PERCEIVE AS YOU RECEIVE: COLLABORATIVE PERCEPTION UNDER PARTIAL RECEPTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Under tight perception deadlines and fluctuating V2X link conditions, an ego agent may need to produce a perception output before collaborative messages are fully received. We study this practical yet underexplored problem of collaborative perception under partial reception. Existing communication-efficient methods primarily optimize message size and accuracy under full reception, without explicitly prioritizing the early delivery of critical evidence or the effective use of partially received messages. We introduce CoDelta, a collaborative perception framework that couples transmission prioritization with ego-centric fusion under partial reception. The sender organizes compact BEV patch representations into a transmission sequence ordered by task relevance. The receiver combines received patches with ego features to generate local additive updates without reconstructing complete collaborator representations. It then aggregates these updates into the ego BEV and refines the resulting representation to restore spatial coherence. Extensive experiments on OPV2V, V2XSet, and DAIR-V2X demonstrate the effectiveness of CoDelta across varying reception ratios. On OPV2V, CoDelta outperforms PragComm by 14.75 percentage points in AP@0.70 at a 10% reception ratio. Code will be released upon acceptance.

1 INTRODUCTION

Collaborative perception extends an agent's perceptual coverage beyond the occlusions and viewpoint limitations of its onboard sensors by leveraging observations from others (Xia et al., 2025; Xu et al., 2025b; Gao et al., 2025; Wen et al., 2026). The challenge, however, is not merely what to communicate, but whether the communicated information arrives in time to affect the current decision. In collaborative driving, the available end-to-end budget is often around 100 ms, whereas the volume of information deliverable within this interval varies dynamically with V2X channel and traffic conditions (Qiu et al., 2021; Huang et al., 2023). Consequently, collaborative messages may remain incomplete when a perception output is required, forcing the receiver to perform perception using the collaborative information available by the deadline. We refer to this setting as collaborative perception under partial reception. This raises a realistic yet underexplored question: how can current perception benefit from a collaborative message that remains incomplete at the perception deadline?

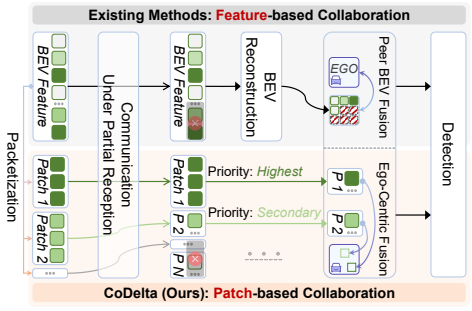


Figure 1: Conceptual comparison under partial reception. Existing methods mainly optimize message size and full-reception accuracy. CoDelta prioritizes local BEV patches for transmission and uses the received patch information to generate local Delta updates, enabling collaboration without complete message reception.

Existing communication-efficient collaborative perception methods largely optimize the trade-off between perception accuracy and communication cost when collaborator messages are fully available (Yang et al., 2023b;a; Tang et al., 2025). For example, Where2comm selects informative regions for transmission, and PragComm further compresses the selected regions using discrete indices (Hu

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

et al., 2022; 2026). Compact messages can increase the likelihood of complete reception, but they do not ensure that partially received information remains useful for collaborative perception.

Time-constrained human teamwork offers a useful intuition (Pfaff, 2012; Avoyan et al., 2025). A good teammate does not attempt to transfer their entire observation stream. Instead, they communicate the evidence most likely to revise the other person's current judgment or compensate for missing information, and add detail only when time permits. By the same logic, the value of collaborative perception depends not only on message volume, but on whether useful evidence arrives early enough to affect the current prediction. Motivated by this observation, we shift collaborative perception away from recovering peer collaborator representations and toward prioritizing evidence that is most useful for updating the ego agent's current perception.

In this paper, we propose CoDelta, a collaborative perception framework for partial reception that jointly designs transmission priority at the collaborator and ego-centric fusion at the receiver. Specifically, Priority-Aware Packetization compactly encodes local BEV patches and prioritizes them for transmission based on task relevance. Ego-Centric Delta Fusion combines each received patch with the corresponding ego patch to predict a local Delta, a learned feature residual that specifies how the ego patch should be adjusted using the collaborator information. These corrections are aggregated into the ego BEV, and the updated representation is refined with contextual interaction. Consequently, CoDelta improves ego perception under partial reception without reconstructing complete collaborator representations.

We evaluate CoDelta on OPV2V, V2XSet, and DAIR-V2X across different reception ratios. The results show that i) CoDelta outperforms the evaluated baselines under partial reception, with only one quarter of each message needed to approach full-reception accuracy; and ii) CoDelta achieves higher perception performance under full reception than standard collaboration with uncompressed features while using over 3000 times less communication.

To sum up, our contributions are threefold. *First*, we study collaborative perception under partial reception and emphasize the need to make partially received information useful for perception beyond simply reducing message size. *Second*, we propose CoDelta, a collaborative perception framework that jointly designs transmission priority and ego-centric fusion, prioritizing task-relevant patches for transmission and using the received patch information to generate local ego updates. *Third*, extensive experiments demonstrate that CoDelta effectively exploits partially received messages at inference time across diverse reception conditions.

2 Related Work

2.1 Communication-Efficient Collaborative Perception

Despite advances in single-agent perception through multimodal fusion and visual reasoning (Chen et al., 2026; Zeng et al., 2026), collaborative perception extends observable coverage by exchanging complementary information among agents (Wang et al., 2020). Communication-efficient methods mainly reduce redundant transmission through selective communication or compact message representations (Xu et al., 2025a). Where2comm selects spatial regions using confidence maps (Hu et al., 2022), while ERMVP samples representative feature vectors for efficient communication (Zhang et al., 2024). PragComm encodes task-critical features as compact dictionary indices (Hu et al., 2026), and InfoCom extracts minimal task-relevant representations through information bottleneck (Wei et al., 2026). GenComm transmits compact spatial conditions and generates complete collaborative features at the receiver for heterogeneous collaboration (Zhou et al., 2026). These methods improve the accuracy–communication trade-off by optimizing complete messages under constrained communication budgets. However, how to exploit an incomplete message when perception must proceed before transmission completes remains underexplored.

2.2 Collaborative Perception under Imperfect Communication

Another line of work improves collaborative perception under communication delay, asynchrony, and interruption. Latency-aware methods align asynchronous features or compensate for spatio-temporal drift before fusion (Lei et al., 2022; Wang & Nordström, 2025; Dai et al., 2025), while interruption-aware methods recover missing collaborative information from historical observa-

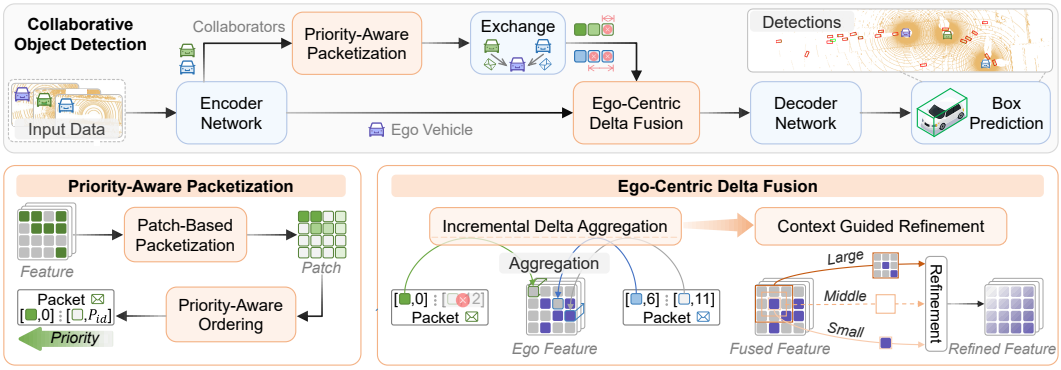


Figure 2: Overview of CoDelta. CoDelta reorganizes communication under partial reception through Priority-Aware Packetization and Ego-Centric Delta Fusion. The former compactly encodes BEV patches and prioritizes them for transmission. The latter integrates received patch information into the ego BEV through local additive Delta updates, followed by contextual refinement.

tions (Ren et al., 2024). These methods mitigate communication degradation by correcting stale features or reconstructing missing information. CoDelta instead studies the practical yet underexplored problem of partial reception, where perception may proceed before the complete message arrives. Rather than compensating for unavailable information, CoDelta prioritizes task-relevant patches for transmission and uses the received patch information to generate local ego updates.

3 Method

3.1 Problem Setup

We consider an ego-centric collaborative perception setting with N collaborators and focus on 3D object detection under partial reception. We denote the ego BEV feature by $\mathbf{F}_{\text{ego}}$ and the collaborator BEV features aligned to the ego frame by $\{\mathbf{F}_i\}_{i=1}^{N}$. All features have dimensions $C \times H \times W$. In this setting, collaboration relies on the ego agent's representation and the portion of each collaborator's message received by the perception deadline.

In CoDelta, each selected BEV patch is encoded into a compact payload, which is paired with its spatial index to form a packet. Collaborator i organizes its message as a priority-ordered packet sequence $\mathcal{P}_i = (p_i^1, \ldots, p_i^{K_i})$, where K_i is the number of selected packets. We assume that the packets available by the perception deadline form an initial segment of this sequence, which we call the received prefix. We represent reception conditions using a common ratio $r \in [0, 1]$, with the received prefix defined as $\mathcal{P}_i^{\le r} = (p_i^1, \ldots, p_i^{\lceil rK_i \rceil})$ and an empty sequence at $r = 0$. The objective is to maximize collaborative detection performance across reception conditions:

$$\max_{\theta} \; \mathbb{E}_{r\sim\mathcal{R}} \left[g\big(\Psi_\theta(\mathbf{F}_r), \mathbf{Y}\big) \right], \quad \mathbf{F}_r := \Phi_\theta\big(\mathbf{F}_{\text{ego}}, \{\mathcal{P}_i^{\le r}\}_{i=1}^{N}\big), \quad \mathcal{P}_i := \Gamma_\theta(\mathbf{F}_i). \tag{1}$$

Here, Γ_θ constructs the ordered packet sequence, Φ_θ integrates received prefixes into the ego representation, and Ψ_θ predicts detections. The trainable parameters are denoted by θ, $\mathcal{R}$ is the distribution over reception ratios, and g evaluates detection performance against labels $\mathbf{Y}$.

3.2 Overview of CoDelta

CoDelta is a collaborative perception framework for partial reception, where only a prefix of each transmitted message may be available by the perception deadline. It converts the available collaborator information into prioritized local updates of the ego BEV representation. As shown in Fig. 2, CoDelta comprises two main modules: Priority-Aware Packetization and Ego-Centric Delta Fusion. Priority-Aware Packetization partitions collaborator BEV features into uniform local patches and encodes each patch as a compact payload. Each payload is paired with its spatial index to form a packet, and selected packets are ordered by predicted task relevance. Ego-Centric Delta Fusion

combines each received payload with the corresponding ego patch to generate a local Delta update. The update is applied directly to the ego BEV without reconstructing the complete collaborator BEV feature. The updated representation is subsequently refined using its surrounding context and fed to the decoder for final prediction, enabling useful collaboration from incomplete message prefixes.

3.3 Priority-Aware Packetization

Priority-Aware Packetization compactly encodes local BEV patches and determines which patches to transmit and in what order. It comprises two stages: Patch-Based Packetization and Priority-Aware Ordering.

Patch-Based Packetization. When reception ends at the perception deadline, treating the entire BEV map as an indivisible transmission unit makes its perceptual utility contingent on complete reception. Patch-level packets allow the regions that have arrived to inform perception. Let $\mathbf{F}_i \in \mathbb{R}^{C\times H\times W}$ denote the BEV feature map of the i-th collaborator. We partition $\mathbf{F}_i$ into a regular grid of patches with size $H_p \times W_p$, yielding $H_g = \lceil H/H_p \rceil$ and $W_g = \lceil W/W_p \rceil$ patches along the two spatial dimensions. Boundary patches are zero-padded when necessary. This partitioning allows local BEV regions to be selected and encoded independently, rather than treating the entire feature map as a single communication unit. For the patch at row u and column v, where $u = 0, \ldots, H_g - 1$ and $v = 0, \ldots, W_g - 1$, its valid spatial support is defined as

$$\Omega_i^{u,v} = \big\{(x,y) \,\big|\, uH_p \le x < \min\big((u+1)H_p, H\big),\ vW_p \le y < \min\big((v+1)W_p, W\big)\big\}. \quad (2)$$

After zero-padding the extracted region to $H_p \times W_p$, we denote the corresponding patch feature by $\mathbf{z}_i^{u,v} = \mathrm{Pad}_{H_p,W_p}\big(\mathbf{F}_i[\Omega_i^{u,v}]\big) \in \mathbb{R}^{C\times H_p\times W_p}$. To reduce communication cost, a shared payload head f_{pay} maps each patch feature to a compact single-channel payload $\mathbf{c}_i^{u,v} = f_{\mathrm{pay}}\big(\mathbf{z}_i^{u,v}\big) \in \mathbb{R}^{1\times H_p\times W_p}$. Each candidate packet stores the patch location and its payload as $p_i^{u,v} = \big(\ell_i^{u,v}, \mathbf{c}_i^{u,v}\big)$, where $\ell_i^{u,v} = (u,v)$ denotes the patch index. These candidate packets are ranked by task relevance in the subsequent ordering stage.

Priority-Aware Ordering. Patch-based encoding alone does not establish a transmission order. Under partial reception, prioritizing task-relevant patches can improve the utility of the information available by the perception deadline. Priority-Aware Ordering therefore ranks patches by predicted task relevance without requiring additional communication rounds. A sender-side saliency head $f_{\mathrm{sal}}(\cdot)$ produces BEV-resolution logits $\hat{\mathbf{S}}_i = f_{\mathrm{sal}}(\mathbf{F}_i) \in \mathbb{R}^{1\times H\times W}$, which are converted into a saliency map $\mathbf{S}_i = \sigma(\hat{\mathbf{S}}_i)$. Because the detection objective provides limited guidance on transmission order, we train the saliency head through internal distillation. The stop-gradient foreground response of the standard detection head serves as a task-aware teacher for saliency prediction. At inference, the sender derives patch priorities from a saliency map predicted solely from $\mathbf{F}_i$, requiring no receiver feedback. We aggregate the saliency responses within the valid support $\Omega_i^{u,v}$ to obtain a priority score for each patch:

$$q_i^{u,v} = \frac{1}{2|\Omega_i^{u,v}|} \sum_{(x,y)\in\Omega_i^{u,v}} \mathbf{S}_i(x,y) + \frac{1}{2} \max_{(x,y)\in\Omega_i^{u,v}} \mathbf{S}_i(x,y). \quad (3)$$

Mean–max aggregation combines the average saliency and peak response within each patch. To explicitly limit the communication volume, we select the top-K_i patches by priority score and arrange their corresponding packets in descending score order, yielding $\mathcal{P}_i = (p_i^1, \ldots, p_i^{K_i})$, with $q_i^1 \ge \cdots \ge q_i^{K_i}$. Here, q_i^k denotes the priority score associated with the k-th packet in the ordered sequence. The sequence is then transmitted to the ego agent over a V2X communication link.

3.4 Ego-Centric Delta Fusion

Ego-Centric Delta Fusion uses the received patch information to generate local additive updates to the ego BEV without reconstructing the complete collaborator feature map. It comprises two stages: Incremental Delta Aggregation and Context-Guided Refinement.

Incremental Delta Aggregation. Conventional collaborative perception treats collaboration as peer collaborator representation recovery followed by fusion. Such a pipeline cannot directly exploit incomplete message prefixes and may incur additional processing latency. CoDelta reformulates collaboration as the incremental accumulation of local Delta updates on the ego BEV representation. Given the received prefixes $\{\mathcal{P}_i^{\le r}\}_{i=1}^N$ at reception ratio r, consider a packet $p_i^{u,v} = (\ell_i^{u,v}, \mathbf{c}_i^{u,v})$, where $\ell_i^{u,v}$ denotes its patch location and $\mathbf{c}_i^{u,v}$ its payload. The location index $\ell_i^{u,v}$ identifies the corresponding ego patch $\mathbf{e}^{u,v} = \mathbf{F}_{\text{ego}}[\Omega^{u,v}] \in \mathbb{R}^{C\times H_p\times W_p}$. An ego-conditioned Delta Predictor f_Δ combines the ego patch with the received payload to generate an additive Delta update $\Delta\mathbf{e}_i^{u,v} = f_\Delta\big(\mathbf{e}^{u,v}, \mathbf{c}_i^{u,v}\big) \in \mathbb{R}^{C\times H_p\times W_p}$. For a single received packet, this residual updates the corresponding ego patch as $\mathbf{e}^{u,v} + \Delta\mathbf{e}_i^{u,v}$, with the received payload determining the adjustment together with the existing ego features. All local Delta updates are computed with respect to the same ego feature map and scattered to their corresponding locations:

$$\mathbf{F}'_r = \mathbf{F}_{\text{ego}} + \sum_{i=1}^{N} \sum_{(u,v)\in\mathcal{I}_i^r} \text{Scatter}(\Delta\mathbf{e}_i^{u,v}, \ell_i^{u,v}), \tag{4}$$

where $\mathcal{I}_i^r$ denotes the patch indices contained in the received prefix $\mathcal{P}_i^{\le r}$. Each local Delta is computed from its own received payload and the same ego BEV, enabling parallel computation without reconstructing the complete collaborator feature map.

Context-Guided Refinement. Incremental Delta aggregation preserves prefix usability because each local Delta is computed independently from the same ego BEV. However, the aggregation only performs location-wise addition and does not model interactions among updated patches or between updated patches and their neighboring features. Sparse or incomplete updates may therefore cause boundary discontinuities and weaken contextual coordination in the aggregated representation. Context-Guided Refinement addresses this limitation with a lightweight context interaction module:

$$\mathbf{F}_r = f_{\text{ref}}(\mathbf{F}'_r), \qquad f_{\text{ref}} : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{C\times H\times W}. \tag{5}$$

Here, f_{ref} denotes the refinement network, and $\mathbf{F}_r$ is the final collaborative representation at reception ratio r. The network operates solely on $\mathbf{F}'_r$ to model interactions between updated regions and their surrounding context, without requiring additional collaborator information. The resulting representation $\mathbf{F}_r$ is then passed to the detection head for final object detection. Architectural details of f_{pay}, f_{sal}, f_Δ, and f_{ref} are provided in the appendix.

3.5 Training and Inference with Partial Reception

CoDelta's training objective consists of three components. The main objective is the standard detection loss $\mathcal{L}_{\text{det}}$ Lu et al. (2023); Xu et al. (2022b). Let $\hat{\mathbf{S}}_i = f_{\text{sal}}(\mathbf{F}_i)$ denote the saliency logits. During training, internal distillation uses a stop-gradient teacher derived from the detection head. Specifically, the teacher map is defined as $\mathbf{T}_i = \text{sg}(\max_a \sigma(\Psi_{\text{cls}}(\mathbf{F}_i))_a)$, where a indexes the anchor dimension, $\sigma(\cdot)$ denotes the sigmoid function, and $\text{sg}(\cdot)$ stops gradients from flowing into the teacher. The saliency head is trained with the task-aware distillation loss $\mathcal{L}_{\text{distill}} = \sum_i f_{\text{BCE}}(\hat{\mathbf{S}}_i, \mathbf{T}_i)$. We further regularize the magnitude of local Delta updates with $\mathcal{L}_{\text{delta}} = \sum_{i,u,v} \|\Delta\mathbf{e}_i^{u,v}\|_2^2$. The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\text{det}} + \lambda_{\text{distill}}\mathcal{L}_{\text{distill}} + \lambda_{\text{delta}}\mathcal{L}_{\text{delta}}, \tag{6}$$

where λ_{distill} and λ_{delta} control the auxiliary loss weights.

At inference time, CoDelta processes the available packet prefixes through Delta aggregation and context-guided refinement for collaborative object detection. We simulate partial reception by retaining the first $\lceil rK_i \rceil$ packets from each collaborator's ordered sequence.

4 Experiments

4.1 Experimental Setup

Datasets and Evaluation Metrics. We evaluate CoDelta on three representative collaborative perception datasets: OPV2V (Xu et al., 2022b), V2XSet (Xu et al., 2022a), and DAIR-V2X (Yu et al.,

Table 1: Detection performance of CoDelta and baselines under partial reception ($r < 1$). CoDelta consistently outperforms the evaluated baselines across reception ratios. Best results are in **bold**.

Reception Ratio r		0.01		0.05		0.10		0.25		0.50		0.75	
	Method	AP@50	AP@70	AP@50	AP@70	AP@50	AP@70	AP@50	AP@70	AP@50	AP@70	AP@50	AP@70
OPV2V	Standard Colla.	0.8673	0.7502	0.8699	0.7517	0.8716	0.7537	0.8757	0.7609	0.9449	0.8775	0.9621	0.9178
	Where2comm	0.8866	0.7903	0.8870	0.7912	0.8881	0.7933	0.8979	0.8081	0.9236	0.8484	0.9392	0.8742
	ERMVP	0.8523	0.7268	0.8527	0.7269	0.8617	0.7375	0.8731	0.7596	0.9090	0.8148	0.9408	0.8907
	GenComm	0.7969	0.6881	0.8052	0.6966	0.8060	0.6971	0.8439	0.7350	0.8992	0.8173	0.9506	0.8944
	PragComm	0.8559	0.7462	0.8584	0.7466	0.8599	0.7486	0.8643	0.7530	0.9415	0.8721	0.9643	0.9116
	InfoCom	0.8534	0.7465	0.8544	0.7476	0.8566	0.7490	0.8590	0.7535	0.9397	0.8738	0.9631	0.9163
	CoDelta (Ours)	**0.9105**	**0.8294**	**0.9341**	**0.8721**	**0.9483**	**0.8961**	**0.9674**	**0.9261**	**0.9711**	**0.9314**	**0.9713**	**0.9328**
V2XSet	Standard Colla.	0.7960	0.6633	0.7991	0.6653	0.8057	0.6697	0.8220	0.6854	0.8947	0.7950	0.9157	0.8363
	Where2comm	0.8405	0.7188	0.8409	0.7191	0.8418	0.7195	0.8433	0.7206	0.8473	0.7237	0.8532	0.7316
	ERMVP	*OOM*		*OOM*		*OOM*		*OOM*		*OOM*		*OOM*	
	GenComm	0.7261	0.6128	0.7429	0.6239	0.7486	0.6246	0.7754	0.6499	0.8699	0.7672	0.9154	0.8351
	PragComm	0.8000	0.6861	0.8018	0.6871	0.8053	0.6910	0.8176	0.7026	0.8934	0.7958	0.9188	0.8338
	InfoCom	0.7661	0.6408	0.7685	0.6430	0.7730	0.6475	0.7929	0.6653	0.8893	0.7920	0.9217	0.8418
	CoDelta (Ours)	**0.8539**	**0.7322**	**0.8848**	**0.7750**	**0.9052**	**0.8059**	**0.9266**	**0.8419**	**0.9340**	**0.8544**	**0.9356**	**0.8572**
DAIR-V2X	Standard Colla.	0.6753	0.5654	0.6808	0.5682	0.6847	0.5693	0.6948	0.5749	0.7397	0.6064	0.7736	0.6297
	Where2comm	0.6845	0.5690	0.6865	0.5705	0.6878	0.5705	0.6916	0.5716	0.7079	0.5812	0.7396	0.6009
	ERMVP	0.6768	0.5650	0.6836	0.5696	0.6861	0.5711	0.7008	0.5802	0.7208	0.5947	0.7605	0.6258
	GenComm	0.6437	0.5378	0.6427	0.5359	0.6437	0.5356	0.6398	0.5320	0.6864	0.5660	0.7424	0.6077
	PragComm	0.6870	0.5692	0.6928	0.5737	0.6951	0.5743	0.7033	0.5789	0.7454	0.6114	0.7750	0.6337
	InfoCom	0.6752	0.5654	0.6789	0.5676	0.6822	0.5696	0.6901	0.5729	0.7334	0.6062	0.7667	0.6307
	CoDelta (Ours)	**0.7262**	**0.6007**	**0.7589**	**0.6258**	**0.7733**	**0.6361**	**0.7831**	**0.6435**	**0.7843**	**0.6463**	**0.7838**	**0.6469**

2022). OPV2V and V2XSet provide large-scale simulated collaborative perception scenarios, while DAIR-V2X represents real-world vehicle-infrastructure collaboration. For DAIR-V2X, we use the expanded annotations introduced in Lu et al. (2023). We follow the official data splits unless otherwise stated. We evaluate detection accuracy, processing latency, and communication cost. Detection accuracy is measured by average precision at IoU thresholds of 0.50 and 0.70, denoted as AP@50 and AP@70. Processing latency measures the time required for communication processing and feature fusion, reported in milliseconds under the same hardware and evaluation settings. Communication cost is measured as transmitted data volume per message and reported in KiB or MiB. For CoDelta, this includes payloads and patch location indices.

Implementation Details. Following prior works (Lu et al., 2023; Wei et al., 2026), we implement all methods in a unified codebase with PointPillars as the shared 3D detector. We first pretrain a base collaborative perception model using complete collaborator feature maps without the CoDelta modules. We then introduce these modules and finetune CoDelta for 20 epochs using Adam with a batch size of 4, an initial learning rate of 0.002, and a multistep learning-rate decay schedule. The auxiliary loss weights are set to $\lambda_{\text{distill}} = 0.5$ and $\lambda_{\text{delta}} = 0.001$. The BEV feature dimensions are $C \times H \times W = 256 \times 100 \times 352$ for OPV2V and V2XSet, and $256 \times 100 \times 252$ for DAIR-V2X. The Delta Predictor f_Δ and the Context-Guided Refinement module f_{ref} each contain two stacked basic blocks. We use a patch size of $(H_p, W_p) = (8, 8)$. By default, each collaborator selects the 128 highest-priority patches for transmission, yielding $K_i = 128$ packets per complete message. The single-channel payloads are uniformly quantized to 8 bits before transmission. We evaluate reception ratios $r \in \{0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0\}$, ranging from severely limited reception to full reception of the selected message.

Compared Methods. We compare CoDelta with No Collaboration, which uses only ego information, Late Collaboration, which fuses detection outputs from individual agents, Standard Collaboration, which fuses complete collaborator BEV features, and five communication-efficient methods: Where2comm (Hu et al., 2022), ERMVP (Zhang et al., 2024), GenComm (Zhou et al., 2026), PragComm (Hu et al., 2026), and InfoCom (Wei et al., 2026). For partial-reception evaluation, we arrange their messages in spatial order and retain the prefix corresponding to each reception ratio. Missing regions are zero-filled before fusion. CoDelta transmits packets in order of decreasing learned priority. Since equal reception ratios do not imply equal received data volumes, we also compare CoDelta with InfoCom at matched communication volumes.

Table 2: Detection performance and communication volume under full reception ($r = 1$). Volume denotes the complete message size per collaborator. Best detection results are in **bold**.

Method	OPV2V			V2XSet			DAIR-V2X		
	Volume	AP@50	AP@70	Volume	AP@50	AP@70	Volume	AP@50	AP@70
No Colla.	0	0.8464	0.7288	0	0.7719	0.6222	0	0.6844	0.5665
Late Colla.	7.031 KB	0.9499	0.8495	7.031 KB	0.9076	0.7120	7.031 KB	0.7258	0.5813
Standard Colla.	34.375 MiB	0.9653	0.9229	34.375 MiB	0.9212	0.8426	24.609 MiB	0.7843	0.6353
Where2comm	3.468 MiB	0.9463	0.8817	3.468 MiB	0.8604	0.7417	2.481 MiB	0.7540	0.6074
ERMVP	0.771 MiB	0.9557	0.9127	0.771 MiB	*OOM*		0.551 MiB	0.7791	0.6324
GenComm	1.074 MiB	0.9642	0.9084	1.074 MiB	0.9247	0.8445	0.769 MiB	0.7659	0.6235
PragComm	0.732 MiB	0.9677	0.9149	0.732 MiB	0.9252	0.8415	0.522 MiB	**0.7859**	0.6410
InfoCom	38.813 KiB	0.9650	0.9202	38.813 KiB	0.9273	0.8485	26.840 KiB	0.7772	0.6371
CoDelta (Ours)	8.250 KiB	**0.9713**	**0.9333**	8.250 KiB	**0.9357**	**0.8578**	8.250 KiB	0.7843	**0.6489**

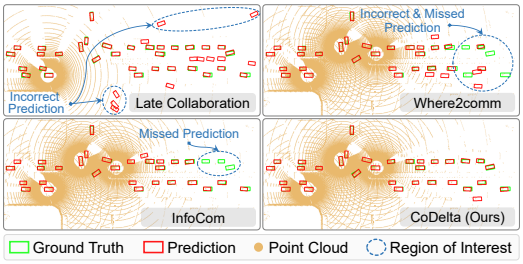


Figure 3: Qualitative comparison of CoDelta and baselines. CoDelta produces more complete and stable predictions.

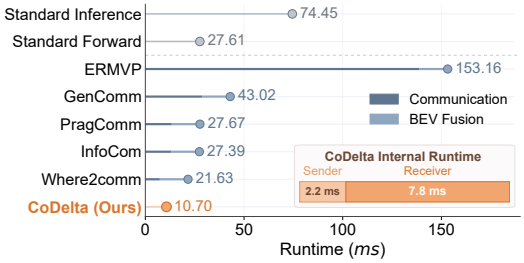


Figure 4: Runtime comparison of CoDelta and baselines. CoDelta achieves the lowest overall runtime.

4.2 Main Results under Partial Reception

Quantitative Comparison. Tables 1 and 2 summarize detection performance under partial and full reception, respectively, and highlight two advantages of CoDelta. *First*, CoDelta consistently outperforms the evaluated baselines under partial reception, particularly at low reception ratios. At $r = 0.1$, CoDelta exceeds the strongest baseline in AP@70 by 10.28, 8.64, and 6.18 percentage points on OPV2V, V2XSet, and DAIR-V2X, respectively. *Second*, CoDelta approaches its own full-reception accuracy with only a partial message prefix. At $r = 0.25$, its AP@70 is just 0.72, 1.59, and 0.54 percentage points below the corresponding results at $r = 1$ on the three datasets.

As shown in Table 2, CoDelta substantially reduces communication overhead while maintaining high detection accuracy. Each collaborator transmits only 8.250 KiB per complete message. Compared with Standard Collaboration, CoDelta reduces the complete message size by factors of approximately 4,267 on OPV2V and V2XSet and 3,054 on DAIR-V2X. Compared with InfoCom, the most communication-efficient baseline, CoDelta reduces the complete message size by 78.7% on OPV2V and V2XSet and 69.3% on DAIR-V2X. Although designed for partial reception, CoDelta remains competitive with all evaluated methods under full reception while requiring less communication. For example, on V2XSet, it surpasses InfoCom by 0.84 and 0.93 percentage points in AP@50 and AP@70, respectively.

Qualitative Comparison. Fig. 3 compares detection results under partial reception. In the highlighted regions, Late Collaboration produces false positives, Where2comm exhibits both false positives and missed detections, and InfoCom misses several objects. CoDelta detects the corresponding objects with fewer false positives, illustrating improved detection coverage in the displayed scene. These observations are consistent with the quantitative results under partial reception.

4.3 Further Analysis

Runtime Analysis. Fig. 4 compares the processing latency of CoDelta with baselines, covering communication processing and feature fusion. CoDelta achieves the lowest reported latency of 10.70 ms, while the fastest baseline Where2comm requires 21.63 ms. Several baselines spend a

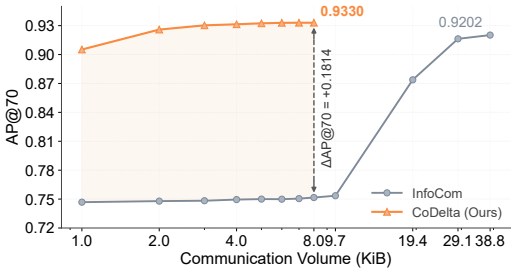


Figure 5: Comparison of CoDelta and InfoCom under matched communication volumes. CoDelta consistently achieves higher accuracy.

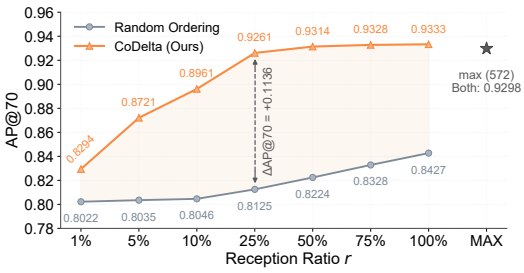


Figure 6: Effect of packet ordering on AP@70. CoDelta's priority ordering consistently outperforms random ordering under partial reception.

Table 3: Ablation studies on the OPV2V dataset.

Variant	AP@50	AP@70
Priority-aware ordering		
Random ordering	0.9175	0.8427
Mean-only priority score	0.9621	0.9150
Max-only priority score	0.9664	0.9237
Component ablation		
w/o internal distillation	0.9438	0.8727
w/o refinement	0.9448	0.8310
w/o Delta regularization	0.9694	0.9274
Full model	**0.9713**	**0.9333**

Table 4: Effect of the packet budget on OPV2V.

Budget K_i	Communication	AP@50	AP@70
32	2.063 KiB	0.9549	0.8986
64	4.125 KiB	0.9676	0.9224
128	8.250 KiB	0.9713	0.9333
512	33.000 KiB	0.9733	0.9382

Table 5: Effect of quantization bits on OPV2V.

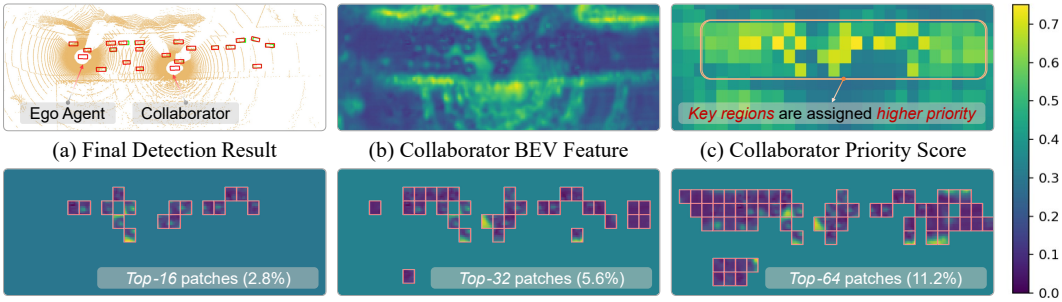
Bits b	Communication	AP@50	AP@70
2	2.250 KiB	0.9656	0.9093
4	4.250 KiB	0.9716	0.9263
8	8.250 KiB	0.9713	0.9333

substantial portion of their processing time on BEV fusion. Thus, lower communication volume does not necessarily translate into lower processing latency. CoDelta directly integrates received payloads through local Delta updates and context-guided refinement, avoiding complete collaborator BEV reconstruction followed by BEV fusion. Its runtime breakdown comprises 2.2 ms for sender-side packetization, 7.8 ms for receiver-side Delta aggregation and refinement, and 0.7 ms for other processing overhead.

Comparison at Matched Communication Volumes. To complement the comparison at equal reception ratios in Table 1, we compare CoDelta with InfoCom on OPV2V at matched received data volumes in Fig. 5. InfoCom is selected because it has the smallest complete message size among the baselines. Across the evaluated communication volumes from 1 to 8 KiB, CoDelta achieves an absolute AP@70 gain of more than 0.15 over InfoCom. This comparison confirms that CoDelta's advantage under partial reception persists after controlling for the amount of received data.

Effect of Priority-Aware Ranking. Fig. 6 compares learned and random ranking on OPV2V. We replace the learned priority scores with random scores to select patches under the same packet budget and evaluate both strategies at the same reception ratios. CoDelta improves rapidly as reception increases and approaches its full-reception accuracy at $r = 0.25$. Its AP@70 advantage over random ranking grows from 2.72 to 11.36 percentage points as r increases from 0.01 to 0.25 and remains substantial at higher reception ratios. When all 572 BEV patches are received rather than the default 128 selected patches, both strategies achieve an AP@70 of 0.9298. These results show that priority-based selection improves detection accuracy for the same number of received patches.

Ablation Studies. Table 3 evaluates the main design choices of CoDelta on OPV2V at $r = 1$. Additional ablation results under partial reception are provided in the appendix. The Random ordering variant reduces AP@70 from 0.9333 to 0.8427, highlighting the benefit of learned priorities for selecting informative patches under a fixed packet budget. Mean–max aggregation achieves an AP@70 of 0.9333, outperforming mean-only and max-only aggregation at 0.9150 and 0.9237, respectively. This comparison supports combining average saliency and peak responses for patch prioritization. Removing Context-Guided Refinement reduces AP@70 by 10.23 percentage points, compared with 2.65 for AP@50. The larger degradation at the stricter IoU threshold suggests that contextual refinement helps improve localization accuracy after local Delta updates. Internal dis-



(a) Final Detection Result (b) Collaborator BEV Feature (c) Collaborator Priority Score

(d) Spatial Coverage of Received Patches at Different Reception Rates

Figure 7: Visualization of sender-side prioritization in CoDelta. Foreground and task-relevant regions receive higher priority, allowing earlier prefixes to cover the most informative local areas first.

tillation improves AP@70 from 0.8727 to 0.9333 by providing explicit foreground supervision for patch prioritization. Delta regularization offers a smaller gain, with AP@70 decreasing to 0.9274 when the penalty on local update magnitudes is removed.

Effect of the Packet Budget. Table 4 evaluates different packet budgets on OPV2V at full reception ($r = 1$). Larger budgets improve detection accuracy but increase communication volume proportionally. Increasing K_i from 64 to 128 improves AP@50 and AP@70 by 0.37 and 1.09 percentage points, respectively, while further increases yield smaller accuracy gains. With $K_i = 128$, CoDelta requires 8.250 KiB per collaborator, reducing communication volume by 75% relative to $K_i = 512$ with AP@50 and AP@70 differences of only 0.0020 and 0.0049. We therefore adopt $K_i = 128$ as the default to balance detection accuracy and communication cost.

Effect of Quantization Precision. Table 5 evaluates payload quantization precision on OPV2V at full reception ($r = 1$). Using 4-bit instead of 8-bit quantization reduces communication volume from 8.250 to 4.250 KiB while retaining nearly identical AP@50, with an AP@70 decrease of 0.70 percentage points. Further reducing the precision to 2 bits lowers the volume to 2.250 KiB but incurs a larger accuracy loss. We adopt 8-bit quantization by default for its higher AP@70, while 4-bit quantization provides a more compact option with a limited accuracy penalty.

Visualization of Patch Prioritization. Fig. 7 visualizes how CoDelta prioritizes collaborator patches. The collaborator BEV features in Fig. 7(b) provide spatial context for the patch-level priority scores in Fig. 7(c), which assign higher scores to selected regions of the scene. Fig. 7(d) shows the spatial coverage of the top-16, top-32, and top-64 patches. Smaller prefixes concentrate on high-priority regions, while larger prefixes extend coverage to additional areas. Together with the detection result in Fig. 7(a), these visualizations illustrate how CoDelta concentrates limited received content in prioritized regions and expands spatial coverage as more packets become available.

5 Conclusion

We presented CoDelta for collaborative perception under partial reception, where collaborative detection relies on the message prefixes available by the perception deadline. CoDelta combines Priority-Aware Packetization with Ego-Centric Delta Fusion to prioritize task-relevant patches and integrate received information as local additive updates to the ego BEV, without reconstructing complete collaborator representations. Experiments on three datasets demonstrate that CoDelta achieves strong detection performance from incomplete message prefixes with compact messages and low processing latency. These results highlight the value of making partially received information directly usable for collaborative perception.

AI use statement

We used generative AI tools to improve the language and clarity of author-prepared manuscript content and to assist with code debugging. The authors reviewed all AI-assisted text for technical accuracy and checked code changes through inspection and testing. We take full responsibility for the final content of this work.

REFERENCES

Ala Avoyan, Haoran He, and Kelin Lu. Team work under time constraints. *Journal of Economic Behavior & Organization*, 237:107170–107184, 2025.

Zhichao Chen, Keshun You, Jie Yang, Lifang Chen, Fan Li, Zhicheng Feng, and Limin Jia. A sparse-to-dense guided fusion framework for three-dimensional object detection in railway environments. *Engineering Applications of Artificial Intelligence*, 178:115095, 2026.

Penglin Dai, Hao Zhou, Quanmin Wei, Xiao Wu, Zhanbo Sun, and Zhaofei Yu. Cooperative perception of multi-agents under the spatio-temporal drift issue. *IEEE Transactions on Intelligent Transportation Systems*, 27(1):1485–1498, 2025.

Xiangbo Gao, Yuheng Wu, Rujia Wang, Chenxi Liu, Yang Zhou, and Zhengzhong Tu. Langcoop: Collaborative driving with language. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 4226–4237, 2025.

Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation. *Advances in neural information processing systems*, 35:1140–1156, 2022.

Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *Advances in neural information processing systems*, 35:4874–4886, 2022.

Yue Hu, Xianghe Pang, Xiaoqi Qin, Yonina C Eldar, Siheng Chen, Ping Zhang, and Wenjun Zhang. Pragmatic communication in multi-agent collaborative perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Tao Huang, Jianan Liu, Xi Zhou, Dinh C Nguyen, Mostafa Rahimi Azghadi, Yuxuan Xia, Qing-Long Han, and Sumei Sun. V2x cooperative perception for autonomous driving: Recent advances and challenges. *arXiv preprint arXiv:2310.03525*, 2023.

Zixing Lei, Shunli Ren, Yue Hu, Wenjun Zhang, and Siheng Chen. Latency-aware collaborative perception. In *European Conference on Computer Vision*, pp. 316–332, 2022.

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 *IEEE International Conference on Robotics and Automation*, pp. 4812–4818, 2023.

Mark S Pfaff. Negative affect reduces team awareness: The effects of mood and stress on computer-mediated team communication. *Human Factors*, 54(4):560–571, 2012.

Hang Qiu, Po-Han Huang, Namo Asavisanu, Xiaochen Liu, Konstantinos Psounis, and Ramesh Govindan. Autocast: scalable infrastructure-less cooperative perception for distributed collaborative driving. In *International Conference on Mobile Systems, Applications and Services*, 2021.

Shunli Ren, Zixing Lei, Zi Wang, Mehrdad Dianati, Yafei Wang, Siheng Chen, and Wenjun Zhang. Interruption-aware cooperative perception for v2x communication-aided autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 9(4):4698–4714, 2024.

Zongheng Tang, Yi Liu, Yifan Sun, Yulu Gao, Jinyu Chen, Runsheng Xu, and Si Liu. Cost: Efficient collaborative perception from unified spatiotemporal perspective. In *IEEE/CVF International Conference on Computer Vision*, pp. 1120–1129, 2025.

Junjie Wang and Tomas Nordström. Latency robust cooperative perception using asynchronous feature fusion. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1–10, 2025.

Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *European conference on computer vision*, pp. 605–621, 2020.

Quanmin Wei, Penglin Dai, Wei Li, Bingyi Liu, and Xiao Wu. Infocom: kilobyte-scale communication-efficient collaborative perception with information bottleneck. In *AAAI Conference on Artificial Intelligence*, volume 40, pp. 29731–29739, 2026.

Jiageng Wen, Shengjie Zhao, Bing Li, Jiafeng Huang, Kenan Ye, and Hao Deng. Simo: Single-modality-operable multimodal collaborative perception. In *International Conference on Learning Representations*, 2026.

Qiming Xia, Wenkai Lin, Haoen Xiang, Xun Huang, Siheng Chen, Zhen Dong, Cheng Wang, and Chenglu Wen. Learning to detect objects from multi-agent lidar scans without manual labels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1418–1428, 2025.

Junhao Xu, Yanan Zhang, Zhi Cai, and Di Huang. Cosdh: communication-efficient collaborative perception via supply-demand awareness and intermediate-late hybridization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6834–6843, 2025a.

Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *European conference on computer vision*, pp. 107–124, 2022a.

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 *International Conference on Robotics and Automation*, pp. 2583–2589, 2022b.

Yunjiang Xu, Lingzhi Li, Jin Wang, Yupeng Ouyang, and Benyuan Yang. Instinct: Instance-level interaction architecture for query-based collaborative perception. In *IEEE/CVF International Conference on Computer Vision*, pp. 25464–25473, 2025b.

Dingkang Yang, Kun Yang, Yuzheng Wang, Jing Liu, Zhi Xu, Rongbin Yin, Peng Zhai, and Lihua Zhang. How2comm: Communication-efficient and collaboration-pragmatic multi-agent perception. *Advances in Neural Information Processing Systems*, 36:25151–25164, 2023a.

Kun Yang, Dingkang Yang, Jingyu Zhang, Hanqi Wang, Peng Sun, and Liang Song. What2comm: Towards communication-efficient collaborative perception via feature decoupling. In *ACM international conference on multimedia*, pp. 7686–7695, 2023b.

Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, et al. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *IEEE/CVF conference on computer vision and pattern recognition*, pp. 21361–21370, 2022.

Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, and Xing Wei. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving. *Advances in Neural Information Processing Systems*, 38:67299–67318, 2026.

Jingyu Zhang, Kun Yang, Yilei Wang, Hanqi Wang, Peng Sun, and Liang Song. Ermvp: Communication-efficient and collaboration-robust multi-vehicle perception in challenging environments. In *IEEE/CVF conference on computer vision and pattern recognition*, pp. 12575–12584, 2024.

Yiyuan Zhang, Xiaohan Ding, and Xiangyu Yue. Scaling up your kernels: Large kernel design in convnets towards universal representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):11692–11707, 2025.

Junfei Zhou, Penglin Dai, Quanmin Wei, Bingyi Liu, Xiao Wu, and Jianping Wang. Pragmatic heterogeneous collaborative perception via generative communication mechanism. *Advances in Neural Information Processing Systems*, 38:54484–54509, 2026.

A Appendix

A.1 Overall Algorithm of CoDelta

Algorithm 1 summarizes the inference pipeline of CoDelta. Lines 1–12 correspond to *Priority-Aware Packetization*, where each collaborator partitions its BEV feature map into local patches and ranks them by predicted task relevance. It then packages the selected patches' encoded payloads and spatial indices into packets ordered by priority. Line 13 applies the partial-reception model by retaining the first $\lceil rK \rceil$ packets of each sequence. Lines 14–20 correspond to *Ego-Centric Delta Fusion*, where each received packet is converted into a local Delta update on the ego BEV, and the aggregated representation is refined to produce $\mathbf{F}_r$.

Algorithm 1 Overall Pipeline of CoDelta

Require: Ego feature $\mathbf{F}_{\text{ego}}$, aligned collaborator features $\{\mathbf{F}_i\}_{i=1}^{N}$, reception ratio r, patch size (H_p, W_p), packet budget K
Ensure: Collaborative representation $\mathbf{F}_r$
1: **for** each collaborator $i = 1, \ldots, N$ **do**
2: Partition $\mathbf{F}_i$ into a regular patch grid with size (H_p, W_p)
3: Predict the full-resolution saliency map $\mathbf{S}_i = \sigma\big(f_{\text{sal}}(\mathbf{F}_i)\big)$
4: Compute patch priority scores $q_i^{u,v}$ using mean–max aggregation
5: Select up to K patches with the highest priority scores
6: **for** each selected patch (u, v) **do**
7: Extract and zero-pad the local feature $\mathbf{z}_i^{u,v}$
8: Generate the payload $\mathbf{c}_i^{u,v} = f_{\text{pay}}(\mathbf{z}_i^{u,v})$
9: Form the packet $p_i^{u,v} = (\ell_i^{u,v}, \mathbf{c}_i^{u,v})$
10: **end for**
11: Sort the selected packets by descending score into the transmission sequence $\mathcal{P}_i$
12: **end for**
13: Retain the prefix $\mathcal{P}_i^{\le r}$ from each packet sequence
14: Initialize $\mathbf{F}'_r \leftarrow \mathbf{F}_{\text{ego}}$
15: **for** each received packet $p_i^{u,v} = (\ell_i^{u,v}, \mathbf{c}_i^{u,v})$ in $\{\mathcal{P}_i^{\le r}\}_{i=1}^{N}$ **do**
16: Crop the corresponding ego patch $\mathbf{e}^{u,v}$ using $\ell_i^{u,v}$
17: Generate the local Delta update $\Delta\mathbf{e}_i^{u,v} = f_{\Delta}(\mathbf{e}^{u,v}, \mathbf{c}_i^{u,v})$
18: Accumulate $\Delta\mathbf{e}_i^{u,v}$ into the corresponding patch of $\mathbf{F}'_r$ at location $\ell_i^{u,v}$
19: **end for**
20: Apply Context-Guided Refinement to obtain $\mathbf{F}_r = f_{\text{ref}}(\mathbf{F}'_r)$
21: **return** $\mathbf{F}_r$

A.2 Architectural Details of CoDelta

This section details the sender-side heads and receiver-side modules of CoDelta. As illustrated in Fig. 8, the saliency and payload heads follow a common convolutional design with different depths. The saliency head f_{sal} contains two blocks, each comprising a 3×3 convolution, GroupNorm, and ReLU, followed by a 1×1 projection to one channel. It produces full-resolution saliency logits, which are passed through a sigmoid function to obtain the saliency map used for patch ranking. The payload head f_{pay} uses one such block followed by a 1×1 projection, mapping each patch feature to a single-channel payload of size $H_p \times W_p$.

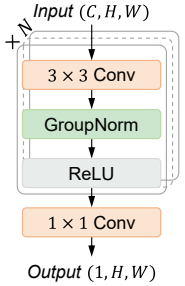


Figure 8: CoDelta sender-side saliency and payload heads.

On the receiver side, the Delta Predictor and Context-Guided Refinement module both use multi-scale convolutional modulation (Guo et al., 2022; Zhang et al., 2025). Each module consists of two stacked basic blocks, whose structures are shown in Fig. 9. In the Delta Predictor block, the ego patch is processed by GroupNorm, a 1×1 convolution, and GELU. Parallel depthwise convolutional branches capture local context

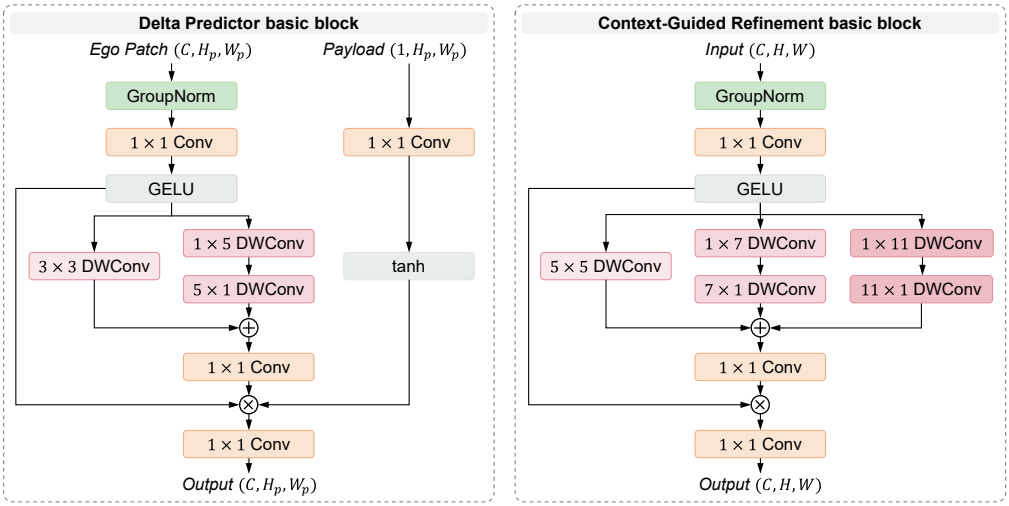


Figure 9: Architectures of the basic blocks used in the Delta Predictor (left) and Context-Guided Refinement module (right). Both use multi-scale convolutional modulation. The Delta Predictor block combines a received payload with the corresponding ego patch to generate a local additive update. The refinement block uses larger spatial kernels to model contextual interactions across patches in the aggregated ego BEV representation.

using a 3×3 kernel and a sequence of 1×5 and 5×1 kernels. The received payload is separately transformed by a 1×1 convolution and a $\tanh$ activation to produce a modulation signal. This signal modulates the ego features together with the multi-scale branch output, followed by a 1×1 projection. The Delta Predictor produces a feature residual with the same dimensions as the ego patch, which is added to the corresponding region during aggregation.

The Context-Guided Refinement module processes the aggregated ego BEV representation. Its basic block uses a 5×5 depthwise convolutional branch and two additional branches with factorized 7×7 and 11×11 kernels. These larger spatial kernels model interactions across patch boundaries and with surrounding features. The combined multi-scale response modulates the input features before a final 1×1 projection, preserving the dimensions of the ego BEV representation.

A.3 Further Discussion of CoDelta

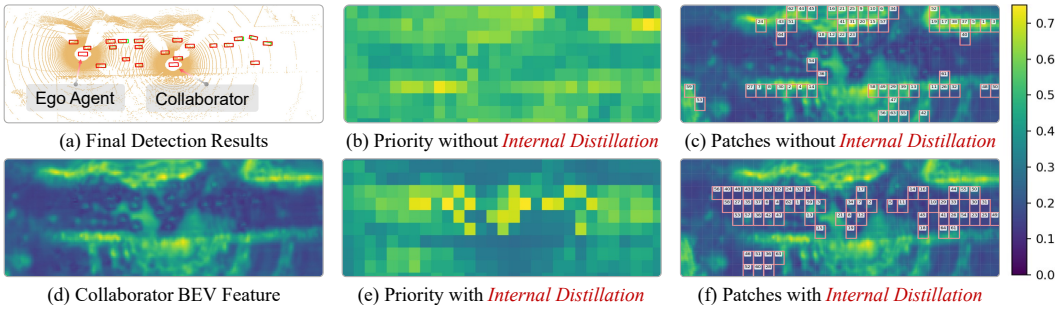
This section discusses CoDelta's communication model and its main limitations.

A.3.1 Communication Modeling Scope

CoDelta models partial reception at the level of logical packets, each containing a compact BEV patch payload and its spatial index. A logical packet defines a unit of information used by the perception model and does not prescribe a network packet format. We assume that the packets available by the perception deadline form a prefix of each collaborator's priority-ordered message. The reception ratio r specifies the available fraction of that message. This abstraction represents incomplete information availability without explicitly modeling the underlying bandwidth, channel dynamics, or packet delivery process. CoDelta focuses on organizing collaborative content and using the received prefixes for perception. Retransmission, scheduling, and channel access control lie outside the evaluated model; studying their interaction with CoDelta remains future work.

A.3.2 Limitations of CoDelta

Sender-side priority estimation. CoDelta uses sender-side foreground saliency as a task-relevant signal for patch prioritization, enabling priority estimation without additional communication rounds. The current scoring mechanism does not explicitly account for the ego agent's information needs. Incorporating ego-specific relevance while retaining this communication efficiency is a direction for future work.



(a) Final Detection Results (b) Priority without *Internal Distillation* (c) Patches without *Internal Distillation*

(d) Collaborator BEV Feature (e) Priority with *Internal Distillation* (f) Patches with *Internal Distillation*

Figure 10: Effect of internal distillation on patch prioritization. (a) Detection results. (d) Collaborator BEV features. (b, c) Priority scores and selected patches without distillation. (e, f) Corresponding results with distillation, showing greater concentration around object regions.

Validation under real-world communication conditions. Our experiments evaluate partial reception through controlled message-prefix truncation, including on real-world perception data. Evaluating CoDelta with measured packet arrivals and perception deadlines in a V2X testbed remains future work.

A.4 Analysis of Design Choices

This section discusses the rationale behind CoDelta's communication granularity, priority estimation, and ego-centric updates, with supporting ablations and visualizations.

A.4.1 Patch-Based Communication

CoDelta uses patches as local communication units to support selective transmission and direct ego updates under partial reception. Treating an entire BEV feature map as a single logical unit provides no explicit spatial granularity for prioritization, whereas selecting individual BEV locations offers finer control over the transmitted content. Patches provide an intermediate granularity: each local region can be independently selected, encoded, and associated with the corresponding ego region.

Patch-based encoding also distributes spatial indexing overhead across multiple feature locations. Each payload requires only one patch index, while individually indexed BEV locations would each incur a separate positional cost. This distinction is particularly relevant for compact single-channel payloads, where an index can occupy as many bytes as an individual feature value or more. Patch-based communication thus combines local selection with shared positional metadata, providing a practical unit for generating ego updates from incomplete messages.

A.4.2 Ego-Centric Updates versus Peer BEV Fusion

Peer BEV fusion treats collaborator features as separate representations that are aligned and fused with ego features. CoDelta instead uses collaborator information to update the ego representation directly. Each received payload is combined with the corresponding ego patch to predict a local additive adjustment, expressing the collaborator's contribution as a change to the ego features. This distinction is particularly relevant under partial reception, when only part of a collaborator's message is available. CoDelta converts the received information into local updates without first constructing a complete collaborator BEV representation. By focusing on how available information can improve the existing ego representation, this design makes incomplete message prefixes directly usable for collaborative perception.

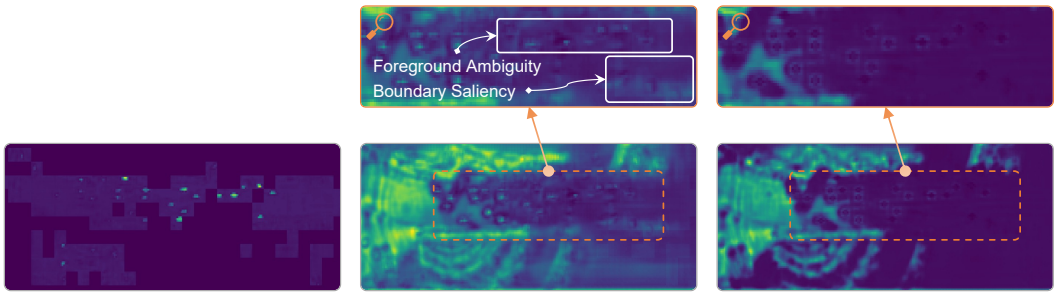
A.4.3 Internal Distillation for Priority Learning

The final detection objective does not explicitly specify which collaborator patches should receive higher transmission priority. CoDelta therefore supervises the saliency head through internal distillation using foreground responses from the detection head as soft targets. These responses provide a detection-relevant signal for patch prioritization. A stop-gradient operation prevents the distillation loss from backpropagating through the teacher branch.



(a) Priority with Average Aggregation (b) Priority with Max Aggregation (c) Priority with *Hybrid Aggregation*

Figure 11: Comparison of patch priority maps using (a) mean aggregation, (b) max aggregation, and (c) hybrid mean–max aggregation. The combined score incorporates both average saliency and local peak responses.



(a) Ego-Centric Incremental Delta (b) BEV Feature before *CG Refinement* (c) BEV Feature after *CG Refinement*

Figure 12: Effect of Context-Guided Refinement on the ego BEV representation. (a) Aggregated local Delta updates. (b) BEV features before refinement. (c) BEV features after refinement. Enlarged views show less pronounced patch boundaries within the highlighted region after refinement.

Fig. 10 visualizes the effect of internal distillation on priority prediction and patch selection. Without distillation, elevated scores and selected patches are dispersed across the feature map. Internal distillation concentrates higher scores around object regions and directs patch selection toward these areas. This spatial correspondence supports the role of foreground supervision in prioritizing detection-relevant content when only a short message prefix is available.

A.4.4 Mean–Max Saliency Aggregation

CoDelta converts the full-resolution saliency map into patch-level priority scores. Mean aggregation captures the average response within a patch but can dilute strong responses confined to a small area. Max aggregation preserves the strongest response without considering the remaining locations. Combining these statistics allows patch priorities to reflect both average saliency and local peaks.

Fig. 11 visualizes the priority maps produced by the three aggregation strategies. Mean-only aggregation yields broader regions of elevated scores, while max-only aggregation emphasizes localized peaks. Hybrid mean–max aggregation incorporates both patterns when assigning patch priorities. These observations complement the ablation results in Table 3, where the combined score achieves higher AP@70 than either statistic alone.

A.4.5 Refinement after Delta Aggregation

CoDelta computes local Delta updates from the same ego BEV and adds them at their corresponding spatial locations. Each received packet can thus contribute independently, although the aggregation itself does not explicitly model interactions across patches. Context-Guided Refinement uses larger spatial kernels to integrate updated regions with their surrounding context before detection.

Fig. 12 shows the aggregated Delta updates and the ego BEV features before and after refinement. The updates in Fig. 12(a) are spatially sparse. Patch boundaries visible in the aggregated features in Fig. 12(b) become less pronounced after refinement in Fig. 12(c). These visual changes illustrate the effect of contextual processing on the aggregated representation. The ablation results in Table 3 further demonstrate its contribution to detection accuracy, particularly at the stricter IoU threshold.

A.5 Supplementary Experimental Details

This section provides additional dataset information, definitions of the ablation variants, and hardware and software configurations to support reproducibility.

A.5.1 Additional Dataset Details

Our evaluation covers simulated vehicle-to-vehicle and vehicle-to-everything collaboration, as well as real-world vehicle-to-infrastructure collaboration.

OPV2V. OPV2V (Xu et al., 2022b) is a simulated vehicle-to-vehicle collaborative perception dataset collected using CARLA and OpenCDA. It contains 11,464 frames and 232,913 annotated 3D vehicle bounding boxes across diverse driving scenes.

V2XSet. V2XSet (Xu et al., 2022a) is a simulated vehicle-to-everything collaborative perception dataset with 58 scenarios. It includes both connected vehicles and roadside infrastructure, providing observations from multiple types of collaborating agents.

DAIR-V2X. DAIR-V2X (Yu et al., 2022) provides real-world vehicle-to-infrastructure perception data collected from vehicle-mounted and roadside sensors. Following Lu et al. (2023), we use the complemented annotations for its cooperative subset. These annotations supplement objects outside the original vehicle-side camera field of view to support 360-degree detection.

A.5.2 Hardware and Software Configurations

For reproducibility and to contextualize the reported processing latency, we summarize the hardware and software environment used in our experiments below.

- **Hardware.** All experiments were conducted on a single machine running Ubuntu 22.04.1 LTS, equipped with an Intel(R) Xeon(R) Platinum 8347C CPU @ 2.10GHz, 32 GB system memory, and an NVIDIA RTX 4090 GPU.
- **Software.** The implementation was based on CUDA 13.0, Python 3.8.20, PyTorch 1.12.0, spconv 2.3.6, and OpenCOOD 0.1.0.

A.5.3 Definitions of Ablation Variants

Table 3 compares the following variants on OPV2V. All variants are evaluated at full reception ($r = 1$). Each ablation modifies the specified component of the full model.

- **Random ordering.** We replace the learned patch priority scores with random scores and select the top-K_i patches accordingly. At $r = 1$, this comparison measures the effect of priority-based patch selection.
- **Mean-only priority score.** We compute each patch score using only the average saliency over its valid spatial support: $q_i^{u,v} = |\Omega_i^{u,v}|^{-1} \sum_{(x,y)\in\Omega_i^{u,v}} \mathbf{S}_i(x, y)$. These scores replace the mean–max scores for patch selection and ordering.
- **Max-only priority score.** We compute each patch score using only the maximum saliency over its valid spatial support: $q_i^{u,v} = \max_{(x,y)\in\Omega_i^{u,v}} \mathbf{S}_i(x, y)$. These scores replace the mean–max scores for patch selection and ordering.
- **Without internal distillation.** We disable the distillation loss by setting $\lambda_{\text{distill}} = 0$. The saliency head architecture and the score-based selection rule are retained to assess the contribution of explicit foreground supervision.
- **Without refinement.** We remove Context-Guided Refinement and pass the aggregated ego representation directly to the detection head: $\mathbf{F}_r = \mathbf{F}'_r$. Delta prediction and aggregation are retained.
- **Without Delta regularization.** We set $\lambda_{\text{delta}} = 0$ to remove the penalty on local update magnitudes. Delta prediction and aggregation are retained.
- **Full model.** The complete CoDelta model uses equally weighted mean–max aggregation, internal distillation, ego-centric Delta updates, Context-Guided Refinement, and Delta regularization. The auxiliary loss weights are $\lambda_{\text{distill}} = 0.5$ and $\lambda_{\text{delta}} = 0.001$.

Table 6: Ablation results on OPV2V under partial reception. Best results are in **bold**.

Variant	$r = 0.1$		$r = 0.5$	
	AP@50	AP@70	AP@50	AP@70
Random ordering	0.8947	0.8046	0.9070	0.8224
Mean-only priority score	0.9142	0.8260	0.9469	0.8842
Max-only priority score	0.9425	0.8954	0.9702	0.9299
w/o internal distillation	0.8960	0.7934	0.9226	0.8394
w/o refinement	0.9356	0.8287	0.9456	0.8312
w/o Delta regularization	0.9449	0.8894	0.9689	0.9250
Full model	**0.9483**	**0.8961**	**0.9711**	**0.9314**

A.6 Communication Volume Calculation

We measure communication volume as the total size of the quantized payloads and spatial indices in a collaborator's message. Fixed packet headers and quantization parameters are excluded. Each patch location (u, v) is encoded as a single integer index in the $H_g \times W_g$ grid, requiring $\lceil \log_2(H_g W_g) \rceil$ bits. The payload and index are each stored in a whole number of bytes. With b bits per payload element, the complete message size is

$$B_i^{\text{full}} = K_i \left(\left\lceil \frac{H_p W_p b}{8} \right\rceil + \left\lceil \frac{\lceil \log_2(H_g W_g) \rceil}{8} \right\rceil \right) \quad \text{bytes}, \tag{7}$$

where K_i is the number of selected packets. The two terms represent the payload size and the patch index size, respectively. We report volumes using $1\,\text{KiB} = 1024$ bytes and $1\,\text{MiB} = 1024^2$ bytes.

Under the default settings of $H_p = W_p = 8$, $b = 8$, and $K_i = 128$, each payload occupies 64 bytes. OPV2V and V2XSet have a patch grid of $13 \times 44 = 572$ locations, requiring a 10-bit index stored in 2 bytes. Each packet therefore occupies 66 bytes, giving a complete message size of 8448 bytes, or 8.250 KiB, per collaborator. DAIR-V2X has $13 \times 32 = 416$ patch locations and requires a 9-bit index, which also occupies 2 bytes. Its default message size is therefore the same.

A.7 Ablations under Partial Reception

Table 6 evaluates the contribution of each component at $r = 0.1$ and $r = 0.5$. The full model achieves the highest AP@50 and AP@70 at both ratios. Random ordering reduces AP@70 by 9.15 and 10.90 percentage points, respectively, demonstrating the benefit of learned priorities when only part of the selected message is received. Mean–max aggregation outperforms mean-only aggregation at both ratios and provides a small improvement over max-only aggregation.

Without internal distillation, AP@70 falls to 0.7934 at $r = 0.1$ and 0.8394 at $r = 0.5$. These reductions support the use of foreground supervision to learn effective priorities for incomplete message prefixes. Context-Guided Refinement helps the model benefit from increased reception: AP@70 rises from 0.8961 to 0.9314 in the full model but remains nearly unchanged at 0.8287 and 0.8312 without refinement. This contrast suggests that contextual processing is important for translating additional received information into detection gains. Delta regularization provides a smaller benefit, with its removal reducing AP@70 by 0.0067 and 0.0064 at the two reception ratios.

A.8 Ethics Statement

This work aims to improve collaborative perception for applications such as autonomous driving. Our experiments use existing benchmark datasets and involve no new personal-data collection or human-subject studies. Although CoDelta improves detection under the evaluated reception conditions, benchmark performance does not establish its safety for deployment. Real-world use would require further validation across diverse traffic and communication conditions, with particular attention to perception failures caused by incomplete or unreliable collaborator information.