boldsymbol{E}^{e,t}_{i,j}\}_{j=1,\dots,M}$ with different resolutions generated from G-CGRU contain various visual context information, and each of them can be used to yield the downstream task.

To make the best use of multi-grain feature maps, we further propose a multi-grain feature enhancement (MGFE) module, which utilizes coarse- to fine-grained collaborative feature maps to guide the reconstruction process from a local to global perspective, as demonstrated in Figure 4. The MGFE consists of two stages:

Stage one A 1×1 convolution is applied to adjust the feature space of the collaborative feature map $\boldsymbol{E}^{e,t}_{i,j}$ to the ego observed feature map $\boldsymbol{F}^{e,t}_{i,j}$, then through global pooling along channels to the adjusted collaborative feature to obtain highlighted informative regions, following multiplied with the local feature map. The output $\boldsymbol{f}^{1}_{s}$ as follows:

$$\boldsymbol{f}^{1}_{s}=\mathcal{G}(\sigma(\boldsymbol{W}_{1\times 1}*\boldsymbol{E}^{e,t}_{i,j}+\boldsymbol{b}))\circ\boldsymbol{F}^{e,t}_{i,j} \tag{5}$$

where $\mathcal{G},\circ$ denotes global max pooling operation along the channels and dot production broadcasting along the channel dimension, respectively.

Stage two The coarse-grained, ego-observed feature map $\boldsymbol{F}^{e,t}_{i,j-1}$ is upsampled to the same dimension as the high-resolution map, and then concatenated with the $\boldsymbol{f}^{1}_{s},\boldsymbol{E}^{e,t}_{i,j}$. The output of stage two $\boldsymbol{f}^{2}_{s}$ as follows:

$$\boldsymbol{f}^{2}_{s}=[L_2(upsample(\boldsymbol{F}^{e,t}_{i,j-1}));L_2(\boldsymbol{f}^{1}_{s});L_2(\boldsymbol{E}^{e,t}_{i,j})] \tag{6}$$

where $upsample$ denotes the deconvolution operation, L_2 refers to the L_2 norm layer[21], which is helpful for combining two (or more) feature maps. To adjust the channels dimension of $\boldsymbol{f}^{2}_{s}$, a 3×3 convolution is applied to $\boldsymbol{f}^{2}_{s}$: $\boldsymbol{F}^{e,t}_{i,j+1}\leftarrow\sigma(\boldsymbol{W}_{3\times 3}*\boldsymbol{f}^{2}_{s}+\boldsymbol{b})$.

After the reconstruction process of the MGFE module, we can obtain the ego agent enhancement feature $\boldsymbol{D}^{e,t}_{i}$. To obtain the final detection outputs $\tilde{\boldsymbol{Y}}^{t}_{i}$, we employ an output header $\boldsymbol{H}(\cdot)$: $\tilde{\boldsymbol{Y}}^{t}_{i}\leftarrow\boldsymbol{H}(\boldsymbol{D}^{e,t}_{i})$. To implement the header $\boldsymbol{H}(\cdot)$, we employ two branches of convolutional layers to classify the foreground-background categories and regress the bounding boxes.

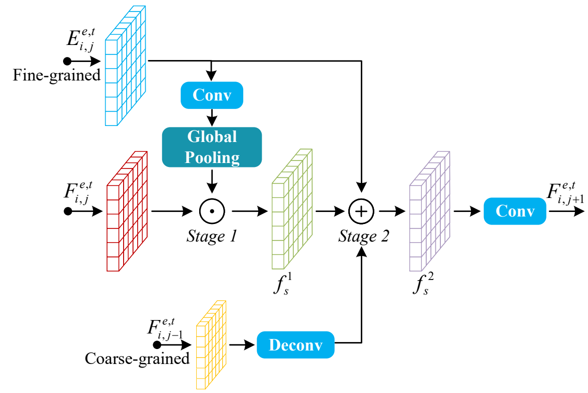


Figure 4. The architecture of multi-grain feature enhancement module. The fine- and coarse-grained feature maps guide the reconstruction process from the local to global perceptive, respectively.

To train our model, we employ the binary cross-entropy loss to supervise foreground-background classification and the smooth L_1 loss to supervise the bounding-box regression:

$$\mathcal{L}=\sum_{i=1}^{N}\mathcal{L}_{det}(\boldsymbol{Y}_i,\tilde{\boldsymbol{Y}}_i) \tag{7}$$

where $\mathcal{L}_{det}$ denotes the classification and regression losses and $\boldsymbol{Y}_i,\tilde{\boldsymbol{Y}}_i$ represents the ground-truth detection and predicted detection in the perception region of the i-th agent, respectively.

4. Experiment

4.1. Collaborative 3D Object detection dataset

To validate our proposed UMC on the LIDAR-based 3D object detection task. Same as [18, 15, 19], we utilize the multi-agent datasets of V2X-Sim[17] and OPV2V[45]. The V2X-Sim is built by the co-simulation of SUMO[25] and CARLA[8], which contains 90 scenes, and each scene includes 100 frames. The OPV2V is co-simulated by OpenCDA[45] and CARLA, including 12K frames of 3D point clouds with 230K annotated 3D boxes.

4.2. Implementation details

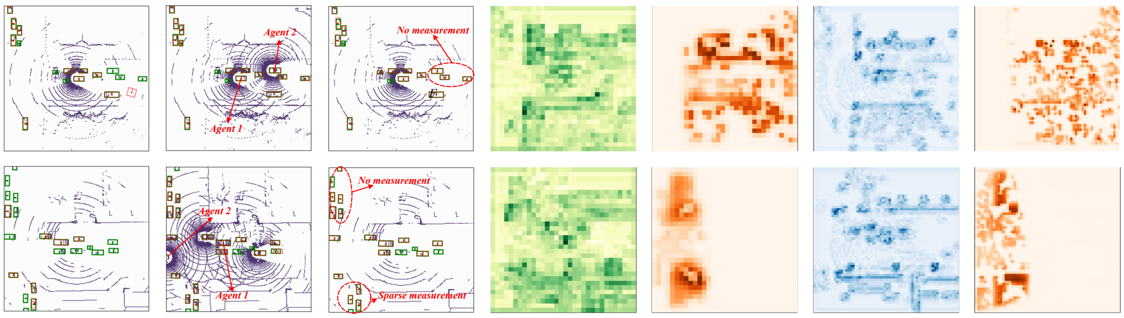
Experimental setting Same as [18, 15, 45], we crop the point clouds located in the region of $[-32m,32m]\times[-32m,32m]\times[0,5m]$ defined in the ego-vehicle Cartesian coordinate system. The size of each voxel cell is set as $0.25m\times 0.25m\times 0.4m$ to get the Bird's-Eyes view map with dimension $256\times 256\times 13$. The SRAR-based transmitted collaborative feature map (TCF) has a dimension of

Table 1. Detection comparison on V2X-Sim dataset. Key: [**Best**, **Second Best**, **Third Best**]

Models	ARSV$_{50/70}$ ↑	ARCV$_{50/70}$ ↑	ARCI$_{50/70}$ ↓	ARTC$_{50/70}$ ↑	AP$_{50/70}$ ↑
No Fusion	76.90/72.35	4.62/3.45	2.30/**1.27**	7.85/4.62	51.87/45.05
Early Fusion	**86.70**/**83.72**	**64.17**/**60.49**	9.55/7.88	**15.67**/**13.33**	**67.79**/**62.29**
Who2com	74.60/69.30	4.70/3.95	**1.80**/1.30	7.58/7.25	49.77/42.30
When2com	75.10/69.69	4.79/4.08	**1.83**/**1.28**	7.58/7.25	52.06/44.21
V2VNet	80.13/75.17	44.66/32.00	**1.33**/**0.97**	10.70/6.83	58.51/48.98
DiscoNet	**86.64**/**82.16**	**58.58**/**46.47**	3.02/1.91	**12.86**/9.32	**65.89**/**56.74**
Where2comm	81.96/77.24	44.97/30.90	11.45/8.45	**16.51**/**11.61**	62.05/54.59
UMC(ours)	**84.67**/**80.68**	**67.01**/**60.04**	2.38/1.70	12.35/**9.46**	**67.80**/**60.01**

Table 2. Detection comparison on OPV2V dataset. Key: [**Best**, **Second Best**, **Third Best**]

Models	ARSV$_{50/70}$ ↑	ARCV$_{50/70}$ ↑	AP$_{50/70}$ ↑
No Fusion	69.33/46.35	12.32/ 4.36	54.69/23.94
Early Fusion	64.62/46.95	**45.00**/**24.39**	55.88/**25.89**
Who2com	69.43/45.68	15.84/6.21	55.36/23.39
When2com	69.27/45.54	15.90/6.22	55.13/23.27
V2VNet	**73.42**/**48.21**	24.65/11.16	**59.03**/**24.71**
DiscoNet	**72.04**/46.94	**37.25**/**22.57**	55.46/23.04
Where2comm	69.79/**47.60**	13.89/6.09	**56.02**/**25.38**
UMC(ours)	**76.56**/**47.68**	**47.82**/**25.06**	**61.90**/24.50



(a) No fusion (b) Early fusion (c) UMC (d) $F_{1,1}^{e,t}$ (e) $S_{2\to 1,1}^{e,t}$ (f) $F_{1,2}^{e,t}$ (g) $S_{2\to 1,2}^{e,t}$

Figure 5. Detection and communication selection for Agent 1. The green and red boxes represent the ground truth (GT) and predictions, respectively. (a-c) shows the results of no fusion, early fusion, and UMC compared to GT. (d) The coarse-grained collaborative feature of Agent 1. (e) Matrix-valued entropy-based selected communication coarse-grained feature map from Agent 2. (f) The fine-grained collaborative feature of Agent 1. (g) Matrix-valued entropy-based selected communication fine-grained feature map from Agent 2.

$32 \times 32 \times 256$. Our proposed UMC's TCFs have the dimension of $32 \times 32 \times 256, 64 \times 64 \times 128$. We train all the models using NVIDIA RTX 3090 GPU with PyTorch[26].

Baselines We consider the single-agent perception model, called *No Fusion*, which only processes a single-view point cloud. Also, we consider the holistic-view perception model based on early collaboration, called *Early Fusion*. We consider five collaboration methods: Who2com[23], When2com[22], Where2comm[12], DiscoNet[18] and V2VNet[39]. To ensure fairness, all the methods share the same encoder and detection header architecture and collaborate at the same intermediate feature layer.

Evaluation Metrics We employ the Average Precision (AP) to validate the overall performance of the model. To systematically analyze the performance of the model, we introduce new metrics from four aspects: i) **ARSV** denotes the **A**verage **R**ecall of agents that are visible from **S**inge-vehicle **V**iew; ii) **ARCV** denotes the **A**verage **R**ecall of agents that are invisible from single-vehicle view but visible from **C**ollaborative-**V**iew, which evaluates model's performance of collaborative process; iii) **ARCI** denotes the **A**verage **R**ecall of **C**ompletely-**I**nvisible agents, which evaluates the performance of detecting agents with limited information; iv) **ARTC** denotes the **A**verage **R**ecall of the agents that were visible at the last time but not visible at this time, which evaluates model's performance of **T**ime **C**ontinuity. Note that more technical details are shown in Appendix.1, and we employ all evaluation metrics at the Intersection-over-Union (IoU) threshold of 0.5 and 0.7.

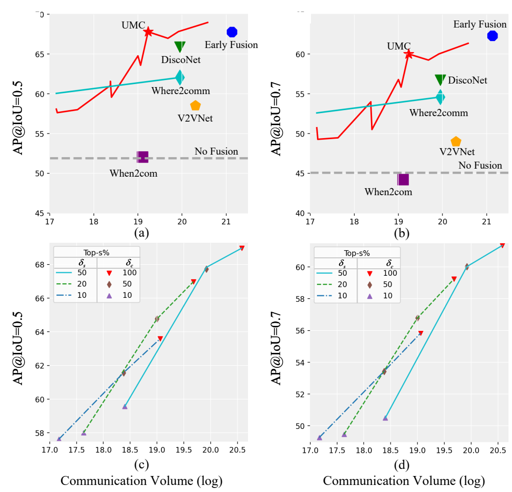


Figure 6. Performance-bandwidth trade-off in log scale.

4.3. Quantitative evaluation

Overall performance comparison Table 1 of AP$_{50/70}$ column shows the comparisons in terms of AP(@IoU=0.5/0.7). We see that i) early fusion achieves the best detection performance when AP@0.7, and there is an obvious improvement over no fusion, *i.e.*, AP@0.5 and AP@0.7 are increased by 30.69% and 38.26% respectively, revealing the effectiveness of collaboration; ii) among the SRAR-based intermediate collaboration methods, the proposed UMC achieves the best performance. Compared to When2com, UMC improves by 30.23% in AP@0.5 and 35.71% in

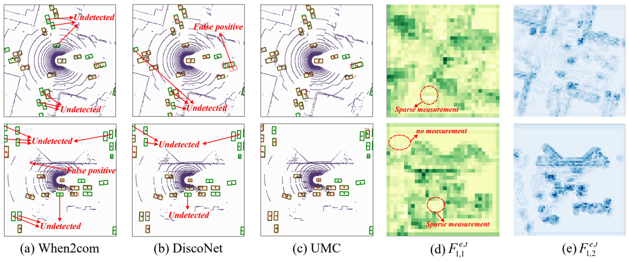


Figure 7. UMC qualitatively outperforms the state-of-the-art methods. The green and red boxes denote ground truth and detection, respectively. (a) Results of When2com. (b) Results of DiscoNet. (c) Results of UMC. (d)-(e) Agent 1's coarse-grained and fine-grained collaborative feature maps, respectively.

AP@0.7. Compared to the knowledge distillation (KD) based DiscoNet, UMC improves by 2.9% in AP@0.5 and 5.76% in AP@0.7.

Proposed new metric performance comparison From the rest column of Tabel 1, we see that i) the ARSV reflects that collaboration can improve the performance of original single-view based objects. The early fusion achieves the best performance, and there is an obvious improvement over no fusion, *i.e.*, ARSV@0.5 and ARSV@0.7 are increased by 12.74% and 15.72% respectively. Note that the DiscoNet inherits the performance of early fusion through knowledge distillation; ii) the ARCV reflects the quality of the collaboration process. The proposed UMC achieves the best performance. Compared to the DiscoNet, UMC improves by 14.39% in ARCV@0.5 and 29.07% in ARCV@0.7. Note that in when2com and who2com, the collaboration is simply integrated by stacking features along channels, resulting in the low quality of collaboration; iii) the ARCI reflects the detection performance under limited information. Thus, a high ARCI may cause more false positives, reducing the precision of the model; iv) the ARTC reflects the time continuity of the model, and the ARTC of existing methods is relatively low, In that sense, our work may unlock the future temporal reasoning in collaborative perception.

Performance-bandwidth trade-off analysis As demonstrated in Figure 6, we comprehensively compare the proposed UMC under different (δ_s, δ_c) values with the baseline methods in terms of the trade-off between performance and communication bandwidth. The no fusion model is shown as a dashed line since the communication volume is zero. We see that i) UMC achieves the best trade-off among the other methods; ii) the detect performance of where2comm is more stable than UMC when bandwidth is reduced. The where2comm is specific for the detection downstream task, which is based on the sparsity of foreground information to reduce communication by filtering the background intensively with a small performance sacrifice. And, our entropy-CS aims to optimize not only detection but also general downstream tasks based on the traditional information theory. Hence, when setting small value of (δ_s, δ_c), the detection performance will degrade more easily than where2comm, shown in Figure 6 (c)-(d).

Table 3. Detection performance with different grains selection. $\boldsymbol{F}^{e,t}_{i,j}(j = 1, 2, 3)$ denotes different grained collaborative features. Note that the C. V. is the short for Communication Volume.

Multi-Grains Selection $\boldsymbol{F}^{e,t}_{i,1}$	$\boldsymbol{F}^{e,t}_{i,2}$	$\boldsymbol{F}^{e,t}_{i,3}$	$\text{AP}_{50/70} \uparrow$	C. V. $\downarrow$
✓	✓		**67.80/60.01**	**19.23**
✓		✓	58.36/50.71	19.84
	✓	✓	60.27/53.63	20.02

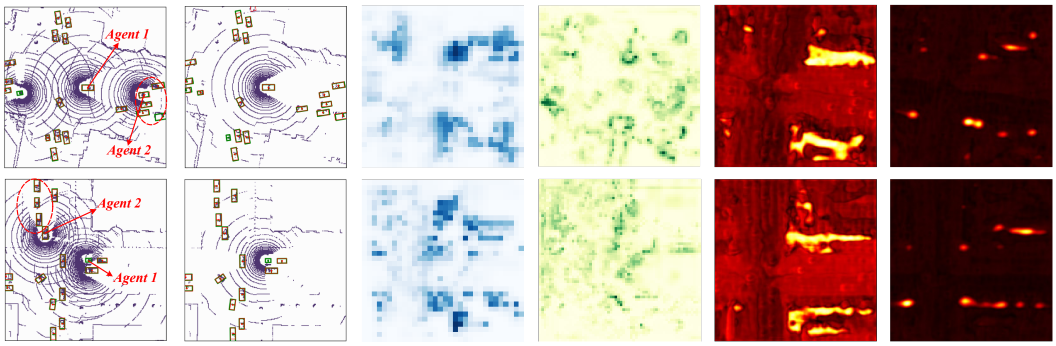
4.4. Qualitative evaluation

Visualization of UMC working mechanism To understand the working mechanism of the proposed UMC, we visualize the detection results, different grains collaborative feature maps, and the corresponding entropy-based transmission maps; see Figure 5. Note that the proposed entropy-based selection is matrix-based, reflecting the informative regions in a cell-level resolution, which is shown as the 2D communication map that is compressed along channels by a sum operation. Figure 5 (a), (b), and (c) represent three detection results of Agent 1 based on the no fusion, early fusion, and the proposed UMC, respectively. We see that with collaboration, UMC is able to detect objects in these sheltered and long-range zones. To further explain the rationale behind the results, Figure 5 (d) and (f) provide the corresponding ego coarse- and fine-grained collaborative feature maps, respectively. We clearly observe that the coarse-grained feature reflects the global structure information and the fine-grained feature contains rich local texture details, the proposed UMC complements their advantages. Figure 5 (e) and (g) show the coarse- and fine-grained communication maps from agent 2 to agent 1, respectively. We observe that with the grains being finer, the communication map becomes more sparse, meaning the entropy-based selection can more precisely find the informative regions relative to ego agents, also, the entropy-based selection ensures lower bandwidth under the multi-grain collaborative process.

Comparison with DiscoNet, When2com Figure 7 shows two examples to compare the UMC with When2com and DiscoNet. We see that UMC is able to detect more objects, especially in edge regions with sparse measurement. The reason is that both When2com and DiscoNet employ only a single coarse-grained global feature map for collaborations, which loses many details of the scene, while UMC combines coarse- and fine-grained features for collaboration, see the visualization in Figure 7 (d)-(e). Remarkably, compared to knowledge-distillation-based DiscoNet, the UMC does not need to pretrain different teacher models (early fusion) when switching to different datasets, and the performance will be influenced greatly when the teacher model is not performing well, as shown in Table 2.

Table 4. Quantitative results of ablation experiments on V2X-Sim.

Variants	Entropy-CS	G-CGRU	MGFE	$\text{ARSV}_{50/70}\uparrow$	$\text{ARCV}_{50/70}\uparrow$	$\text{ARCI}_{50/70}\downarrow$	$\text{ARTC}_{50/70}\uparrow$	$\text{AP}_{50/70}\uparrow$
(1)	✓	✓	✓	86.67/80.68	67.01/60.04	2.38/1.70	12.35/9.46	**67.80/60.01**
(2)		✓	✓	87.67/83.29	67.88/62.17	2.87/1.66	12.01/9.55	65.72/58.78
(3)	✓	✓		79.38/74.90	40.35/34.14	1.49/1.28	3.79/3.52	56.73/49.22
(4)		✓		87.72/82.13	60.58/46.53	3.13/1.43	12.78/9.98	67.30/56.19



(a) Early Fusion (b) UMC (c) $E_{1,1}^{e,t}$ (d) $E_{1,2}^{e,t}$ (e) $D_1^{e,t}$ (f) D_1

Figure 8. Detection and feature map visualization for Agent 1. Green/red boxes represent GT/predictions. (a) and (b) denote the results of early fusion and UMC. (c) and (d) denote the coarse- and fine-grained collaborative maps generated from the G-CGRU module. respectively. (e) and (f) denote the UMC and early fusion's downstream features $\boldsymbol{D}_1$ for final detection output $\tilde{\boldsymbol{Y}}_1$, respectively.

4.5. Ablation study

We validate several key designs of our framework, including the entropy-CS, the G-CGRU, and the MGFE module. The quantitative results are shown in Table 4.

Effect of grains selection In tabel 3 we investigate the detection performance when employing different grain selections. Note that the shape of $\boldsymbol{F}_{i,j}^{e,t}(j=1,2,3)\in\mathbb{R}^{W,H,C}$ is $32\times32\times256$, $64\times64\times128$ and $128\times128\times64$, respectively. We see that i) applying collaborative feature maps on the selection of $(\boldsymbol{F}_{i,1}^{e,t},\boldsymbol{F}_{i,2}^{e,t})$ has the best trade-off between performance and bandwidth; and ii) the selection of $(\boldsymbol{F}_{i,1}^{e,t},\boldsymbol{F}_{i,3}^{e,t})$ for collaboration has the worst detection performance. This is because the grains of feature difference between the two is too large, resulting in deviation in feature fusion process; and iii) the performance of selection of $(\boldsymbol{F}_{i,2}^{e,t},\boldsymbol{F}_{i,3}^{e,t})$ is slightly worse than $(\boldsymbol{F}_{i,1}^{e,t},\boldsymbol{F}_{i,2}^{e,t})$, because both $(\boldsymbol{F}_{i,2}^{e,t},\boldsymbol{F}_{i,3}^{e,t})$ focus on the local texture details, resulting in the loss of global structure information; and iv) on the premise of performance of $(\boldsymbol{F}_{i,1}^{e,t},\boldsymbol{F}_{i,2}^{e,t})$, the heavy bandwidth of the $(\boldsymbol{F}_{i,1}^{e,t},\boldsymbol{F}_{i,2}^{e,t},\boldsymbol{F}_{i,3}^{e,t})$ selection makes the potential performance improvement meaningless.

Effect of collaboration strategy Table 4 compares the different ablated networks. We see that i) the UMC (variants 1) achieves the best detection performance among the other ablated networks; and ii) variant 4 reflects that the redesigned graph-based collaborative GRU module has an obvious improvement over no fusion, *i,e*, AP@0.5 and AP@0.7 are increased by 26.66% and 24.72% respectively, see the Figure 8 (c) and (d); and iii) variant 2 is composed of variants 4 and multi-grain feature enhance module. Compared to variant 4, variant 2 improves by 12.05% in ARCV@0.5 and 33.61% in ARCV@0.7, showing the effectiveness of the MGFE module. Meanwhile, as demonstrated in Figure 8 (e) and (f), the MGFE module can enhance the downstream feature map compared to the early fusion method; and iv) from the results of variants 3 and 4, and the results of variants 1 and 2, we observe that the MGFE module can give positive guidance to the trainable entropy-based communication selection module because the MGFE module allows the entropy-CS to comprehensively select regions from local and global viewpoints.

5. Conclusion

We propose a unified, bandwidth-efficient collaborative perception framework named UMC. Its core concept is to utilize the multi-resolution technique to cover the intrinsic interactions among the communication, collaboration, and reconstruction processes in MCP. To validate it, we conduct experiments on the V2X-Sim and OPV2V datasets and propose a brand new evaluation metric. Comprehensive quantitative and qualitative experiments show that the UMC achieves an outstanding performance-bandwidth trade-off among the existing collaborative perception methods.

Acknowledgments: This work was supported in part by Shanghai Municipal Science and Technology Major Project under Grant 2018SHZDZX01, ZJ Lab, the Shanghai Center for Brain Science and Brain-Inspired Technology, in part by Shanghai Rising Star Program under Grant 21QC1400900, in part by Tongji-Qomolo Autonomous Driving Commercial Vehicle Joint Lab Project, and in part by Xiaomi Young Talents Program.

References

[1] Qi Chen. F-cooper: feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds. *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, 2019.

[2] Qi Chen, Sihai Tang, Qing Yang, and Song Fu. Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds. *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 514–524, 2019.

[3] Shanzhi Chen, Jinling Hu, Yan Shi, Li Zhao, and Wen Li. A vision of c-v2x: Technologies, field testing, and challenges with chinese development. *IEEE Internet of Things Journal*, 7(5):3872–3881, 2020.

[4] Wei-Ting Chen, H. Fang, Cheng-Lin Hsieh, Cheng-Che Tsai, I-Hsiang Chen, Jianwei Ding, and Sy-Yen Kuo. All snow removed: Single image desnowing algorithm using hierarchical dual-tree complex wavelet representation and contradict channel loss. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4176–4185, 2021.

[5] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *SSST@EMNLP*, 2014.

[6] Gavin E Crooks. On measures of entropy and information. *Tech. Note*, 9:v4, 2017.

[7] Jiaxun Cui, Hang Qiu, Dian Chen, Peter Stone, and Yuke Zhu. Coopernaut: End-to-end driving with cooperative perception for networked vehicles. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[8] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[9] Lingxiao He, Jian Liang, Haiqing Li, and Zhenan Sun. Deep spatial feature reconstruction for partial person reidentification: Alignment-free approach. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7073–7082, 2018.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] Xin Hong, Pengfei Xiong, Renhe Ji, and Haoqiang Fan. Deep fusion network for image completion. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 2033–2042, New York, NY, USA, 2019. Association for Computing Machinery.

[12] Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. In *Thirty-sixth Conference on Neural Information Processing Systems (Neurips)*, November 2022.

[13] Hongwu Kuang, Bei Wang, Jianping An, Ming Zhang, and Zehan Zhang. Voxel-fpn: Multi-scale voxel feature aggregation for 3d object detection from lidar point clouds. *Sensors (Basel, Switzerland)*, 20, 2020.

[14] 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.

[15] Zixing Lei, Shunli Ren, Yue Hu, Wenjun Zhang, and Siheng Chen. Latency-aware collaborative perception. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[16] Ruoteng Li, Robby T. Tan, and Loong Fah Cheong. All in one bad weather removal using architectural search. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3172–3182, 2020.

[17] Yiming Li, Dekun Ma, Ziyan An, Zixun Wang, Yiqi Zhong, Siheng Chen, and Chen Feng. V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving. *IEEE Robotics and Automation Letters*, 7(4):10914–10921, 2022.

[18] Yiming Li, Shunli Ren, Pengxiang Wu, Siheng Chen, Chen Feng, and Wenjun Zhang. Learning distilled collaboration graph for multi-agent perception. In *NeurIPS*, 2021.

[19] Yiming Li, Juexiao Zhang, Dekun Ma, Yue Wang, and Chen Feng. Multi-robot scene completion: Towards task-agnostic collaborative perception. In *6th Annual Conference on Robot Learning*, 2022.

[20] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017.

[21] Wei Liu, Andrew Rabinovich, and Alexander C. Berg. Parsenet: Looking wider to see better. *ArXiv*, abs/1506.04579, 2015.

[22] Yen-Cheng Liu, Junjiao Tian, Nathan Glaser, and Zsolt Kira. When2com: Multi-agent perception via communication graph grouping. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4105–4114, 2020.

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

[24] Zhuang Liu, Zhiqiu Xu, Hung-Ju Wang, Trevor Darrell, and Evan Shelhamer. Anytime dense prediction with confidence adaptivity. *International Conference on Learning Representations (ICLR)*, 2022.

[25] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[27] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchi-

cal vq-vae. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10770–10779, 2021.

[28] C. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017.

[29] C. Qi, L. Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017.

[30] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[31] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015.

[34] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[35] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10526–10535, 2020.

[36] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–779, 2019.

[37] Martin Simon, Stefan Milz, Karl Amende, and Horst-Michael Groß. Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. In *ECCV Workshops*, 2018.

[38] Angtian Wang, Yihong Sun, Adam Kortylewski, and Alan Loddon Yuille. Robust object detection under occlusion with context-aware compositionalnets. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12642–12651, 2020.

[39] Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Binh Yang, Wenyuan Zeng, James Tu, and Raquel Urtasun. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *ECCV*, 2020.

[40] Wentao Wang, Jianfu Zhang, Li Niu, Haoyu Ling, Xue Yang, and Liqing Zhang. Parallel multi-resolution fusion network for image inpainting. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14539–14548, 2021.

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

[42] Zhenda Xie, Zheng Zhang, Xizhou Zhu, Gao Huang, and Steve Lin. Spatially adaptive inference with stochastic feature sampling and interpolation. In *European Conference on Computer Vision (ECCV)*, August 2020.

[43] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers. In *Conference on Robot Learning (CoRL)*, 2022.

[44] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. *ArXiv*, abs/2203.10638, 2022.

[45] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *ICRA*, 2022.

[46] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors (Basel, Switzerland)*, 18, 2018.

[47] Binh Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[48] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, and Zaiqing Nie. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[49] Xiaoding Yuan, Adam Kortylewski, Yihong Sun, and Alan Loddon Yuille. Robust instance segmentation through reasoning about multi-object occlusion. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11136–11145, 2021.

[50] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14816–14826, 2021.

[51] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high-quality image inpainting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1486–1494, 2019.

[52] Kaihao Zhang, Rongqing Li, Yanjiang Yu, Wenhan Luo, and Changsheng Li. Deep dense multi-scale network for snow removal using semantic and depth priors. *IEEE Transactions on Image Processing*, 30:7419–7431, 2021.

[53] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.