CGDP: COMPLEMENTARITY-GUIDED DUAL-PRETEXT LEARNING FOR COLLABORATIVE PERCEPTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Collaborative perception enables multiple agents to fuse complementary observations for comprehensive scene understanding, yet learning effective representations typically relies on expensive 3D bounding box annotations. Reconstruction-based pretraining offers a promising alternative by recovering masked geometry from unlabeled multi-agent point clouds, but existing methods treat all observations uniformly during masking and condition reconstruction solely on collaborative inputs, neglecting the varying contributions of different agents and providing no explicit mechanism to distinguish what can be recovered from ego-only perception. To bridge this gap, we propose a Complementarity-Guided Dual-Pretext (CGDP) learning framework that introduces two key designs, namely an ego-relative spatial masking strategy that preserves ego-points while selectively masking non-ego regions, and a dual reconstruction objective that combines complementarity-weighted masked collaborative reconstruction with ego-to-collaborative scene prediction. Both tasks are jointly optimized on a shared encoder-decoder architecture using targets constructed directly from unlabeled point clouds, and only the pretrained encoder is transferred to downstream detection without introducing any additional inference overhead. Extensive experiments on V2X-Real and OPV2V demonstrate that CGDP achieves state-of-the-art 3D detection performance, validating that explicitly modeling observation complementarity during pretraining substantially benefits collaborative perception. The code and model will be publicly available.

1 INTRODUCTION

Collaborative perception enables agents to jointly interpret a shared environment by exchanging observations, benefiting autonomous driving, robotics, and intelligent transportation through expanded sensing coverage, reduced occlusion, improved long-range recognition, and greater robustness to incomplete local observations (Xu et al., 2022a; Hu et al., 2022; Lu et al., 2024; Ding et al., 2025; Liu et al., 2026). However, annotating 3D objects across multiple agents, viewpoints, and synchronized frames is costly and complex, motivating self-supervised pretraining that derives prediction tasks from unlabeled multi-agent observations to learn transferable representations before downstream task training.

Recent single-agent pretraining methods attempt to explore from multiple perspectives, such as masked patch reconstruction (Pang et al., 2022), geometric prediction (Yan et al., 2024b), BEV-aware learning (Tian et al., 2023; Lin et al., 2024), multi-view modeling (Yan et al., 2024a), or image-to-3D knowledge transfer (Dong et al., 2023); however, they mainly model individual observations or auxiliary views and overlook geometry newly supplied by collaborators. Cooperative pretraining methods such as CORE (Wang et al., 2023), CooPre (Zhao et al., 2025), and TurboTrain (Zhou et al., 2025) introduce reconstruction-based multi-agent pretraining, yet do not explicitly prioritize collaborator-complementary regions or couple collaborative reconstruction with ego-only reconstruction of the same jointly observed scene.

As illustrated in Fig. 1, single-agent pretraining overlooks collaborator-added geometry, while conventional fused multi-agent pretraining does not explicitly prioritize such complementary regions.

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.

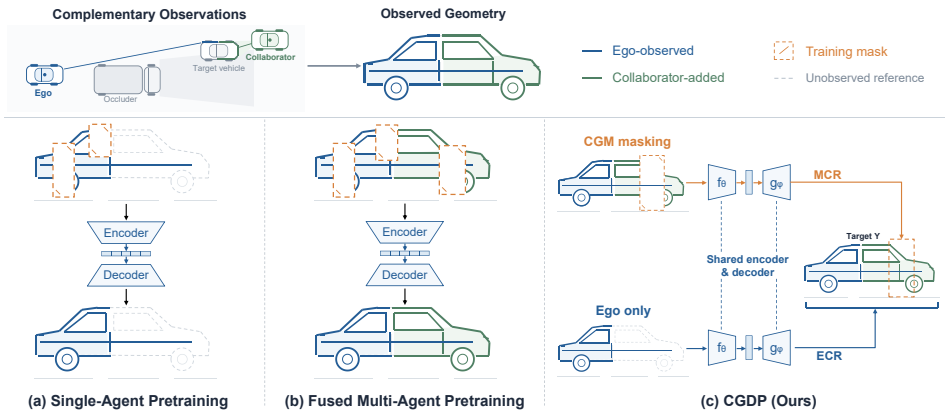

Figure 1: Comparison of collaborative pretraining paradigms under complementary multi-agent observations. (a) Single-agent pretraining. (b) Fused multi-agent pretraining. (c) CGDP with complementarity-guided dual-pretext learning.

These limitations motivate two key insights: (1) the masking distribution should emphasize regions with weak ego support but strong collaborator support, and (2) the encoder should learn under both collaborative and ego-only observations. A common target constructed from aligned multi-agent observations provides unified supervision for both conditions.

Based on these ideas, we propose a Complementarity-Guided Dual-Pretext (CGDP) framework for collaborative perception pretraining. Specifically, Complementarity-Guided Masking (CGM) estimates relative geometric support from occupied-voxel statistics and selects regions where collaborators provide additional evidence. Masked Collaborative Reconstruction (MCR) reconstructs the jointly observed geometry from partially collaborative input, while Ego-Conditioned Scene Reconstruction (ECR) reconstructs the same target from ego-only input. The two tasks share an encoder and decoder and are jointly optimized before the decoder is discarded for downstream transfer. Extensive experiments on V2X-Real and OPV2V demonstrate that CGDP achieves SOTA 3D detection performance, validating that explicitly modeling observation complementarity during pretraining substantially benefits collaborative perception.

The main contributions of this paper are as follows:

- We propose a Complementarity-Guided Dual-Pretext (CGDP) framework for Self-supervised collaborative perception pretraining.
- We introduce Complementarity-Guided Masking (CGM), which uses relative ego–collaborator geometric support to select and weight reconstruction regions containing complementary scene information.
- We design a dual-pretext objective that combines Masked Collaborative Reconstruction (MCR) and Ego-Conditioned Scene Reconstruction (ECR), training a shared representation from partially collaborative and ego-only inputs.
- We conduct comprehensive experiments on V2X-Real and OPV2V across collaboration modes, fusion architectures, annotation budgets, distance ranges, and controlled ablations, demonstrating the effectiveness and transferability of the proposed framework.

2 Related Work

2.1 Collaborative Perception

Collaborative perception combines observations from multiple agents to enlarge spatial coverage and alleviate occlusion. Existing methods can typically be categorized into early, intermediate, and late collaboration according to the stage of information exchange. Early collaboration (Chen et al., 2019; Arnold et al., 2020; Yu et al., 2022) shares detailed raw measurements but incurs high com-

munication cost, late collaboration (Li et al., 2022; Xu et al., 2022c) exchanges compact predictions with limited geometric information, and intermediate collaboration (Guo et al., 2021; Xu et al., 2023; 2025) exchanges learned features as a compromise. However, their performance depends on representation quality, while supervised collaborative detection requires costly annotations across agents, scenes, and viewpoints.

2.2 Self-Supervised Point Cloud Pretraining

Self-supervised point cloud pretraining learns geometric representations from unlabeled observations and reduces dependence on manual annotations. Point-MAE (Pang et al., 2022) reconstructs masked point patches, MaskFeat3D (Yan et al., 2024b) predicts intrinsic surface features, GeoMAE (Tian et al., 2023) combines centroid, occupancy, normal, and curvature prediction, and BEV-MAE (Lin et al., 2024) uses BEV-guided masking and point-density prediction for outdoor scenes. MVNet (Yan et al., 2024a) exploits multiple views, while ACT (Dong et al., 2023) uses cross-modal teachers for 3D representation learning. These methods mainly target individual point clouds or auxiliary views and do not distinguish ego geometry from collaborator-provided evidence. They therefore cannot directly model the asymmetric and complementary observations required by multi-agent perception.

2.3 Collaborative Reconstruction and Pretraining

Collaborative reconstruction uses jointly observed geometry as supervision for regions poorly covered by one agent. CORE (Wang et al., 2023) incorporates reconstruction into collaborative perception, CooPre (Zhao et al., 2025) reconstructs masked regions of fused multi-agent point clouds, and TurboTrain (Zhou et al., 2025) extends this idea to temporal point-cloud and occupancy prediction. However, they sample masks over non-empty BEV regions without explicitly considering ego-collaborator complementarity. Their reconstruction budgets may therefore focus on redundant regions rather than collaborator-supported regions that are weakly observed by the ego agent. They also do not explicitly couple partial-collaboration reconstruction with ego-only reconstruction of the same jointly observed scene. In this paper, we propose a complementarity-guided dual-pretext framework to address these limitations.

3 Methodology

3.1 Framework Overview

As illustrated in Fig. 2, we propose a Complementarity-Guided Dual-Pretext (CGDP) framework for collaborative 3D perception. It aims to exploit geometry beyond ego coverage and learn scene representations from collaborative and local observations. The CGDP consists of Complementarity-Guided Masking (CGM), Masked Collaborative Reconstruction (MCR), and Ego-Conditioned Scene Reconstruction (ECR). CGM selects complementary regions, MCR reconstructs their geometry from the remaining collaborative context, and ECR recovers the jointly observed scene from ego-only input.

Specifically, given N agents with point clouds $\mathcal{P}_i$, where $i = 0$ denotes the ego agent, we align and aggregate their observations in the ego frame to construct a common reconstruction target. During pretraining, CGM uses agent-specific voxel occupancy to select spatial blocks and removes collaborator points within them while preserving all ego points. The masked collaborative and ego-only inputs are processed by the same 3D encoder and reconstruction decoder. MCR applies complementarity-weighted supervision within masked blocks, while ECR supervises all occupied target cells, jointly optimizing the shared parameters. After pretraining, the decoder is discarded and the encoder is fine-tuned for downstream detection, where per-agent BEV features are aggregated by a fusion module and processed by a detection head.

3.2 Complementarity-Guided Masking

To focus pretraining on regions where collaborators supplement limited ego observations, we introduce CGM. It measures ego-relative observation complementarity and uses the resulting scores

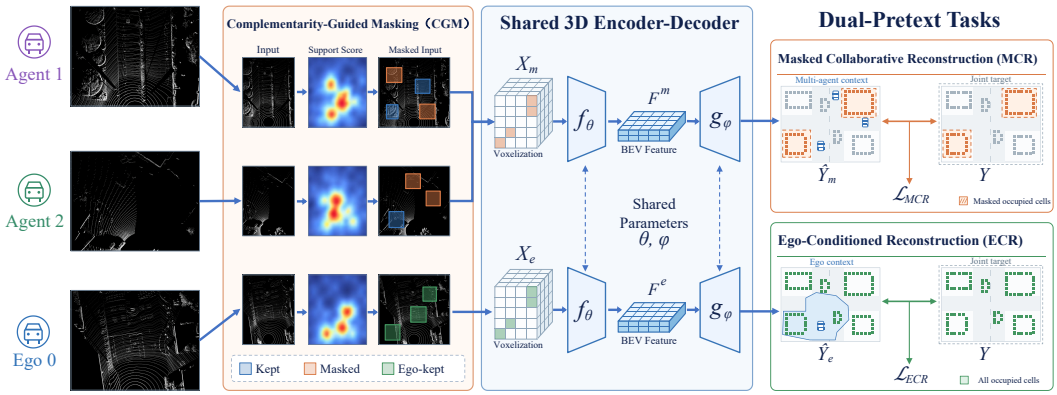

Figure 2: Overview of the CGDP. Complementarity-guided masking and reconstruction from both collaborative and ego-only observations train a shared encoder and decoder against a common scene target, yielding pretrained representations for collaborative 3D detection. Masking and reconstruction are used only during pretraining, and only the encoder is transferred to the downstream detector.

to guide BEV block selection. As shown in Fig. 3, collaborator points within selected blocks are removed, while all ego points and part of the collaborative context are retained.

Let $\mathcal{P}_i$ denote the aligned, jointly augmented, and cropped point cloud of agent i, with $i = 0$ denoting the ego agent. For each block $b \in \mathcal{B}$, let $n_i(b)$ be the number of distinct occupied fine voxels for agent i. We define its normalized support and complementarity as:

$$v_i(b) = \text{clip}\left(\frac{\log(1+n_i(b))}{\log(1+n_{\text{cap}})}, 0, 1\right), \qquad s_i(b) = (1 - v_0(b))v_i(b). \tag{1}$$

The logarithmic normalization reduces the influence of locally dense sampling. We aggregate collaborator contributions by $s(b) = \max_{1 \le i < N} s_i(b)$ and define the candidate set as $\mathcal{C} = \{b \in \mathcal{B} : s(b) > \tau\}$. Taking the maximum avoids accumulating overlapping support from multiple collaborators. We sample masked blocks with weights proportional to $s(b)$ under constraints that retain local collaborative context. The selected set $\mathcal{M}$ determines the collaborator points removed from the MCR input. Geometric configurations and the complete sampling procedure are provided in Appendices A.2 and B.1.

3.3 Masked Collaborative Reconstruction

To learn geometric representations from complementary observations without detection annotations, we introduce MCR. It reconstructs jointly observed geometry in selected regions from preserved ego observations and the remaining collaborative context, with reconstruction errors weighted by complementarity. Let $b(p)$ denote the block containing point p. The masked input and common reconstruction target are

$$X_{\text{m}} = \mathcal{P}_0 \cup \bigcup_{i=1}^{N-1} \{p \in \mathcal{P}_i : b(p) \notin \mathcal{M}\}, \qquad Y = \mathcal{T}\left(\bigcup_{i=0}^{N-1} \mathcal{P}_i\right). \tag{2}$$

Here, $\cup$ denotes point concatenation, and $\mathcal{T}$ constructs cell-wise point-set targets from the unmasked observations. Let Y_c contain the valid normalized target points in cell c and $\mathcal{O} = \{c : |Y_c| > 0\}$. A shared encoder f_θ and decoder g_ϕ predict:

$$\widehat{Y}^{\text{m}} = g_\phi\left(f_\theta(\mathcal{V}(X_{\text{m}}))\right), \tag{3}$$

where $\mathcal{V}$ denotes fine voxelization. Network configurations and target construction are detailed in Appendices A.2 and B.2.

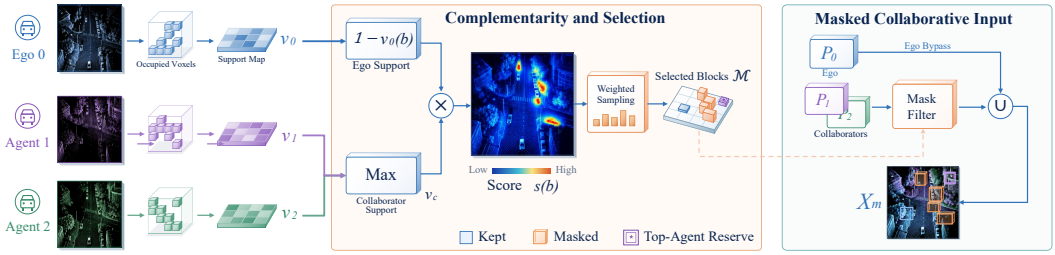

Figure 3: Complementarity-guided masking (CGM). Collaborator points are masked in selected complementary regions while all ego points are preserved.

For a non-empty mask set, let $\mathcal{O}_b$ contain the occupied target cells in block b. We average reconstruction errors within each block and assign complementarity-based weights:

$$\ell_b = \frac{1}{\max(1, |\mathcal{O}_b|)} \sum_{c \in \mathcal{O}_b} \mathrm{CD}_1(\widehat{Y}_c^{\mathrm{m}}, Y_c), \qquad w_b = \frac{s(b)}{\sum_{b' \in \mathcal{M}} s(b')}. \tag{4}$$

The per-sample MCR objective is

$$\ell_{\mathrm{MCR}} = \sum_{b \in \mathcal{M}} w_b \ell_b. \tag{5}$$

Here, CD_1 denotes the symmetric L_1 Chamfer distance. Block-wise averaging controls the contribution of varying target-cell counts, while complementarity weighting emphasizes geometry supported by collaborators. The distance definition and boundary cases are provided in Appendix B.3.

3.4 Ego-Conditioned Scene Reconstruction

To prepare the encoder for extracting per-agent features before downstream fusion, we introduce ECR. It reconstructs the jointly observed scene from the unmasked ego point cloud alone, allowing collaborator observations to provide geometric supervision for representation learning. ECR shares the encoder, decoder, and reconstruction target with MCR:

$$X_{\mathrm{e}} = \mathcal{P}_0, \qquad \widehat{Y}^{\mathrm{e}} = g_\phi\left(f_\theta(\mathcal{V}(X_{\mathrm{e}}))\right). \tag{6}$$

Supervision covers all occupied target cells, yielding the per-sample objective

$$\ell_{\mathrm{ECR}} = \frac{1}{\max(1, |\mathcal{O}|)} \sum_{c \in \mathcal{O}} \mathrm{CD}_1(\widehat{Y}_c^{\mathrm{e}}, Y_c). \tag{7}$$

Collaborator observations contribute to the reconstruction target without entering this branch as input. Together, MCR and ECR couple reconstruction under partial collaboration and ego-only observation through a shared prediction mapping.

3.5 Training

To transfer the geometric representations learned through self-supervision to downstream detection, we adopt a two-stage training strategy. We first pretrain a shared encoder and decoder by jointly optimizing the batch-averaged MCR and ECR objectives without detection annotations:

$$\mathcal{L}_{\mathrm{pre}} = \lambda_m \mathcal{L}_{\mathrm{MCR}} + \lambda_e \mathcal{L}_{\mathrm{ECR}}, \qquad \lambda_m = \lambda_e = 1. \tag{8}$$

After pretraining, we discard the reconstruction decoder and jointly fine-tune the encoder, fusion module, and detection head using detection annotations. CGM and reconstruction are used only during pretraining and introduce no additional inference modules. Loss aggregation and training settings are detailed in Appendices B.3 and A.3, respectively.

Table 1: Quantitative comparison on V2X-Real dataset.

Dataset	Method	Car		Pedestrian		Truck		Mean	
		AP0.3	AP0.5	AP0.3	AP0.5	AP0.3	AP0.5	AP0.3	AP0.5
V2X-Real VC	AttFuse	67.7	65.8	45.0	27.6	55.5	52.0	56.1	48.5
	F-Cooper	64.9	62.7	45.5	26.7	39.8	35.1	50.1	41.5
	V2X-ViT	63.3	61.2	42.2	24.8	38.4	36.0	48.0	40.0
	CooPre (AttFuse)	71.5	70.2	46.9	28.0	61.9	58.3	60.1	52.2
	CooPre (F-Cooper)	67.0	65.7	43.6	25.4	49.3	44.9	53.3	45.3
	CooPre (V2X-ViT)	66.1	64.5	45.3	27.7	43.5	40.1	51.7	44.1
	Ours (AttFuse)	**76.3** +8.6	**75.3** +9.5	**50.2** +5.2	**30.7** +3.1	**61.1** +5.6	**58.9** +6.9	**62.6** +6.5	**54.9** +6.4
	Ours (F-Cooper)	**71.0** +6.1	**69.6** +6.9	**45.7** +0.2	**27.3** +0.6	**50.8** +11	**48.4** +13.3	**55.8** +5.7	**48.4** +6.9
	Ours (V2X-ViT)	**70.9** +7.6	**69.1** +7.9	**46.5** +4.3	**27.8** +3.0	**43.7** +5.3	**39.8** +3.8	**53.7** +5.7	**45.5** +5.5
V2X-Real V2V	AttFuse	66.9	65.3	46.0	30.4	53.8	50.2	55.5	48.6
	F-Cooper	64.6	63.0	41.9	26.9	48.9	44.8	51.8	44.9
	V2X-ViT	62.6	58.7	47.5	30.5	49.3	47.0	53.1	45.4
	CooPre (AttFuse)	71.4	70.1	49.7	33.7	57.5	55.5	59.5	53.1
	CooPre (F-Cooper)	68.6	67.0	42.0	26.4	51.0	48.1	53.9	47.2
	CooPre (V2X-ViT)	66.9	64.5	46.7	29.2	54.1	52.3	55.9	48.7
	Ours (AttFuse)	**75.8** +8.9	**73.8** +8.5	**51.9** +5.9	**35.0** +4.6	**54.6** +0.8	**52.4** +2.2	**60.7** +5.2	**53.7** +5.1
	Ours (F-Cooper)	**72.1** +7.5	**70.0** +7.0	**43.5** +1.6	**27.2** +0.3	**52.2** +3.3	**48.1** +3.3	**55.9** +4.1	**48.4** +3.5
	Ours (V2X-ViT)	**69.4** +6.8	**67.2** +8.5	**47.7** +0.2	30.3 -0.2	**54.5** +5.2	**52.5** +5.5	**57.2** +4.1	**50.0** +4.6
V2X-Real IC	AttFuse	84.5	82.1	61.0	40.6	59.9	59.1	68.5	60.6
	F-Cooper	79.1	76.2	57.0	35.8	52.0	45.7	62.7	52.6
	V2X-ViT	81.0	76.9	44.9	29.2	37.3	36.8	54.4	47.6
	CooPre (AttFuse)	86.2	84.1	60.5	39.3	61.5	61.0	69.4	61.4
	CooPre (F-Cooper)	84.1	80.7	58.4	33.9	51.8	50.7	64.8	55.1
	CooPre (V2X-ViT)	81.4	77.3	45.8	29.1	37.4	37.2	54.8	47.9
	Ours (AttFuse)	**88.3** +3.8	**83.6** +1.5	**61.9** +0.9	**40.8** +0.2	**61.0** +1.1	**60.7** +1.6	**70.4** +1.9	**61.7** +1.1
	Ours (F-Cooper)	**84.5** +5.4	**83.0** +6.8	**60.9** +3.9	**38.6** +2.8	**52.3** +0.3	**50.3** +4.6	**65.9** +3.2	**57.3** +4.7
	Ours (V2X-ViT)	**82.6** +1.6	**81.0** +4.1	**47.1** +2.2	28.9 -0.3	**38.8** +1.5	**37.6** +0.8	**56.1** +1.7	**49.1** +1.5
V2X-Real I2I	AttFuse	83.4	81.9	67.0	44.8	63.6	62.8	71.3	63.2
	F-Cooper	85.1	82.0	66.1	44.3	51.3	50.8	67.5	59.0
	V2X-ViT	82.0	79.5	68.9	45.4	59.0	57.0	69.9	60.6
	CooPre (AttFuse)	84.6	82.7	66.5	45.2	65.2	62.5	72.1	63.5
	CooPre (F-Cooper)	85.4	82.1	66.6	45.1	51.9	51.1	68.0	59.4
	CooPre (V2X-ViT)	84.0	82.4	66.9	44.4	59.1	58.7	70.0	61.9
	Ours (AttFuse)	**87.7** +4.3	**84.8** +2.9	**69.6** +2.6	**48.0** +3.2	**65.4** +1.8	61.9 -0.9	**74.2** +2.9	**64.9** +1.7
	Ours (F-Cooper)	**86.5** +1.4	**84.3** +2.3	**67.9** +1.8	**45.4** +1.1	**52.1** +0.8	**51.4** +0.6	**68.8** +1.3	**60.3** +1.3
	Ours (V2X-ViT)	**86.8** +4.8	**84.0** +4.5	**69.1** +0.2	**45.5** +0.1	**62.4** +3.4	**60.4** +3.4	**72.7** +2.8	**63.3** +2.7

4 EXPERIMENTS

4.1 EXPERIMENT SETTING

Datasets. We evaluate our framework on V2X-Real (Xiang et al., 2025) and OPV2V (Xu et al., 2022b). On V2X-Real, we consider vehicle-centric (VC), infrastructure-centric (IC), vehicle-to-vehicle (V2V), and infrastructure-to-infrastructure (I2I) collaboration. On OPV2V, we evaluate on the Default and Culver City test sets. Pretraining uses only training point clouds without detection annotations. Dataset configurations and evaluation regions are provided in Appendix A.1.

Evaluation metrics. We measure detection performance using AP based on BEV IoU. For V2X-Real, we report AP at IoU thresholds of 0.3 and 0.5 for cars, pedestrians, and trucks, together with mAP over the three categories. For OPV2V, we report vehicle AP at IoU thresholds of 0.5 and 0.7. Green denotes the improvements over the corresponding base detector for all experiments.

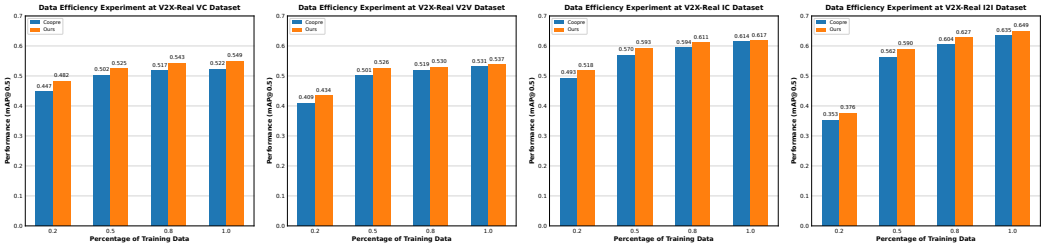

Figure 4: Data efficiency experiment. We evaluate the performance gains of our method over CooPre under different ratios of labeled fine-tuning data on V2X-Real with AttFuse.

Table 2: Quantitative comparison on OPV2V.

Method	Default		Culver	
	$AP_{0.5}$	$AP_{0.7}$	$AP_{0.5}$	$AP_{0.7}$
No Fusion	71.3	60.4	64.6	51.7
Early Fusion	87.7	81.3	82.1	73.8
Late Fusion	84.6	77.5	80.8	68.2
AttFuse	89.3	82.6	87.5	76.0
F-Cooper	87.8	81.7	89.0	79.7
V2X-ViT	89.8	83.7	88.7	80.2
CooPre (AttFuse)	91.7	86.6	88.7	80.0
CooPre (F-Cooper)	89.0	83.4	90.2	82.0
CooPre (V2X-ViT)	91.9	86.5	89.9	82.4
Ours (AttFuse)	**94.1** +4.8	**90.3** +7.7	**91.2** +3.7	**84.8** +8.8
Ours (F-Cooper)	**92.3** +4.5	**89.0** +7.3	**90.8** +1.8	**83.7** +4.0
Ours (V2X-ViT)	**93.2** +3.4	**88.8** +5.1	**90.5** +1.8	**83.8** +3.6

Table 3: Comparison with prior works.

Dataset	Pretraining Method	Overall	0–30m	30–50m	50–100m
V2X-Real VC	No Pretrain	48.5	61.0	53.5	36.9
	BEVContrast	48.8	61.2	52.5	39.2
	Random Masking	50.9	63.9	55.4	40.3
	CORE	51.6	64.7	55.4	40.7
	STAR	50.0	63.9	55.6	38.4
	BEV-MAE	51.1	64.5	55.7	39.4
	CooPre	52.2	64.9	55.9	42.1
	Ours	**54.9**	**67.2**	**57.4**	**44.1**
V2X-Real V2V	No Pretrain	48.6	64.2	52.3	36.0
	BEVContrast	46.3	63.1	54.1	32.9
	Random Masking	50.9	65.5	54.7	40.4
	CORE	51.9	65.9	55.0	41.8
	STAR	51.2	66.0	54.4	40.6
	BEV-MAE	50.5	65.4	54.6	38.7
	CooPre	53.1	66.1	55.4	42.7
	Ours	**53.7**	**66.5**	**56.2**	**43.0**

Implementation details. Our framework is implemented in PyTorch based on OpenCOOD, using SECOND (Yan et al., 2018) as the encoder and AttFuse, F-Cooper, and V2X-ViT as downstream fusion modules. We jointly optimize MCR and ECR during pretraining and transfer the encoder to each downstream detector for fine-tuning. Detailed model configurations and training settings are provided in Appendices A.2 and A.3.

4.2 Quantitative Analysis

Main results. As shown in Table 1, our method consistently outperforms the corresponding baseline in overall mAP across all four collaboration modes on V2X-Real using AttFuse, F-Cooper, and V2X-ViT. Taking AttFuse as an example, our method improves car $AP_{0.5}$ by 9.5% and 8.5% over the corresponding baseline under the VC and V2V settings, respectively, while improving pedestrian $AP_{0.5}$ by 3.1% and 4.6%. Improvements in overall mAP are also observed under the IC and I2I settings. Although category-specific gains vary, with slight lower performance in a few individual cases, the consistent improvements in overall performance demonstrate the effectiveness of the proposed pretraining framework across different collaboration modes and fusion architectures.

As shown in Table 2, our method improves detection performance on both the Default and Culver City test sets of OPV2V. These results demonstrate that the benefits of the proposed pretraining framework extend to simulated V2V scenarios. Together with the results on V2X-Real, they support its applicability to collaborative perception in both real-world and simulated environments.

Comparison with other pretraining methods. We compare pretraining methods using SECOND as the 3D backbone and AttFuse as the fusion module. As shown in Table 3, our method outperforms the compared baselines on both VC and V2V, improving overall mAP@0.5 over CooPre by 2.7% and 0.6%, respectively. Consistent gains across all three distance ranges suggest that our approach learns more transferable representations for collaborative detection, with particularly pronounced benefits in the VC setting.

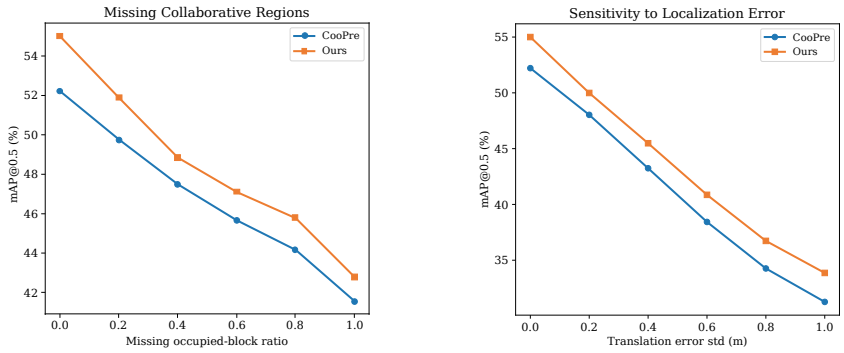

Figure 5: Robustness assessment. Comparison with CooPre on V2X-Real under the VC setting with AttFuse, evaluated at different missing occupied-block ratios (left) and planar translation error standard deviations (right).

Table 4: Components ablation of CGDP.

CGM	MCR	ECR	mAP@0.3	mAP@0.5
–	–	–	56.1	48.5
–	✓	–	57.8	50.6
–	–	✓	57.2	50.1
–	✓	✓	58.4	51.2
✓	✓	✓	**62.6**	**54.9**

Table 5: Mask-ratio ablation of CGM.

Dataset	Method	Ratio	mAP@0.3	mAP@0.5
V2X-Real VC	AttFuse	0.5	57.4	49.2
		0.6	60.3	52.1
		0.7	**62.6**	**54.9**
		0.8	59.8	51.4
		0.9	56.4	48.5

Data efficiency. We investigate the benefits of our pretraining method under limited labeled data on V2X-Real using AttFuse. We pretrain on the entire training set without detection annotations and fine-tune using 20%, 50%, 80%, and 100% of the labeled data. As shown in Fig. 4, our method consistently outperforms CooPre across all settings, with more pronounced gains under limited annotations in V2V and IC. We attribute these improvements to complementarity-guided masking and dual reconstruction, which encourage learning geometric information beyond ego observations and support effective detection with fewer annotations.

4.3 Ablation Studies

Component-wise ablation. We evaluate the contributions of CGM, MCR, and ECR on V2X-Real under the VC setting using SECOND and AttFuse. The baseline trains the detector from scratch without pretraining. For controlled comparison, MCR uses uniform random masking at a ratio of 0.7 and uniform reconstruction weights when CGM is disabled, while all variants follow the same fine-tuning protocol.

As shown in Table 4, either MCR or ECR improves the baseline, and their combination yields further gains, demonstrating their complementary roles in representation learning. Adding CGM achieves the best performance among all variants, improving mAP@0.3 and mAP@0.5 by 4.2% and 3.7% over MCR + ECR, respectively, validating the effectiveness of complementarity guidance within the complete framework.

Masking strategy and ratio. Table 5 evaluates masking ratios from 0.5 to 0.9 under our complementarity-guided masking strategy on V2X-Real VC using AttFuse. A ratio of 0.7 achieves the best performance on both metrics. This suggests a favorable balance between reconstructing complementary geometry and retaining collaborative context to support reconstruction alongside the preserved ego observations.

Performance at different perception ranges. We evaluate performance across different perception ranges. As shown in Table 3, our method outperforms the compared methods in both VC and

V2V, with the largest gains over training from scratch observed at long ranges. We attribute this trend to complementarity-guided masking and dual reconstruction, which emphasize geometry provided by collaborators beyond ego observations, supporting detection in distant regions.

Robustness assessment. We evaluate the robustness of our method to missing collaborative regions and localization errors on V2X-Real under the VC setting with AttFuse. Specifically, we separately vary the proportion of removed occupied BEV blocks and the standard deviation of planar translation errors for collaborating agents, while keeping ego observations and ground-truth annotations unchanged. As shown in Fig. 5, although detection performance decreases with increasing perturbation severity, our method consistently achieves higher mAP@0.5 than CooPre across all evaluated levels, demonstrating its robustness under incomplete collaborative observations and spatial misalignment.

4.4 Visualization

Fig. 6 presents qualitative comparisons on V2X-Real. Compared with training from scratch and CooPre, our method provides more consistent object localization and reduces spurious predictions in the illustrated scenes. These observations complement the quantitative results and support the effectiveness of complementarity-guided pretraining in learning geometric representations for collaborative detection. Additionally, an illustration of complementarity-guided masking and reconstruction weighting is provided in Appendix C.1.

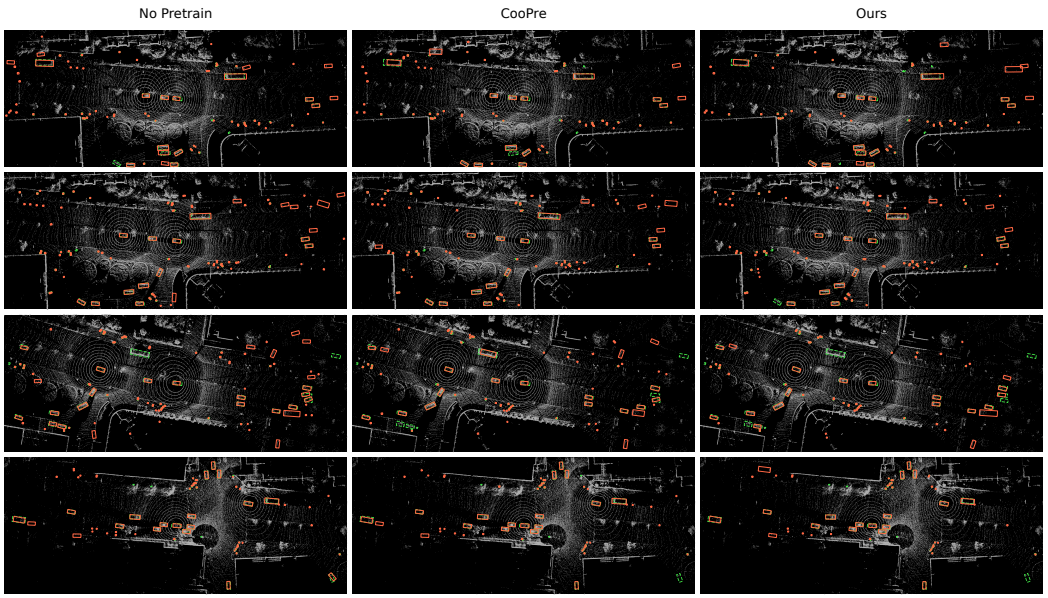

Figure 6: Qualitative detection results on V2X-Real in bird's-eye view. Each row shows the same scene across three methods, namely training from scratch, CooPre, and ours, from left to right.

5 Conclusion

In this paper, we present a complementarity-guided dual-pretext framework for self-supervised collaborative perception pretraining. CGM identifies complementary regions, MCR reconstructs their geometry from the remaining collaborative context, and ECR recovers the jointly observed scene from ego-only input. By jointly optimizing these reconstruction tasks, our framework learns transferable 3D representations without detection annotations. Experiments on V2X-Real and OPV2V demonstrate its effectiveness across different collaboration modes and fusion architectures.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing by improving the grammar, wording, and readability of the manuscript and providing critical feedback on the clarity of the methodology description and experimental presentation. The research ideas, methodology, implementation, experiments, data, and reported results were developed and produced by the authors. We reviewed, verified, and revised all AI-assisted textual edits. We take responsibility for the final content of this work, including its technical claims and experimental results.

REPRODUCIBILITY STATEMENT

To support reproducibility, Section 3 provides a detailed description of the overall methodological framework and each module, and Section 4 provides the execution details of the experiments as well as the datasets, data splits, and evaluation protocols used. Appendix A provides additional implementation details, while Appendix B offers additional method details. Furthermore, after this paper is accepted, we will publicly release the source code and models.

REFERENCES

Eduardo Arnold, Mehrdad Dianati, Robert de Temple, and Saber Fallah. Cooperative perception for 3d object detection in driving scenarios using infrastructure sensors. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):1852–1864, 2020.

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)*, pp. 514–524. IEEE, 2019.

Zihan Ding, Jiahui Fu, Si Liu, Hongyu Li, Siheng Chen, Hongsheng Li, Shifeng Zhang, and Xu Zhou. Point cluster: A compact message unit for communication-efficient collaborative perception. In *International Conference on Learning Representations*, 2025.

Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2D image transformers help 3D representation learning? In *International Conference on Learning Representations*, 2023.

Jingda Guo, Dominic Carrillo, Sihai Tang, Qi Chen, Qing Yang, Song Fu, Xi Wang, Nannan Wang, and Paparao Palacharla. Coff: Cooperative spatial feature fusion for 3-d object detection on autonomous vehicles. *IEEE Internet of Things Journal*, 8(14):11078–11087, 2021.

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*, volume 35, 2022.

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

Zhiwei Lin, Yongtao Wang, Shengxiang Qi, Nan Dong, and Ming-Hsuan Yang. BEV-MAE: Bird's eye view masked autoencoders for point cloud pre-training in autonomous driving scenarios. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3531–3539, 2024.

Genjia Liu, Anning Hu, Yue Hu, Wenjun Zhang, and Siheng Chen. Rate-distortion optimized pragmatic communication for collaborative perception. In *International Conference on Learning Representations*, 2026.

Yifan Lu, Yue Hu, Yiqi Zhong, Dequan Wang, Yanfeng Wang, and Siheng Chen. An extensible framework for open heterogeneous collaborative perception. In *International Conference on Learning Representations*, 2024.

Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Computer Vision – ECCV 2022*, 2022.

Xiaoyu Tian, Haoxi Ran, Yue Wang, and Hang Zhao. GeoMAE: Masked geometric target prediction for self-supervised point cloud pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Binglu Wang, Lei Zhang, Zhaozhong Wang, Yongqiang Zhao, and Tianfei Zhou. CORE: Cooperative reconstruction for multi-agent perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8710–8720, 2023.

Hao Xiang, Zhaoliang Zheng, Xin Xia, Runsheng Xu, Letian Gao, Zewei Zhou, Xu Han, Xinkai Ji, Mingxi Li, Zonglin Meng, Li Jin, Mingyue Lei, Zhaoyang Ma, Zihang He, Haoxuan Ma, Yunshuang Yuan, Yingqian Zhao, and Jiaqi Ma. V2X-Real: A largs-scale dataset for vehicle-to-everything cooperative perception. In *Computer Vision – ECCV 2024*, volume 15110 of *Lecture Notes in Computer Science*, pp. 455–470. Springer, 2025.

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

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 *Computer Vision – ECCV 2022*, volume 13699 of *Lecture Notes in Computer Science*, pp. 107–124. Springer, 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 *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2583–2589. IEEE, 2022b.

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)*, pp. 2583–2589. IEEE, 2022c.

Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers. In *Conference on Robot Learning*, pp. 989–1000. PMLR, 2023.

Siming Yan, Chen Song, Youkang Kong, and Qixing Huang. Multi-view representation is what you need for point-cloud pre-training. In *International Conference on Learning Representations*, 2024a.

Siming Yan, Yuqi Yang, Yu-Xiao Guo, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Qixing Huang. 3D feature prediction for masked-autoencoder-based point cloud pretraining. In *International Conference on Learning Representations*, 2024b.

Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

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

Seth Z. Zhao, Hao Xiang, Chenfeng Xu, Xin Xia, Bolei Zhou, and Jiaqi Ma. Coopre: Cooperative pretraining for V2X cooperative perception. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11765–11772, 2025.

Zewei Zhou, Seth Z. Zhao, Tianhui Cai, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. TurboTrain: Towards efficient and balanced multi-task learning for multi-agent perception and prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4391–4402, 2025.

A More Implementation Details

A.1 Datasets and Evaluation Protocols

We conduct experiments on V2X-Real (Xiang et al., 2025) and OPV2V (Xu et al., 2022b). V2X-Real is a real-world collaborative perception dataset with vehicle-mounted and roadside LiDAR observations. We consider four collaboration modes: vehicle-centric (VC), infrastructure-centric (IC), vehicle-to-vehicle (V2V), and infrastructure-to-infrastructure (I2I). We use the designated training and test splits and retain up to four agents, including the ego agent, within a communication range of 50 m. The BEV evaluation range is $x \in [-100, 100]$ m and $y \in [-40, 40]$ m.

OPV2V is a simulated V2V collaborative perception dataset generated using CARLA and OpenCDA. It contains 73 scenarios, 11,464 synchronized frames, and 232,913 annotated 3D vehicle bounding boxes. We pretrain and fine-tune on the training split and evaluate on the Default and Culver City test sets. Each sample retains up to five agents within a communication range of 70 m. The BEV evaluation range is $x \in [-140, 140]$ m and $y \in [-40, 40]$ m.

Following the OpenCOOD evaluation protocol, we compute average precision (AP) using BEV intersection over union (IoU). On V2X-Real, we report AP at IoU thresholds of 0.3 and 0.5 for cars, pedestrians, and trucks, together with their arithmetic mean, denoted as mAP. On OPV2V, we report vehicle AP at IoU thresholds of 0.5 and 0.7. Pretraining uses only training point clouds without detection annotations. Data-efficiency experiments use 20%, 50%, 80%, and 100% of the labeled fine-tuning data while retaining the full pretraining set. Robustness experiments separately introduce missing collaborative regions and planar translation errors, with ego observations and ground-truth annotations unchanged.

A.2 Model Configurations

The encoder consists of mean voxel feature encoding, a SECOND sparse 3D backbone (Yan et al., 2018), and height compression. Each point is represented by its three-dimensional coordinates and intensity. The encoder produces a 256-channel BEV feature map with a spatial downsampling factor of eight. The shared reconstruction decoder comprises a 3×3 convolutional projection, three residual blocks with dilation rates of 1, 2, and 4, and a 1×1 coordinate prediction head. Each residual block contains two 3×3 convolutions, and the decoder uses 256 hidden channels while preserving the BEV spatial resolution.

Table 6 summarizes the geometric configurations. CGM computes occupied-voxel statistics on the same fine voxel grid used for encoder input. Each masking block spans 4×4 reconstruction cells, corresponding to 3.2×3.2 m in BEV. Each reconstruction cell spans the full vertical extent of the input domain. MCR and ECR share all encoder and decoder parameters and process their respective inputs through separate forward passes. For downstream detection, the pretrained encoder is paired with AttFuse, F-Cooper, or V2X-ViT, using the corresponding fusion module and detection head.

Table 6: Geometric configurations for pretraining. Spatial dimensions are specified along the x, y, and z axes.

Configuration	V2X-Real	OPV2V
Input voxel size (m)	$0.1 \times 0.1 \times 0.2$	$0.1 \times 0.1 \times 0.1$
Reconstruction cell size (m)	$0.8 \times 0.8 \times 8.0$	$0.8 \times 0.8 \times 4.0$
CGM block size (BEV cells)	4×4	4×4
Input points per voxel (max.)	5	5
Target points per cell (max.)	64	64
Predicted points per cell	20	20

A.3 Pretraining and Fine-Tuning Settings

Our framework is implemented in PyTorch based on OpenCOOD, and experiments are conducted on a single NVIDIA RTX PRO 6000 GPU. Table 7 summarizes the training configurations. We pretrain for 15 epochs on both datasets and fine-tune for 20 epochs on V2X-Real and 40 epochs on

Table 7: Training configurations. Batch size, optimizer, peak learning rate, and weight decay apply to both pretraining and fine-tuning.

Configuration	V2X-Real	OPV2V
Pretraining epochs	15	15
Fine-tuning epochs	20	40
Batch size	4	4
Optimizer	Adam	AdamW
Peak learning rate	2×10^{-3}	2×10^{-3}
Weight decay	10^{-4}	10^{-2}
Minimum learning rate	2×10^{-5}	2×10^{-5}
Pretraining warm-up epochs	2	0

OPV2V, using a batch size of four throughout. Both stages use Adam on V2X-Real and AdamW on OPV2V, with a peak learning rate of 2×10^{-3}. The learning rate follows a cosine schedule with a minimum value of 2×10^{-5}.

Data augmentation includes random flipping along the x axis, rotation within $[-\pi/4, \pi/4]$, and scaling within $[0.95, 1.05]$. During pretraining, the same sampled augmentation is applied to all aligned agent point clouds before constructing the inputs and reconstruction target. Unless otherwise specified, the target masking ratio is $\rho = 0.7$, and the two reconstruction objectives are equally weighted with $\lambda_m = \lambda_e = 1$.

The encoder from the final pretraining epoch initializes each downstream detector. The reconstruction decoder is discarded, and the encoder, fusion module, and detection head are jointly fine-tuned using detection annotations. Evaluation uses the final fine-tuning checkpoint. OPV2V training does not use a validation split. CGM and reconstruction are used only during pretraining and introduce no additional modules into downstream inference.

B Additional Method Details

B.1 Detailed Design of Complementarity-Guided Masking

CGM estimates observation support from occupied-voxel statistics after coordinate alignment, shared augmentation, and spatial cropping. The fine voxel grid and BEV block dimensions follow Appendix A.2. For each agent, multiple points within the same fine voxel contribute only once to the count $n_i(b)$. We compute the normalized support and ego-relative complementarity using Eq. equation 1, with $n_{\text{cap}} = 64$. The aggregated score is $s(b) = \max_{1 \le i < N} s_i(b)$, and candidate blocks satisfy $\mathcal{C} = \{b \in \mathcal{B} : s(b) > \tau\}$, where $\tau = 0.05$.

To retain informative collaborative context, we identify the agent with the largest total complementarity over the block grid, $i^\star = \arg\max_{1 \le i < N} \sum_{b \in \mathcal{B}} s_i(b)$. Among candidate blocks with $s_{i^\star}(b) > 0$, we reserve the block with the largest contribution from this agent and denote the reserved set by $\mathcal{R}$. If no such block exists, $\mathcal{R} = \varnothing$. We then partition $\mathcal{C}$ into four-connected components $\mathcal{C}_k$, where adjacent blocks share a horizontal or vertical edge. For a component containing $m_k = |\mathcal{C}_k| > 1$ blocks, its masking quota is

$$q_k = \min\left\{\text{round}(\rho m_k),\ m_k - \max\left(1, \lceil \eta m_k \rceil\right),\ |\mathcal{C}_k \setminus \mathcal{R}|\right\}. \tag{9}$$

The default configuration uses $\rho = 0.7$ and $\eta = 0.3$. Singleton components remain unmasked. From each $\mathcal{C}_k \setminus \mathcal{R}$, we sample q_k blocks without replacement, with sampling weights proportional to $s(b)$. The selected blocks form $\mathcal{M}$; unused quotas are not redistributed across components.

The retention constraint preserves at least $\max(1, \lceil \eta m_k \rceil)$ blocks in each component, while the reserved block retains context from the selected collaborator. Consequently, the realized masking ratio, $|\mathcal{M}| / \max(1, |\mathcal{C}|)$, may be lower than the requested ratio ρ. This ratio is measured over candidate blocks. CGM removes all collaborator points within $\mathcal{M}$ and preserves every ego point. When no collaborators or candidate blocks are available, $\mathcal{M} = \varnothing$ and MCR contributes no reconstruction loss.

B.2 Reconstruction Target Construction

MCR and ECR use a common target constructed from the complete set of aligned, augmented, and cropped observations. Specifically, the operator $\mathcal{T}$ in Eq. equation 2 partitions the unmasked point concatenation $\bigcup_{i=0}^{N-1} \mathcal{P}_i$ into reconstruction cells. Point concatenation preserves observations from all agents without cross-agent deduplication. CGM changes the collaborative input but leaves this target unchanged.

Each reconstruction cell corresponds to one BEV location and spans the full vertical extent of the input domain. We retain at most 64 target points per cell. For a point p in cell c, its normalized coordinates are

$$\widetilde{p}_c = \left(\frac{p_x - \mu_x(c)}{\Delta_x}, \frac{p_y - \mu_y(c)}{\Delta_y}, \frac{p_z}{\Delta_z} \right), \tag{10}$$

where $(\mu_x(c), \mu_y(c))$ is the BEV cell center and $(\Delta_x, \Delta_y, \Delta_z)$ are the cell dimensions specified in Appendix A.2. Horizontal coordinates are centered within each cell, whereas the vertical coordinate is scaled without subtracting a cell center or the lower range boundary.

Let Y_c contain the valid normalized target points and $\mathcal{O} = \{c : |Y_c| > 0\}$ denote the occupied target cells. Cells with fewer than 64 points are padded for batched computation, and the padded entries are excluded from distance matching and loss averaging. The decoder predicts $K = 20$ points per cell in the same normalized coordinate system. Reconstruction supervises point coordinates; intensity is retained as an encoder input feature.

B.3 Detailed Reconstruction Objectives

For a predicted point set A and a nonempty target point set B within the same reconstruction cell, we use the symmetric L_1 Chamfer distance:

$$\mathrm{CD}_1(A, B) = \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} \|a - b\|_1 + \frac{1}{|B|} \sum_{b \in B} \min_{a \in A} \|b - a\|_1. \tag{11}$$

Each direction is normalized by the number of points in its source set. Point matching is performed within individual cells, and target-empty cells receive no geometric supervision.

MCR first averages cell errors within each selected block and then combines the block means using the complementarity weights in Eq. equation 4. This prevents the number of occupied target cells from directly scaling a block's contribution. The weights are computed over the entire selected mask set. A selected block without valid target cells contributes zero, and its weight is not redistributed to other blocks. MCR supervises the jointly observed geometry within masked blocks, including any ego observations retained there. ECR averages reconstruction errors over all occupied cells of the same target, using only ego observations as input.

For a mini-batch, let $\ell_t^{(j)}$ denote the per-sample objective for sample j and task $t \in \{\mathrm{MCR}, \mathrm{ECR}\}$, as defined in Eqs. equation 5 and equation 7. Let $\mathcal{I}_t$ contain the samples with valid supervision for task t. An MCR sample is valid when at least one positively weighted selected block contains an occupied target cell; an ECR sample is valid when its target contains at least one occupied cell. The batch-level objective is

$$\mathcal{L}_t = \begin{cases} \dfrac{1}{|\mathcal{I}_t|} \displaystyle\sum_{j \in \mathcal{I}_t} \ell_t^{(j)}, & |\mathcal{I}_t| > 0, \\ 0, & |\mathcal{I}_t| = 0. \end{cases} \tag{12}$$

The two tasks may have different valid-sample sets. An empty mask disables MCR for that sample, while ECR remains active if its target is nonempty. The masked collaborative and ego-only inputs undergo separate forward passes through the same encoder and decoder. Their batch-level objectives are combined according to Eq. equation 8 and optimized through a single backward pass and parameter update.

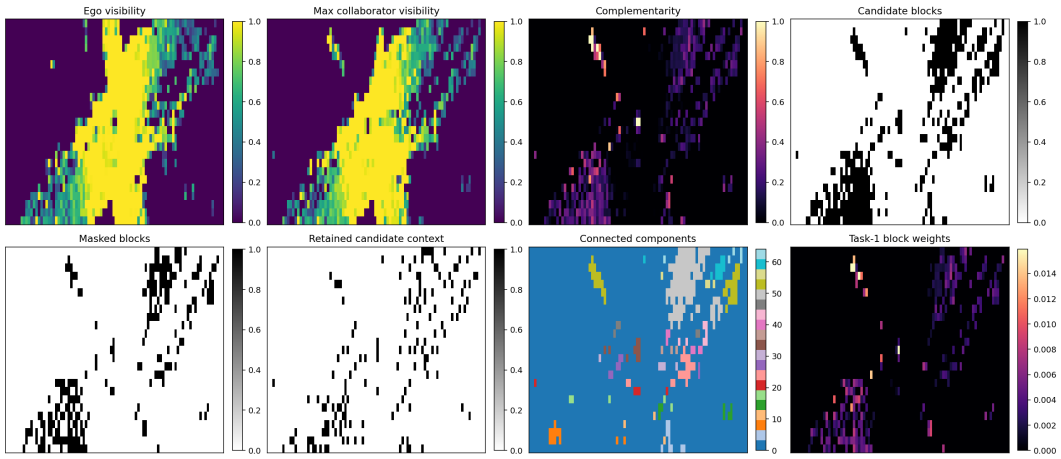

Figure 7: Illustration of complementarity-guided masking and reconstruction weighting. The top row presents ego visibility, maximum collaborator visibility, complementarity scores, and candidate blocks. The bottom row presents the selected masked blocks, retained collaborative context, connected components, and complementarity-based weights for MCR.

C Additional Visualizations

C.1 Visualization of Complementarity-Guided Masking

Fig. 7 visualizes CGM on V2X-Real. The complementarity map highlights regions with limited ego support and stronger collaborator observations. These scores guide block selection under context-retention constraints and determine the MCR weights. The visualization illustrates how spatial masking and reconstruction weighting emphasize complementary geometry.

C.2 Additional Qualitative Detection Results

Qualitative comparison. Fig. 8 compares the downstream detection results obtained with different initialization strategies. Our method produces predictions that capture vehicle locations and orientations across the illustrated scenes, including regions with relatively sparse point returns. These examples illustrate the ability of the learned representation to support object localization when geometric evidence is unevenly distributed across the scene. The comparisons complement the quantitative evaluation by showing how representation quality is reflected in individual detections.

Learning from complementary observations. A key strength of our method is that the pretraining objective explicitly accounts for differences between ego and collaborator observations. Complementarity-guided masking (CGM) prioritizes regions with limited ego support and stronger collaborator observations, while retaining all ego points and part of the collaborative context. Masked collaborative reconstruction (MCR) then requires the model to recover geometry in these selected regions. Complementarity-based reconstruction weights further emphasize regions where collaborators provide additional geometric evidence. Together, these choices direct learning toward structures that are difficult to characterize from the ego scan alone, providing a targeted training signal for collaborative perception.

Complementary benefits of the two reconstruction tasks. Ego-conditioned scene reconstruction (ECR) complements MCR by requiring the same encoder and decoder to reconstruct the jointly observed scene from ego-only input. Whereas MCR learns to exploit retained collaborative context, ECR encourages the representation to capture broader geometric relationships under restricted observations. Their shared optimization therefore trains the encoder under two different levels of information availability. This offers a plausible explanation for the method's ability to support detection in regions with incomplete local evidence, namely that the representation is trained both to use

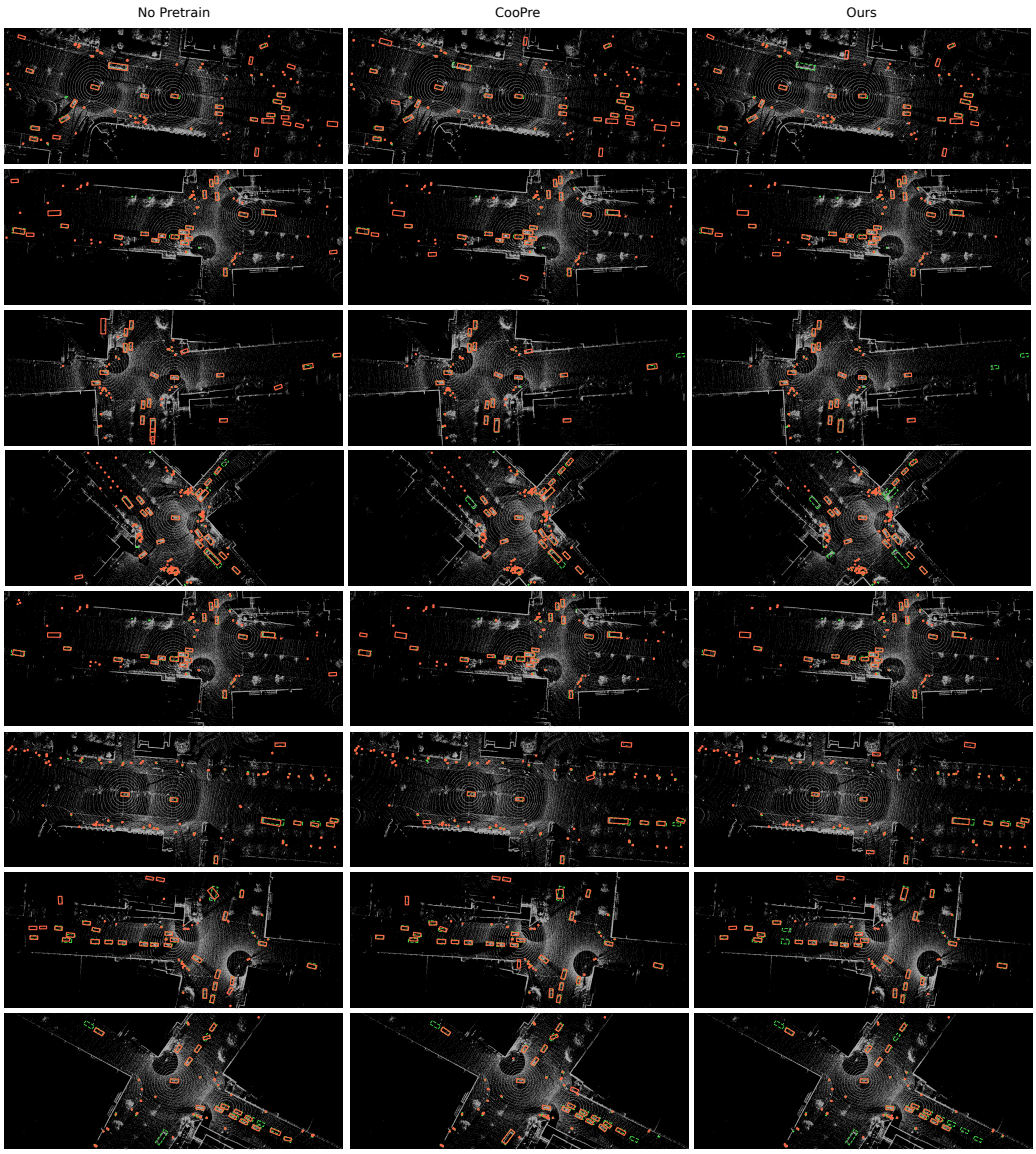

Figure 8: Qualitative detection comparisons on V2X-Real in bird's-eye view. Each row shows a representative frame, with columns corresponding to No Pretrain, CooPre, and Ours, respectively. Gray points depict LiDAR observations; green dashed boxes and orange-red solid boxes indicate ground-truth annotations and predictions, respectively.

complementary observations and to infer geometry when some observations are unavailable. These benefits are transferred through encoder initialization, and the reconstruction decoder is discarded for downstream detection.