SlimComm: Doppler-Guided Sparse Queries for Bandwidth-Efficient Cooperative 3-D Perception

Anonymous ICCV submission

Paper ID *****

Abstract

Accurate cooperative 3-D perception under tight Vehicle-to-Vehicle(V2V) bandwidth budgets remains a major challenge for Connected Autonomous Vehicles (CAVs). We present SlimComm, a bandwidth-aware framework that fuses LiDAR with 4-D radar while exchanging only a handful of semantically important Bird's-Eye-View (BEV) features. SlimComm first places sparse query locations guided by two priors: (i) a motion-centric Dynamic Map derived from radar Doppler and (ii) a Confidence Map highlighting likely foreground cells and occlusion shadows. Features gathered at these queries from neighbouring CAVs are then fused by a gated multi-scale deformable-attention block. Because no public multi-agent radar benchmark with per-point Doppler exists, we release OPV2V-R and Adver-City-R CARLA-based extensions of two popular V2X suites, together with our radar-simulation toolbox and full code. On these datasets, SlimComm achieves a balance between accuracy and efficiency, matching or exceeding the performance of prior baselines while transmitting only ∼10 % of data. We will re-evaluate SlimComm on real-sensor data as soon as such a dataset becomes publicly available.

1. Introduction

Autonomous driving and other unmanned systems have made rapid strides, spurred by the demand for reliable 3-D object detection [11]. Yet dependable perception in complex outdoor environments remains difficult because of occlusions, adverse weather, and sensor-specific limitations [13]. LiDAR supplies centimetre-level geometry but degrades at long range and in fog or rain [2]; conversely, *4-D* radar offers resilient range–Doppler measurements with coarse angular resolution [2, 21].

Collaborative perception through Vehicle-to-Everything (V2X) communication can overcome these weaknesses by allowing vehicles to see beyond their own field of view. However, broadcasting dense Bird's-Eye-View (BEV) feature maps quickly overwhelms capacity of typical DSRC/C-V2X links. The key question is therefore which information to share and from whom to request it.

We answer this question with **SlimComm**, a proactive cooperative-perception framework that transmits only a sparse set of high-value queries instead of full feature maps. The query strategy follows two steps:

(i) *Dynamic Map.* Ego-motion-compensated radar Doppler forms a motion-centric map; heuristic queries placed on this map focus on moving objects.

(ii) *Exploratory queries.* A confidence prior highlights likely foreground cells; additional queries dropped in the resulting occlusion shadows prompt collaborators to recover partially or fully hidden objects.

Each ego vehicle broadcasts its queries. Chosen neighbours warp their local BEV features into the ego frame, extract a halo-enriched context window around every query (see Sec. 4.4), and return those features. The ego agent then fuses ego and neighbour responses using a gated multi-scale deformable-attention block.

Contributions.

- A Doppler-compensated **Dynamic Map** that converts raw 4-D radar velocities into a motion prior for query placement;
- A **semantic-prior query strategy** targeting dynamic objects and occlusion-prone regions, achieving a superior accuracy–bandwidth trade-off;
- A **communication-efficient collaborator-selection scheme** that enriches returned features with halo context while sending only ∼10 % of the bytes required by full-map sharing;
- **OPV2V-R** and **Adver-City-R**: CARLA-based extensions of two popular V2X benchmarks augmented with 4-D radar, released together with our radar-simulation toolbox and full training code to enable reproducible research until public multi-agent Doppler datasets become available.

2. Related Work

2.1. Communication-Efficient Cooperative Perception

Early frameworks such as V2VNet [19], Attentive Fusion [25] and AdaFusion [14] exchange *dense* BEV feature maps among CAVs. Although this maximises accuracy, it quickly overloads the 10–20 Mbit s^{-1} V2X channel; compressing the maps alleviates traffic but sacrifices precision.

Later work therefore embraces sparse communication. Where2Comm [6] transmits only high-confidence BEV cells, When2Com [10] triggers exchange when ego-view uncertainty spikes, and CoSDH [22] prunes messages via a supply–demand model. SCOPE [26], StreamLTS [31] and DelAwareCol [1] add spatio-temporal filters to squeeze bandwidth further.

However, these methods still rely on purely statistical confidence or entropy cues; they rarely encode semantic knowledge of motion or occlusion. Consequently, much of the transmitted data corresponds to static background, not objects of interest.

2.2. Sensing Modalities

Most cooperative systems rely on LiDAR alone or LiDAR–camera fusion [1, 3, 6, 10, 14, 19, 22, 25, 26, 31]. By contrast, LiDAR–radar fusion has recently boosted single-vehicle perception: LiRaFusion introduces a learnable gating scheme [16]; InterFusion adds pillar-wise attention [18]; RLNet compensates radar noise [23]; and BiLRFusion couples the modalities bidirectionally [20]. None of these approaches addresses inter-vehicle occlusion or bandwidth constraints in multi-agent settings. Exploiting radar Doppler within a cooperative framework therefore remains largely open.

2.3. Collaborative-Perception Datasets

A comprehensive review of V2X datasets is provided in [29]. OPV2V [25] and AdverCity [8] supply simulated LiDAR–camera data in CARLA, whereas DAIR-V2X [30] offers real-world LiDAR–camera recordings. V2X-R [7] introduces cooperative radar but omits per-point Doppler and mounts only a front-facing sensor, limiting motion analysis and 360° coverage.

Summary. Current methods either broadcast full feature maps for high accuracy or reduce bandwidth with loosely guided sparsity. They overlook **occlusion recovery**, vital for both perception completeness and efficient collaboration, and under-utilise radar in multi-agent contexts, partly due to the lack of suitable public datasets. By contrast, **SlimComm** combines Doppler-aware, occlusion-guided queries with LiDAR–radar cooperation to improve the accuracy–bandwidth trade-off.

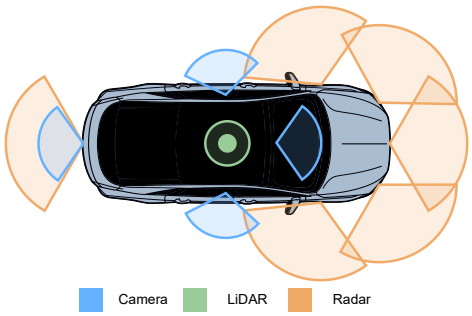

Figure 1. Sensor-suite configuration for each CAV. The six-radar setup is designed to provide full 360° Doppler velocity coverage, with three radars covering the front, one the rear, and two monitoring adjacent lanes.

3. Dataset

To address the gap in publicly available V2X benchmarks and to properly evaluate multi-agent models that fuse 4-D radar, we introduce radar-augmented versions of the OPV2V [25] and Adver-City [8] datasets. Our goal is to benchmark models that leverage point clouds enriched with Doppler velocity under a variety of driving conditions.

3.1. Dataset Creation

OPV2V covers generic urban traffic with diverse intersection types, occlusion patterns and flow densities [25]. Adver-City instead concentrates on crash-relevant situations derived from real accident statistics, including complex junctions and rural roads with restricted sight lines [8]. By equipping both suites with an identical sensor package we enable consistent evaluation of radar-enhanced perception across complementary scenarios.

OPV2V-R and **Adver-City-R** are generated with the same CARLA [4] and OpenCDA [24] pipeline, recorded at 10Hz, and annotated with fully compatible 3-D bounding boxes. For Adver-City-R we select the *ClearDay* weather profile, remove roadside units, and harmonise vehicle classes with OPV2V-R (e.g. no micro-cars). Pedestrian labels are retained but can be ignored during training to match OPV2V-R. Fig. 2 shows that both datasets have a median of ≈ 2 neighbouring CAVs per frame. OPV2V-R exhibits a broader tail (up to five neighbours), whereas Adver-City-R is more narrowly distributed.

Each CAV carries RGB cameras, a LiDAR, GNSS/IMU,

Sensor	Specification
4× Camera	RGB, 800 × 600, 110° FOV
1× LiDAR	64 ch., 1.3 M pts/s, 120 m, –25°–2° vert. FOV, 0.02 m noise, 20 Hz
6× Radar	0.06 M pts/s, 150 m, 120° horiz. FOV, 30° vert. FOV
GPS/IMU	GNSS alt. noise 0.001 m; IMU heading 0.1°, speed 0.2 m/s

Table 1. Sensor-suite configuration.

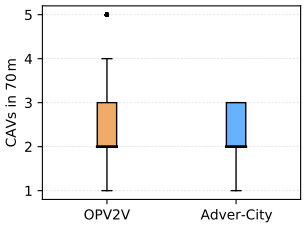

Figure 2. Number of neighbouring CAVs ($\leq$70 m) per frame.

and six simulated radars that output XYZ coordinates and Doppler velocity (see Tab. 1 for specifications and Fig. 1 for mounting positions). Cameras provide 360° coverage; the LiDAR is roof-mounted; three radars cover the front bumper, one the rear, and two under the side mirrors look backward to monitor adjacent lanes.

To our knowledge, no *publicly available* V2V dataset yet provides synchronised radar XYZ + Doppler for multi-agent perception.[1]

4. Method

4.1. Overall Architecture

Fig. 5 outlines the end-to-end cooperative-perception pipeline. Each agent first feeds voxelised LiDAR and radar data into a shared **Encoder**, producing multi-scale feature tensors $\{\boldsymbol{F}_{i,l} \in \mathbb{R}^{C_l \times H_l \times W_l}\}$.

Alongside these features, the encoder outputs three semantic priors (see Sec. 4.2): a Dynamic Map $\boldsymbol{D}_i$, a Confidence Map $\boldsymbol{C}_i$, and a Foreground Density Map $\boldsymbol{V}_i$. These maps guide the **Ego Query Generator**(see Sec. 4.3), which converts the down-sampled $\boldsymbol{D}_{i,l}$ and

Each neighbour consults its own density map $\boldsymbol{V}_j$ to decide whether to participate, thereby filtering out uninformative links and reducing bandwidth. Agents that opt in return *halo-enriched features* $\{\boldsymbol{H}_{j\to i,l}\}$ (see Sec. 4.4). These responses are merged by the **Gated Multi-Scale Deformable Fusion** module into per-scale fused features $\{\tilde{\boldsymbol{F}}_{i,l}\}$, which are then aggregated across scales to form a unified BEV tensor $\tilde{\boldsymbol{S}}_i$. A lightweight detection head operating on $\tilde{\boldsymbol{S}}_i$ produces the final 3-D predictions. Overall, the framework selects only informative collaborators and integrates complementary observations into a compact BEV representation, achieving accurate yet bandwidth-efficient cooperative perception.

4.2. Encoder

Backbone. As shown in Fig. 3, following PointPillars [9], LiDAR and radar point clouds are pillarised into BEV tensors $\boldsymbol{L}_i$ and $\boldsymbol{R}_i$, concatenated, and fed to a ResNet BEV backbone [5]. The backbone yields multi-scale features $\{\boldsymbol{F}_{i,l}\}$ and a high-resolution map $\boldsymbol{S}_i$. A classification head along with sigmoid and channel-wise max operating on $\boldsymbol{S}_i$ produces the confidence map $\boldsymbol{C}_i \in [0,1]^{H\times W}$. The next two subsections detail the auxiliary priors used later for query generation and collaborator selection. $\boldsymbol{C}_{i,l}$ into sparse query locations $\{\boldsymbol{Q}_{i\to j,l}\}$ for deformable attention. Before feature exchange, the ego agent transmits its query locations to neighbouring agents through the **Communicator**.

[1] A real-world V2X-Radar dataset has been reported in [27], but the data were not released at the time of writing.

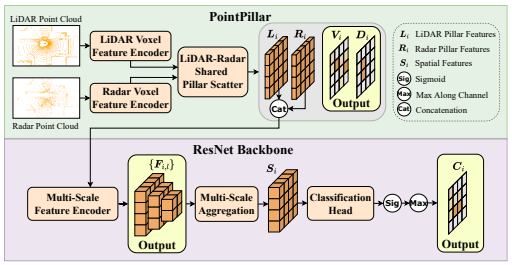

Figure 3. Encoder overview. LiDAR–radar pillars are concatenated in BEV space and processed by a ResNet backbone, which outputs a feature pyramid and three semantic priors.

4.2.1. Dynamic Map

Radar Doppler information is converted into a binary motion mask after ego-motion compensation. Let $\mathbf{v}_i^{\text{veh}}$ be the ego velocity of vehicle i in its own frame. For the k-th radar on that vehicle, the velocity in radar coordinates is

$$\mathbf{v}_{i,k}^{\text{veh}} = \mathbf{R}_{i,k}\, \mathbf{v}_i^{\text{veh}}, \tag{1}$$

with $\mathbf{R}_{i,k}$ the extrinsic rotation matrix.

The measured Doppler velocity for point n is

$$v_{i,k,n}^{\text{Doppler}} = \left(\mathbf{v}_{i,k,n}^{\text{abs}} - \mathbf{v}_{i,k}^{\text{veh}}\right) \cdot \mathbf{u}_{i,k,n}, \tag{2}$$

where $\mathbf{u}_{i,k,n} = \mathbf{p}_{i,k,n}/\|\mathbf{p}_{i,k,n}\|$ is the line-of-sight unit vector. Re-arranging gives the compensated radial velocity

$$v_{i,k,n}^{\text{radial}} = v_{i,k,n}^{\text{Doppler}} + \mathbf{v}_{i,k}^{\text{veh}} \cdot \mathbf{u}_{i,k,n}. \tag{3}$$

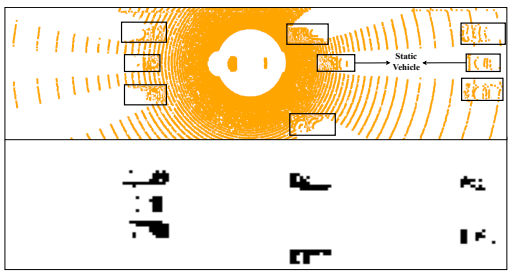

Figure 4. Top: Raw point cloud with GT bounding boxes. Bottom: Dynamic Map. All dynamic vehicles are captured as dynamic grids (black), while the two static vehicles are correctly excluded from dynamic regions.

A BEV cell is marked dynamic if any radar point inside it satisfies $|v^{\text{radial}}_{i,k,n}| > v_{\text{th}}$, with $v_{\text{th}} = 1.0$ m/s. This yields the binary Dynamic Map $\boldsymbol{D}_i \in \{0,1\}^{H\times W}$. Fig. 4 demonstrates that the dynamic map reliably distinguishes moving and static objects, achieving precise dynamic–static separation.

4.2.2. Foreground Density Map

The Foreground Density Map highlights informative regions by combining height-based foreground masking with point density.

Foreground masking. Using thresholds $T_{\text{lower}} = -1.2$ m, $T_{\text{upper}} = 0$ m, and $T_{\text{max}} = 1.0$ m, a cell is background if it contains (i) any point above T_{max} (tall static structures), or (ii) all points outside $[T_{\text{lower}}, T_{\text{upper}}]$ (ground or noise). Foreground masks from LiDAR and radar are combined via

$$\boldsymbol{FG}_i = \boldsymbol{FG}_i^{\text{L}} \vee \boldsymbol{FG}_i^{\text{R}}. \tag{4}$$

Density scaling. Point counts $\boldsymbol{N}_i^p$ are normalised by the pillar capacity N_{max}:

$$\boldsymbol{DS}_i = \boldsymbol{N}_i^p / N_{\text{max}}. \tag{5}$$

Final map. The Foreground Density Map is the element-wise product

$$\boldsymbol{V}_i = \boldsymbol{FG}_i \odot \boldsymbol{DS}_i. \tag{6}$$

It suppresses empty or background cells while retaining dense foreground evidence, and is later used for collaborator selection.

4.3. Ego Query Generator

To minimise communication overhead, the query generator produces a sparse set of reference points focused on the most informative regions of the scene. As shown in Fig. 6, it runs per scale in two stages, yielding Heuristic Reference Points (HRP) for refining visible objects and Exploratory Reference Points (ERP) for probing occluded areas.

Heuristic Branch. This branch focuses on regions that already exhibit strong object evidence. For each scale l, the dynamic and confidence maps are down-sampled, $\boldsymbol{D}_i \to \boldsymbol{D}_{i,l}$ and $\boldsymbol{C}_i \to \boldsymbol{C}_{i,l}$, and HRP locations are drawn from two candidate pools:

1. every grid cell flagged as dynamic in $\boldsymbol{D}_{i,l}$;
2. the highest-scoring cells in $\boldsymbol{C}_{i,l}$ that are not in Pool 1, selected until the per-scale budget N_l^r is met.

The resulting set, $\boldsymbol{R}^h_{i,l} \in \mathbb{R}^{N_l^r \times 2}$, stores BEV coordinates (u, v). Embeddings $\boldsymbol{E}^h_{i,l} \in \mathbb{R}^{N_l^r \times C_l}$ are obtained by bilinearly sampling the ego BEV feature map, yielding a rich, context-aware starting point for object refinement.

Exploratory Branch. Occlusions often hide critical objects; this branch explicitly seeks them.

1. **Occluder identification.** Significant peaks in the confidence map are extracted as occluder centroids,
$$\boldsymbol{C}^o_{i,l} = \text{MaxPool}_{3\times 3}(\boldsymbol{C}_{i,l}) \ \wedge \ \boldsymbol{C}_{i,l}, \tag{7}$$
where $\wedge$ denotes element-wise logical AND. Border pixels are first masked out to suppress artificial edges. A per-scene percentile threshold is then applied.
2. **Shadow sampling.** For each centroid, a shadow ERP is placed at a stochastic, biased offset, forming $\boldsymbol{R}^e_{i,l} \in \mathbb{R}^{N_l^r \times 2}$ (see Fig. 6).
3. **Contextual embedding.** Each ERP embedding concatenates three signals: (i) the occluder's BEV feature, (ii) the 2-D shadow offset, and (iii) a scale-specific learnable exploration token $\boldsymbol{t}_l$. The token $\boldsymbol{t}_l \in \mathbb{R}^{C_l}$ is a single parameter vector shared by all ERPs at level l, initialised with Xavier normal noise and updated end-to-end during training. The concatenated vector is passed to a two-layer MLP, yielding $\boldsymbol{E}^e_{i,l} \in \mathbb{R}^{N_l^r \times C_l}$.

Query aggregation. For each scale l, we concatenate the HRP and ERP into an anchor set

$$A_{i,l} = R^{\text{H}}_{i,l} \cup R^{\text{E}}_{i,l}, \qquad |A_{i,l}| = 2N_l^r. \tag{8}$$

Two-stage offset strategy. A lightweight MLP predicts one coarse 2-D offset for every anchor, forming the nudged centres $\tilde{A}_{i,l} = A_{i,l} + O_{i,l}$.

Query-offset regularisation. To prevent the HRP and ERP branches from collapsing into identical behaviour, we introduce an auxiliary loss that encourages exploratory offsets to be larger than heuristic ones by a scale-dependent margin δ_l. Let $O^{\text{H}}_{i,l}$ and $O^{\text{E}}_{i,l}$ denote the two offset subsets; the loss reads

$$\mathcal{L}_{\text{offset}} = \sum_l \Big[\delta_l - \big(\mathbb{E}\|O^{\text{E}}_{i,l}\|_2 - \mathbb{E}\|O^{\text{H}}_{i,l}\|_2\big)\Big]_+, \tag{9}$$

with $[\cdot]_+$ the ReLU and δ_l the average occluder-to-shadow distance at that scale. This term is added to the main detection objective with a dynamic weight proportional to the primary PointPillar detection loss value.

Fine sampling. Each nudged centre $\tilde{a} \in \tilde{A}_{i,l}$ is passed to Deformable Attention, which predicts a learnable 3×3 halo ($n_{\text{points}} = 9$) of fine offsets, yielding the final sampling locations used to gather collaborators features. The final set of sampling locations used to gather collaborator features is then given by:

$$\boldsymbol{Q}i, l = \tilde{a} + \Delta^{\text{fine}} h, p \mid \tilde{a} \in \tilde{A}i, l, \forall h, p. \tag{10}$$

The two-stage design lets the network first learn a global (coarse) correction and then a local, detail-oriented sampling pattern (fine) without incurring additional bandwidth.

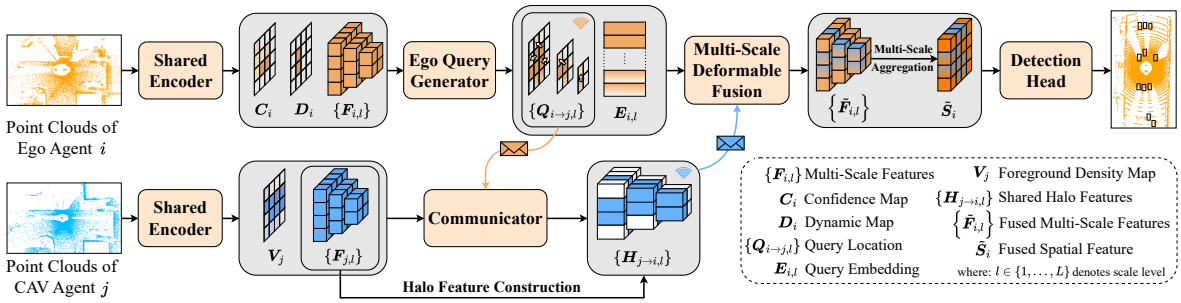

Figure 5. End-to-end cooperative perception. Each agent voxelises its LiDAR–radar points and runs a **shared encoder** that outputs multi-scale features and three semantic priors (dynamic, confidence, density). The **Ego Query Generator** uses those priors to emit sparse queries, neighbours respond with **halo-enriched** features, and a **gated deformable fusion** module merges everything into a BEV tensor for detection.

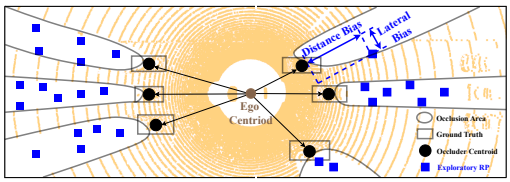

Figure 6. Exploratory reference points are placed behind occluder centroids at a learned distance and lateral bias.

4.4. Communicator

To enable efficient and adaptive multi-agent perception, the module first selects collaborators and then transmits only sparse, halo-enriched features that are spatially aligned with the ego's query locations.

Collaborator Selection. Prior to feature exchange, the ego broadcasts its query locations $\boldsymbol{Q}_{i,l}$ and global pose to neighbouring agents. Because these queries are defined in the ego coordinate frame, each candidate agent j warps its Foreground Density Map $\boldsymbol{V}_j$ into that frame, yielding $\boldsymbol{V}_{j\to i}$. An agent becomes a collaborator whenever at least one query lands on a BEV cell whose warped foreground density exceeds the communication threshold:

$$\max_{(u,v)\in \boldsymbol{Q}_{i,l=0}} \boldsymbol{V}_{j\to i}(u,v) > \tau_{com}, \tag{11}$$

where (u, v) are BEV grid indices. The test is performed only at the finest scale l=0, whose higher resolution captures the most detailed occupancy information.

Halo-enriched Sparse Feature Encoding. Most existing methods [6, 14, 25, 28] perform early-stage projection: they first transform every CAV's point cloud into the ego frame and then learn all subsequent features there. In real-time V2X, however, a vehicle may connect to several neighbours; repeatedly transforming and processing identical point clouds in multiple coordinate frames quickly becomes computationally prohibitive. Some methods [12, 15] sidesteps this by extracting features in each agent's own frame and warping them into collaborators' frames.

Following the feature warping strategy adopted in prior works, we introduce halo enrichment: Each collaborator first warps its multi-scale feature maps $\{\boldsymbol{F}_{j,l}\}$ into the ego frame, then augments every grid cell with the features of its 3×3 neighbourhood, concatenated along the channel dimension. This enriches the spatial context for subsequent deformable attention, reducing the impact of limited sampling density and providing robustness against the minor pixel-level misalignments that can occur during feature warping.

Finally, only the halo-enriched features at the ego agent's query locations are transmitted, $\{\boldsymbol{H}_{j\to i,l} \in \mathbb{R}^{N_l^q \times 9C_l}\}$, providing semantically rich yet bandwidth-efficient inputs for cross-agent fusion.

4.5. Gated Multi-Scale Deformable Fusion

After halo-enriched, sparse features arrive from the selected collaborators, fusion proceeds in three steps: Step 1: CAV aggregation, Step 2: deformable cross-attention, and Step 3: gated residual blending.

Step 1 — CAV aggregation. For each scale l, features from the N participating CAVs are averaged:

$$\boldsymbol{H}_{i,l}^{\text{CAV}} = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{H}_{j\to i,l} \tag{12}$$

We use simple averaging as a robust, parameter-free baseline to create a consensus representation, which prevents any single noisy collaborator from dominating the fused feature.

Step 2 — Deformable cross-attention. The ego's query embeddings $\boldsymbol{E}_{i,l}$, query locations $\boldsymbol{Q}_{i,l}$, and the aggregated tensor $\boldsymbol{H}_{i,l}^{\text{CAV}}$ are passed to a multi-head deformable-attention module [32]. Each query samples its reference point and learned offsets, producing fused query features that are scattered back to the BEV grid:

$$\boldsymbol{F}_{i,l}^{\text{CAV}} = \text{Scatt}\left(\text{DeAttn}\left(\boldsymbol{E}_{i,l}, \boldsymbol{Q}_{i,l}, \boldsymbol{H}_{i,l}^{\text{agg}}\right), \boldsymbol{Q}_{i,l}\right) \quad (13)$$

with untouched cells filled with zeros.

Step 3 — Gated residual blending. Because $\boldsymbol{F}_{i,l}^{\text{CAV}}$ and the ego features $\boldsymbol{F}_{i,l}$ come from different distributions, a spatial gate modulates their mixture:

$$\tilde{\boldsymbol{F}}_{i,l} = (1 - \boldsymbol{G}_{i,l}) \odot \boldsymbol{F}_{i,l} + \boldsymbol{G}_{i,l} \odot \boldsymbol{F}_{i,l}^{\text{CAV}} \quad (14)$$

where

$$\boldsymbol{G}_{i,l} = \sigma\left(\text{Conv}_{1\times1}\left[\boldsymbol{F}_{i,l} \,\|\, \boldsymbol{F}_{i,l}^{\text{CAV}}\right]\right) \in [0,1]^{C_l \times H_l \times W_l} \quad (15)$$

Here $\|$ denotes channel concatenation, σ is the sigmoid, and $\odot$ is element-wise multiplication. The 1×1 convolution learns to emphasise useful collaborative cues. The rational behind using a gating mechanism is that the aggregated CAV features $\boldsymbol{F}_{i,l}^{\text{CAV}}$ and the original ego features $\boldsymbol{F}_{i,l}$ originate from different sources and exhibit distinct distributions. Directly adding them may disrupt feature consistency. The gating network learns to adaptively balance these two streams, suppressing noise and highlighting useful fused content.

Multi-scale aggregation and detection. The collection $\{\tilde{\boldsymbol{F}}_{i,l}\}$ is forwarded to the same ResNet aggregation block used in the encoder (Sec. 4.2), producing a unified BEV tensor $\tilde{\boldsymbol{S}}_i$. A PointPillars detection head then processes $\tilde{\boldsymbol{S}}_i$ to generate the final 3-D predictions.

5. Experiment

5.1. Experimental Setup

Our experiments are conducted on the Adver-City-R and OPV2V-R datasets, as introduced in Sec. 3.1. The voxelization settings include a grid size of (0.4 m, 0.4 m, 4 m), a maximum of 32 points per voxel, and a communication range of 70 m. The detection area is defined as a cuboid with dimensions 281.6 m (length), 80 m (width), and 4 m (height), with each vehicle located at the center of its own detection region.

For our model, we adopt the Adam optimizer with a learning rate of 0.002, a weight decay of 1×10^{-4}, and an epsilon value of 1×10^{-10}. Gradient clipping is applied with a maximum norm of 1. A MultiStep learning rate scheduler is used with a decay factor $\gamma = 0.1$, and learning rate drops scheduled at epochs 10 and 15.

We use three scales for cross-agent fusion. The corresponding feature map shapes are $(128, 100, 352)$, $(256, 50, 176)$, and $(512, 25, 88)$, respectively. The number of reference points per scale is set to 200, 100, and 50 for each query generator branch. Each reference point is associated with 9 learnable offsets and 4 attention heads. Offsets that fall on the same location are only transmitted once. Occluder centroids are selected with percentile thresholds $p_l \in \{0.5, 0.5, 0.5\}$.

We adopt the PointPillars detection loss—focal loss for classification ($\alpha = 0.25$, $\gamma = 2.0$) and smooth-L_1 regression on the 7-D bounding box with $\lambda_{\text{box}} = 2.0$—and augment it with an offset-regularization term (see Sec. 4.3). The auxiliary loss is weighted at $0.1\times$ the current detection loss and uses an adaptive margin δ_l proportional to the average occluder-to-shadow distance at each scale. This encourages the exploratory branch to learn larger offsets than the heuristic branch, ensuring the two branches maintain distinct behaviours.

All baseline methods are trained using their original configurations, including batch sizes, optimizers, projection strategy, and learning rates. Since no existing method supports LiDAR-Radar fusion for cooperative perception, we implement the same encoder as our model (introduced in Sec. 4.2) for them to ensure a fair comparison. Specifically, a radar voxel feature encoder (VFE) branch is added, and the resulting radar pillar features are concatenated with the LiDAR features before being passed to their corresponding backbones. All models are trained on two RTX 3090 GPUs.

5.2. Quantitative Evaluation

To evaluate our proposed framework, we benchmark its performance against several leading collaborative perception methods. Tab. 2 shows the detection performance across two distinct data splits designed to test the model under varying levels of environmental complexity. The Dense scenarios in AdverCity feature a 2.67x increase in the number of vehicles compared to the Sparse scenarios, thereby significantly amplifying the degree of occlusion [8].

5.2.1. Bandwidth Measurement

During evaluation, each collaborator transmits only its non-zero feature values; zeros are skipped via sparse encoding. As stated in our experimental setup, these evaluations are performed with **neither feature compression nor bandwidth limits applied** to ensure a fair and direct comparison of algorithmic efficiency.

Therefore, the payload for each transmitted feature is its full float32 size (4 bytes). The total payload for a given scene s is $B_s = 4\sum_l N_{s,l}$ [bytes], and the average bandwidth reported in our results is $\overline{MB}$ [MB/frame]. This metric represents the raw, uncompressed data payload required

by the algorithm before any quantization or channel-specific encoding would be applied in a real-world deployment.

Adver-City-R				
Method	AP@0.5↑ G. / S. / D.	AP@0.7↑ G. / S. / D.	CV↓	BD↓
AttFusion [25]	0.64 / 0.69 / 0.63	0.47 / 0.54 / 0.46	19.28	13.47
S-AdaFusion [14]	0.66 / 0.71 / **0.65**	**0.54** / 0.59 / **0.52**	19.52	16.14
SCOPE [26]	0.23 / 0.17 / 0.24	0.14 / 0.13 / 0.14	18.78	9.43
Where2Comm [6]	0.47 / 0.47 / 0.47	0.23 / 0.33 / 0.29	18.30	6.20
SlimComm (Ours)	**0.67 / 0.72 / 0.65**	**0.54 / 0.63 / 0.52**	**14.97**	**1.13**
OPV2V-R				
Method	AP@0.5↑	AP@0.7↑	CV↓	BD↓
AttFusion [25]	0.89	0.80	19.29	6.72
S-AdaFusion [14]	**0.91**	**0.85**	20.47	16.35
SCOPE [26]	0.73	0.66	18.59	4.75
Where2Comm [6]	0.86	0.77	18.74	4.45
SlimComm (Ours)	0.87	0.80	**16.07**	**0.63**

Table 2. Detection performance (AP@0.5 / AP@0.7), communication volume (CV, measured in $\log_2$ scale), and bandwidth usage (BD, in MB/frame) across methods at General scenarios. AdverCity results are split into General (G.), Sparse (S.), and Dense (D.) scenarios.

As shown in Tab. 2, SlimComm consistently achieves a superior balance between detection performance and communication efficiency across both benchmarks. On **Adver-City-R**, which is characterized by more complex scenarios with a higher average number of neighboring vehicles per scene, SlimComm delivers state-of-the-art precision. The increased vehicle density and resulting occlusions cause the query-generation mechanism to adaptively increase its bandwidth usage to **1.13 MB** to gather the necessary information. This is particularly effective in the **Dense scenarios**, where methods like Where2Comm and SCOPE, lacking semantic prior guidance, fail to capture critical occluded regions. In contrast, SlimComm's occlusion-aware ERP mechanism successfully identifies these areas, matching the accuracy of data-intensive methods while remaining highly efficient.

This efficiency is further highlighted on the **OPV2V-R** dataset, which has fewer vehicles on average in its test scenarios. Here, SlimComm's performance remains highly competitive with top-tier methods while requiring only **0.63 MB** of bandwidth—an approximate **7x reduction** compared to the next most efficient method (Where2Comm at 4.45 MB). This demonstrates that SlimComm's query-based strategy successfully adapts to scene complexity, preserving high-quality perception while drastically cutting communication overhead and proving its viability for real-world, bandwidth-constrained applications.

5.3. Ablation Studies

Impact of Different Components. We conduct ablation studies to evaluate the contribution of each component in our framework, as shown in Tab. 3. On Adver-City-R, introducing the exploratory branch improves the AP@0.5 from 0.58 to 0.61 and AP@0.7 from 0.41 to 0.49, demonstrating the benefit of explicitly exploring occluded or uncertain areas. Further incorporating the halo-enrichment mechanism brings an additional performance gain, reaching AP@0.5 of 0.67 and AP@0.7 of 0.54, which confirms that using spatial context is an effective strategy to mitigate feature warping distortion and sample sparsity. On OPV2V-R, a similar trend is observed.

Module			Adver-City-R		OPV2V-R	
HRP	ERP	HE	AP@0.5↑	AP@0.7↑	AP@0.5↑	AP@0.7↑
✓			0.58	0.41	0.83	0.73
✓	✓		0.61	0.49	0.83	0.76
✓	✓	✓	0.67	0.54	0.87	0.80

Table 3. Ablation study on different module combinations. HRP: Heuristic Branch in Query Generator, ERP: Exploratory Branch in Query Generator, HE: Halo-enrichment

Impact of Reference Points. An ablation study was conducted to investigate how the number of queries affects performance and communication cost, as shown in Tab. 4. These queries are initialized from a set of reference points derived from semantic priors. Using a minimal set of (50, 25, 15) reference points per scale leads to the lowest detection accuracy, indicating insufficient coverage of critical object and occlusion regions. Increasing the number to (100, 50, 25) and further to the default setting of (200, 100, 50) progressively improves both AP@0.5 and AP@0.7. This confirms the trade-off between accuracy and bandwidth: while more queries enhance perception quality, they also increase communication cost. Considering this balance, the (200, 100, 50) setting was adopted for our main experiments.

Adver-City-R				
RP	AP@0.5↑	AP@0.7↑	CV↓	BD↓
(50, 25, 15)	0.59	0.48	12.81	0.29
(100, 50, 25)	0.61	0.48	14.22	0.65
(200, 100, 50)	0.67	0.54	14.97	1.13

Table 4. Ablation study on the number of Reference Points (RP) per scale used in the query generator. A higher number of queries improves accuracy (AP) at the cost of increased communication volume (CV) and bandwidth (BD).

Impact of Communication Threshold. To analyze the effectiveness of the collaborator selection mechanism, an ablation study was performed on the communication threshold τ_{com}, as shown in Tab. 5. This threshold determines the minimum foreground density required for a neighboring agent to respond to a query.

As the threshold increases from 0 to 0.75 on the OPV2V-R dataset, there is a clear trade-off between accuracy and efficiency. A stricter threshold filters out more potential collaborators, leading to a steady decrease in communication volume (CV) and bandwidth (BD). However, this reduction in communication comes at the cost of lower detection accuracy, as potentially valuable information is discarded. The setting of $\tau_{com} = 0$ (used in our main experiments) was chosen to strike a balance, effectively pruning uninformative links without significantly compromising perception quality.

OPV2V-R				
τ_{com}	AP@0.5↑	AP@0.7↑	CV↓	BD↓
0	0.87	0.80	16.07	0.63
0.25	0.85	0.76	15.53	0.61
0.5	0.84	0.75	13.15	0.52
0.75	0.83	0.73	9.72	0.38

Table 5. Ablation study on the communication threshold (τ_{com}). A higher threshold reduces communication but can also decrease detection accuracy. Our default setting is $\tau_{com} = 0$.

6. Limitations and Future Work

Although **SlimComm** advances bandwidth-aware cooperative perception, several avenues remain open for exploration. A critical next step is real-world validation. All current experiments rely on CARLA simulation because no public multi-agent radar dataset with per-point Doppler is yet available. The SlimComm framework will be re-trained and evaluated as soon as such data are released (e.g., the forthcoming V2X-Radar [27]).

A key challenge in transitioning from simulation to the real world is handling errors from communication delay and imperfect localization. The current design does not account for these, nor for the feature warping misalignment they cause. Future work must mitigate these issues by introducing synchronization mechanisms, pose refinement modules, and feature alignment techniques such as spatial cross-attention or offset correction. The inherent robustness of the halo-enrichment strategy against minor pixel-level warping errors will also be investigated.

To improve the realism of simulations in the interim, pursuing higher-fidelity radar simulation is planned. The present simulator approximates multipath and ghost targets; the intention is to incorporate more advanced simulation engines, such as the C-Shenron radar engine [17], to model these artefacts more faithfully and to release an updated *OPV2V-R*$^{+}$ split with the richer sensor physics.

Further research will also focus on algorithmic enhancements. The current LiDAR-radar fusion strategy uses a simple concatenation approach; given their distinct distributions, a more advanced fusion method should be researched. Similarly, the framework's query count is fixed a priori. A learned scheduler that implements an adaptive query budget based on scene complexity or link congestion could further reduce average bandwidth. For scenarios where Doppler is unavailable, a Doppler-free fallback using optical flow or LiDAR scene-flow surrogates will be investigated.

Looking at broader applications, occlusion-aware queries are also valuable for motion forecasting. Extending SlimComm to jointly detect and predict trajectories is a promising next step. Finally, while sparse queries reveal less scene detail than full maps, they still leak location cues.

Addressing these directions, with a primary focus on real-world validation and robustness to localization errors, aims to turn SlimComm into a deployable, real-time module for next-generation V2X perception systems.

7. Conclusion

We presented **SlimComm**, a proactive, query-driven framework that unites LiDAR and *4-D* radar for bandwidth-efficient cooperative perception. Guided by motion-centric and confidence-based priors, SlimComm drops sparse queries on dynamic objects and occlusion shadows, prompting neighbours to return only the most informative BEV features. A gated multi-scale deformable-attention block then fuses ego and halo-enriched neighbour features.

Experiments on the new *OPV2V-R* and *Adver-City-R* benchmarks show that SlimComm matches full-map sharing while transmitting just $\sim$10 % of the data, and it consistently outperforms prior sparse-communication baselines. All code, datasets and our radar-simulation toolbox will be released to foster reproducible research.

Remaining challenges, including richer LiDAR–radar coupling, delay-aware fusion, and real-world evaluation, are detailed in Section 6. Addressing them will move SlimComm closer to deployment in next-generation V2X perception systems.

References

[1] Ahmed N. Ahmed, Siegfried Mercelis, and Ali Anwar. Delawarecol: Delay aware collaborative perception. *IEEE Open Journal of Vehicular Technology*, 6:1164–1177, 2025. 2

[2] Igal Bilik. Comparative analysis of radar and lidar technologies for automotive applications. *IEEE Intelligent Transportation Systems Magazine*, 15(1):244–269, 2023. 1

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

[4] 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. 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 3

[6] Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. In *Advances in Neural Information Processing Systems*, 2022. 2, 5, 7

[7] Xun Huang, Jinlong Wang, Qiming Xia, Siheng Chen, Bisheng Yang, Xin Li, Cheng Wang, and Chenglu Wen. V2x-r: Cooperative lidar-4d radar fusion with denoising diffusion for 3d object detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 27390–27400, 2025. 2

[8] Mateus Karvat and Sidney Givigi. Adver-City: Open-Source Multi-Modal Dataset for Collaborative Perception Under Adverse Weather Conditions, 2025. arXiv:2410.06380 [cs]. 2, 6

[9] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast Encoders for Object Detection from Point Clouds. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12689–12697, 2019. 3

[10] 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. 2

[11] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L. Rus, and Song Han. Bevfusion: Multi-task Multi-sensor Fusion with Unified Bird's-eye View Representation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2774–2781, 2023. 1

[12] Yifan Lu, Quanhao Li, Baoan Liu, Mehrdad Dianati, Chen Feng, Siheng Chen, and Yanfeng Wang. Robust collaborative 3d object detection in presence of pose errors. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4812–4818. IEEE, 2023. 5

[13] Kun Qian, Shilin Zhu, Xinyu Zhang, and Li Erran Li. Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 444–453, 2021. 1

[14] Donghao Qiao and Farhana Zulkernine. Adaptive feature fusion for cooperative perception using lidar point clouds. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1186–1195, 2023. 2, 5, 7

[15] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PVRCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection. *Int. J. Comput. Vision*, 131, 2022. 5

[16] Jingyu Song, Lingjun Zhao, and Katherine A Skinner. Lira-fusion: Deep adaptive lidar-radar fusion for 3d object detection. 2024. 2

[17] Satyam Srivastava, BITS Pilani, Jerry Li, Pushkal Mishra, Kshitiz Bansal, and Dinesh Bharadia. A Realistic Radar Simulation Framework for CARLA. 8

[18] Li Wang, Xinyu Zhang, Baowei Xv, Jinzhao Zhang, Rong Fu, Xiaoyu Wang, Lei Zhu, Haibing Ren, Pingping Lu, Jun Li, and Huaping Liu. Interfusion: Interaction-based 4D Radar and LiDAR Fusion for 3D Object Detection. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12247–12253, 2022. 2

[19] 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 *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 605–621. Springer, 2020. 2

[20] Yingjie Wang, Jiajun Deng, Yao Li, Jinshui Hu, Cong Liu, Yu Zhang, Jianmin Ji, Wanli Ouyang, and Yanyong Zhang. Bi-LRFusion: Bi-directional lidar–radar fusion for 3d dynamic object detection. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 13394–13404, 2023. 2

[21] Baowei Xu, Xinyu Zhang, Li Wang, Xiaomei Hu, Zhiwei Li, Shuyue Pan, Jun Li, and Yongqiang Deng. Rpfa-net: a 4d radar pillar feature attention network for 3d object detection. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3061–3066, 2021. 1

[22] Junhao Xu, Yanan Zhang, Zhi Cai, and Di Huang. Cosdh: Communication-efficient collaborative perception via supply-demand awareness and intermediate-late hybridization. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 6834–6843, 2025. 2

[23] Ruoyu Xu and Zhiyu Xiang. Rlnet: Adaptive fusion of 4d radar and lidar for 3d object detection. In *European Conference on Computer Vision*, pages 181–194. Springer, 2025. 2

[24] Runsheng Xu, Yi Guo, Xu Han, Xin Xia, Hao Xiang, and Jiaqi Ma. Opencda: an open cooperative driving automation framework integrated with co-simulation. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1155–1162. IEEE, 2021. 2

[25] 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 *2022 International Conference on Robotics and Automation (ICRA)*, pages 2583–2589, 2022. 2, 5, 7

[26] Kun Yang, Dingkang Yang, Jingyu Zhang, Mingcheng Li, Yang Liu, Jing Liu, Hanqi Wang, Peng Sun, and Liang Song. Spatio-temporal domain awareness for multi-agent collaborative perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23383–23392, 2023. 2, 7

[27] Lei Yang, Xinyu Zhang, Chen Wang, Jun Li, Jiaqi Ma, Zhiying Song, Tong Zhao, Ziying Song, Li Wang, Mo Zhou, Yang Shen, and Chen Lv. V2x-radar: A multi-modal dataset with 4d radar for cooperative perception. *arXiv preprint arXiv:2411.10962*, 2024. 3, 8

[28] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3DSSD:

Point-Based 3D Single Stage Object Detector. In *CVPR*, 2020. 5

[29] Melih Yazgan, Mythra Varun Akkanapragada, and J. Marius Zöllner. Collaborative perception datasets in autonomous driving: A survey. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 2269–2276, 2024. 2

[30] 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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21329–21338. IEEE. 2

[31] Yunshuang Yuan and Monika Sester. Cosense3d: an agent-based efficient learning framework for collective perception. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6, 2024. 2

[32] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 6