GeoDP: Geometry-anchored Dynamic Prompt for Heterogeneous Collaborative Perception

Anonymous authors
Paper under double-blind review

Abstract

Collaborative perception improves the perception performance of individual agents through information sharing, but feature heterogeneity caused by different sensors and perception models poses significant challenges for practical applications. Existing methods mainly focus on pairwise adapters or naive feature-space alignment, which either scale poorly as model diversity increases or lack essential geometric information and cannot dynamically adapt to different models. To address these issues, we propose GeoDP, a Geometry-anchored Dynamic Prompt framework for heterogeneous collaborative perception. Specifically, we leverage a geometry foundation model to construct a Geometry-anchored Unified Space (GUS), which provides the essential geometric information required for 3D collaborative perception and offers a unified reference feature space for heterogeneous agents. Furthermore, a Dynamic Prompt Adapter (DPA) generates input-conditioned prompts based on the discrepancy between each agent feature and the geometry anchor, scaling flexibly and dynamically to diverse models. Finally, Adaptive Spatial Fusion (ASF) assigns different weights to the adapted features for reliable collaboration. Extensive experiments on OPV2V and DAIR-V2X demonstrate that GeoDP achieves new state-of-the-art performance across both seen and unseen models, outperforming existing methods by a large margin and exhibiting strong robustness. The code and model will be publicly available.

1 Introduction

Collaborative perception (Wang et al., 2020; Li et al., 2021; Xu et al., 2022c) allows multiple agents to alleviate occlusion and long-range limitations via exchanging complementary perception information, which is of great significance for autonomous driving, embodied intelligence, and spatial intelligence. The past few years have witnessed the rapid development of collaborative perception across early (Arnold et al., 2020; Yu et al., 2022), intermediate (Xu et al., 2023b; 2025), and late (Li et al., 2022b; Xu et al., 2022c) collaboration under homogeneous multi-agent conditions. However, collaborating agents in real-world applications are often heterogeneous due to different sensor modalities and model architectures, producing features with inconsistent spatial resolutions, channel dimensions, response distributions, and semantic structures, which may lead to severe performance degradation (Xu et al., 2022a; Lu et al., 2024).

To boost performance in heterogeneous multi-agent collaborative perception, existing studies mainly follow two paradigms, namely pairwise feature adaptation and feature-space alignment. Pairwise feature adaptation, as shown in Fig. 1(a), trains a dedicated adapter for each pair of collaborating agents (Xu et al., 2022a; Luo et al., 2024; Xia et al., 2025). Naive feature-space alignment, as shown in Fig. 1(b), maps heterogeneous features into a predefined unified space through agent-specific adapters (Lu et al., 2024; Gao et al., 2025; Shao et al., 2026; Wang et al., 2026). However, we argue that two major issues remain: (1) alignment targets are often defined by specific agents or initial model configurations, which can introduce representation bias and leave diverse features without independent geometric guidance; (2) fixed adapters cannot adjust to input-specific feature discrepancies as agents and scenes change, hindering generalization to unseen models in real-world deployments.

Inspired by recent geometry foundation models that provide rich 3D cues from multi-view images (Wang et al., 2025), and by prompt learning that enables lightweight adaptation (Jia et al., 2022),

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.

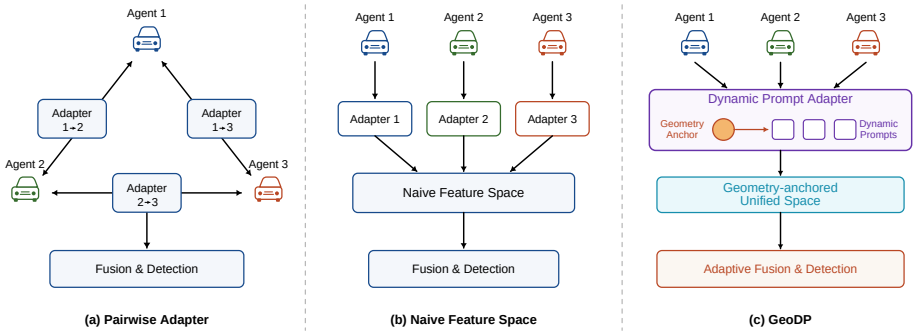

Figure 1: Comparison of heterogeneous collaborative perception paradigms. (a) Pairwise feature adaptation trains a dedicated adapter for each pair of collaborating agents. (b) Naive feature-space alignment maps heterogeneous features into a predefined unified space. (c) GeoDP constructs a Geometry-anchored Unified Space and uses dynamic prompts to adapt heterogeneous features.

in this paper, we propose GeoDP, a geometry-anchored dynamic prompt framework for heterogeneous collaborative perception, as illustrated in Fig. 1(c). GeoDP leverages a geometry foundation model (Wang et al., 2025) to construct the Geometry-anchored Unified Space (GUS), providing geometric guidance and a unified reference for heterogeneous feature alignment. A Dynamic Prompt Adapter (DPA) generates input-conditioned prompts from the discrepancy between each agent's features and a geometry anchor, enabling flexible adaptation to different models. Adaptive Spatial Fusion (ASF) then weights the adapted features spatially for reliable collaboration. Extensive experiments on OPV2V and DAIR-V2X demonstrate that our GeoDP significantly outperforms existing methods on both seen and unseen models, while maintaining strong robustness under cross-modal, cross-dataset, and pose-noise conditions.

In summary, the main contributions of this paper are as follows:

- We propose GeoDP, a novel Geometry-anchored Dynamic Prompt framework for multi-agent heterogeneous collaborative perception.
- We construct a Geometry-anchored Unified Space (GUS) using a geometry foundation model, which provides essential geometric information and a unified model-independent reference space for heterogeneous agents.
- We design a Dynamic Prompt Adapter (DPA) that generates input-conditioned prompts from agent-to-anchor discrepancies, dynamically adapting both seen and unseen features without modifying local perception models.
- Extensive experiments on simulated and real-world datasets demonstrate that GeoDP achieves new SOTA performance across both seen and unseen models, outperforming existing methods by a large margin and exhibiting strong robustness.

2 Related Work

2.1 Collaborative Perception

Collaborative perception alleviates occlusion and limited sensing ranges by sharing multi-view observations among agents (Wang et al., 2020; Li et al., 2021; Xu et al., 2022c). Existing methods can mainly be categorized into early, intermediate, and late collaboration according to the stage of information exchange. Early collaboration shares raw sensor data, preserving rich information at high communication cost (Chen et al., 2019; Arnold et al., 2020), whereas late collaboration exchanges compact predictions but loses fine-grained cues (Arnold et al., 2020; Teufel et al., 2023; Xu et al., 2023a). Intermediate collaboration shares learned features and achieves a favorable balance between perception performance and communication cost (Wang et al., 2020; Xu et al., 2022b). However, these methods generally assume that collaborating agents share a consistent feature space and overlook the feature heterogeneity caused by different sensor modalities and perception models,

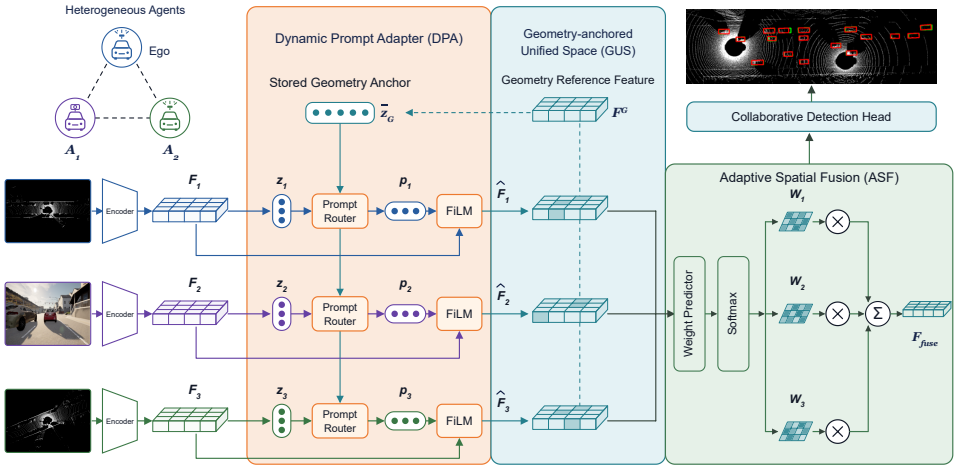

Figure 2: Framework overview. GeoDP first constructs a Geometry-anchored Unified Space (GUS) to provide essential geometric information and a unified reference feature space. Then, a Dynamic Prompt Adapter (DPA) generates input-conditioned prompts from agent-to-anchor discrepancies, dynamically adapting both seen and unseen features. Finally, Adaptive Spatial Fusion (ASF) assigns different weights to the adapted features for reliable collaboration.

making their interaction and aggregation mechanisms difficult to apply directly when features differ in spatial resolution, channel dimension, response distributions, and semantic structures.

2.2 Heterogeneous Collaborative Perception

Heterogeneous collaborative perception aims to mitigate feature discrepancies caused by different sensor modalities and perception models. MPDA (Xu et al., 2022a) and PnPDA (Luo et al., 2024) use domain adaptation to align heterogeneous features through pairwise mappings, while HEAL (Lu et al., 2024) enables new agents to join an existing collaborative system through a shared representation space. STAMP (Gao et al., 2025), NegoCollab (Shao et al., 2026), CoDS (Han et al., 2025), and GT-Space (Wang et al., 2026) further improve heterogeneous compatibility through model-agnostic mapping, common-representation negotiation, domain separation, and ground-truth feature spaces, respectively. PolyInter (Xia et al., 2025) and PEARL (Maleki & Radha, 2026) introduce prompts for cross-model feature interpretation and feature-space alignment. However, these methods either scale poorly as model diversity increases or lack essential geometric information and cannot dynamically adapt to different models. To bridge this gap, we construct a geometry-anchored unified space and design a dynamic prompt adapter to effectively improve heterogeneous collaborative perception.

3 Methodology

3.1 Framework Overview

As illustrated in Fig. 2, we propose GeoDP for collaborative agents with heterogeneous sensor modalities and perception architectures. GeoDP consists of a Geometry-anchored Unified Space (GUS), a Dynamic Prompt Adapter (DPA), and an Adaptive Spatial Fusion module (ASF). GUS provides a shared geometric reference for heterogeneous features, DPA dynamically adapts each input feature, and ASF spatially aggregates the adapted collaborative information. Specifically, the i-th agent first extracts a heterogeneous feature $\mathbf{F}_i$ using its individual perception encoder. During offline construction, a geometry-aware bird's-eye view (BEV) encoder produces the unified-space reference feature $\mathbf{F}^G$. Its descriptor $\mathbf{z}_G$ is accumulated through EMA to form the stored geometry anchor $\bar{\mathbf{z}}_G$. During online inference, DPA extracts an agent descriptor $\mathbf{z}_i$ from $\mathbf{F}_i$ and generates a dynamic prompt $\mathbf{p}_i$ according to its discrepancy from $\bar{\mathbf{z}}_G$, producing the unified-space adapted feature $\widehat{\mathbf{F}}_i$. ASF then predicts a spatial weight map $\mathbf{W}_i$ for each agent and aggregates the adapted features into $\mathbf{F}_{\text{fuse}}$, which is processed by the collaborative detection head.

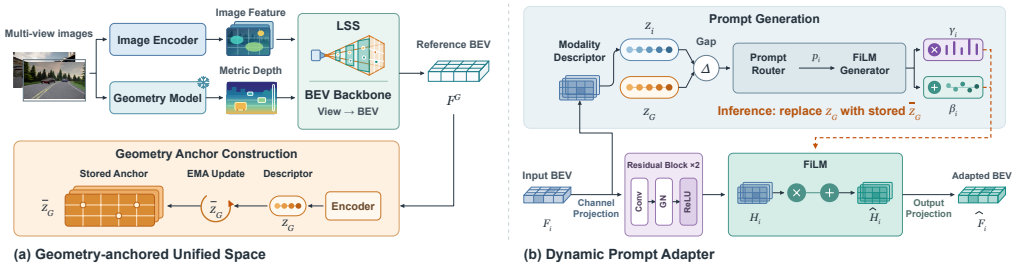

Figure 3: Architectures of the core modules in GeoDP. (a) Geometry-anchored Unified Space (GUS) provides a geometric reference for feature alignment and stores a geometry anchor for inference. (b) Dynamic Prompt Adapter (DPA) generates input-conditioned prompts from agent-to-geometry discrepancies to adapt heterogeneous features to GUS.

3.2 Geometry-anchored Unified Space

Shared BEV representations provide a common spatial reference that preserves geometric and semantic information for multi-sensor feature fusion (Liu et al., 2023). However, spaces defined by a specific agent may be biased toward its representation, making heterogeneous features with large domain gaps difficult to align (Shao et al., 2026). We therefore construct the Geometry-anchored Unified Space (GUS) from multi-view geometric information to provide a stable unbiased reference for feature alignment.

As shown in Fig. 3(a), we construct a geometry-anchored unified space in offline manner. Given the multi-view images $\mathcal{I}_i = \{\mathbf{I}_i^v\}_{v=1}^V$ of the i-th agent, a frozen geometry foundation model instantiated with VGGT (Wang et al., 2025) predicts multi-view depth, which is converted into metric depth. A camera encoder then extracts image features, while the discretized depth is used to lift and project them onto a common BEV grid following Lift-Splat-Shoot (Philion & Fidler, 2020):

$$\{\mathbf{D}_i^v\}_{v=1}^V = \mathcal{C}(\mathcal{G}_{\mathrm{VGGT}}(\mathcal{I}_i)), \tag{1}$$

$$\mathbf{F}_i^G = \mathcal{S}_{\mathrm{BEV}}\left(\{\mathcal{E}_{\mathrm{cam}}(\mathbf{I}_i^v) \otimes \mathcal{Q}(\mathbf{D}_i^v), \mathbf{K}_i^v, \mathbf{T}_i^v\}_{v=1}^V\right), \tag{2}$$

where $\mathcal{C}(\cdot)$ and $\mathcal{Q}(\cdot)$ denote depth-scale conversion and discretization, respectively, and $\mathcal{S}_{\mathrm{BEV}}(\cdot)$ denotes geometric projection. The resulting features are spatially aligned and aggregated into a geometry reference feature $\mathbf{F}^G$. We train this reference branch with 3D detection supervision, enabling $\mathbf{F}^G$ to preserve both geometric structure and task-relevant semantics. Its spatial resolution, channel structure, and semantic organization define the unified feature space.

To retain this geometric reference during online collaboration, we encode $\mathbf{F}^G$ into a compact descriptor and construct a stored geometry anchor using exponential moving averaging (Tarvainen & Valpola, 2017; Li et al., 2022a):

$$\bar{\mathbf{z}}_G^{(t)} = \mu \bar{\mathbf{z}}_G^{(t-1)} + (1-\mu)\frac{1}{B}\sum_{b=1}^{B} \mathcal{E}_G\left(\mathbf{F}_b^{(G,t)}\right), \tag{3}$$

where $\mathcal{E}_G(\cdot)$ is the geometry descriptor encoder and μ is the momentum coefficient. After GUS construction, the VGGT-based reference branch is removed from online inference, while the stored geometry anchor $\bar{\mathbf{z}}_G$ is retained to guide heterogeneous feature adaptation.

3.3 Dynamic Prompt Adapter

The discrepancy between heterogeneous features and GUS varies across perception models and input scenes, making fixed adaptation insufficient for flexible alignment. DPA therefore characterizes each incoming feature relative to the geometry anchor and generates an input-conditioned prompt to dynamically control its adaptation.

As illustrated in Fig. 3(b), DPA maps heterogeneous agent features into GUS using prompts conditioned on their discrepancies from the geometric reference. Given an agent feature $\mathbf{F}_i$, an input-specific channel projection first produces $\mathbf{X}_i = \mathcal{P}_i(\mathbf{F}_i)$. An intrinsic feature descriptor then encodes

its channel statistics, global response, and cross-channel correlations:

$$\mathbf{z}_i = \mathrm{Norm}(\mathcal{E}_z[\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i, \mathbf{r}_i, \mathbf{g}_i]), \tag{4}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ denote the channel-wise mean and standard deviation, while $\mathbf{r}_i$ and $\mathbf{g}_i$ represent the global-response and Gram features, respectively (Gatys et al., 2016). At inference, the normalized descriptor $\mathbf{z}_i$ characterizes the current feature distribution without taking sensor or model labels as input.

During training, $\mathbf{z}_i$ is compared with the current reference descriptor $\mathbf{z}_G$ extracted from $\mathbf{F}^G$. Their difference and element-wise interaction are combined to construct the geometry-gap representation, from which the prompt router generates an input-conditioned prompt:

$$\boldsymbol{\delta}_i = [\mathbf{z}_i, \mathbf{z}_G, \mathbf{z}_i - \mathbf{z}_G, \mathbf{z}_i \odot \mathbf{z}_G], \qquad \mathbf{p}_i = \mathcal{R}_p(\boldsymbol{\delta}_i). \tag{5}$$

At inference, $\mathbf{z}_G$ in Eq. 5 is replaced by the stored geometry anchor $\bar{\mathbf{z}}_G$. Consequently, $\mathbf{p}_i$ varies with the scene and input-feature distribution rather than remaining fixed for each perception model.

The generated prompt controls feature adaptation through feature-wise linear modulation (FiLM) (Perez et al., 2018). The projected feature is first processed by a residual refinement block, while the prompt is mapped to channel-wise scaling and bias parameters:

$$\mathbf{H}_i = \mathcal{B}(\mathbf{X}_i), \qquad [\boldsymbol{\gamma}_i, \boldsymbol{\beta}_i] = \mathcal{M}(\mathbf{p}_i). \tag{6}$$

The modulation and output projection are then formulated as:

$$\widehat{\mathbf{H}}_i = (1 + \boldsymbol{\gamma}_i) \odot \mathrm{GN}(\mathbf{H}_i) + \boldsymbol{\beta}_i, \qquad \widehat{\mathbf{F}}_i = \mathcal{P}_o(\widehat{\mathbf{H}}_i), \tag{7}$$

where $\mathrm{GN}(\cdot)$ denotes group normalization (Wu & He, 2018) and $\mathcal{P}_o(\cdot)$ denotes output projection. The residual scaling form $1 + \boldsymbol{\gamma}_i$ improves adaptation stability. When necessary, $\widehat{\mathbf{F}}_i$ is resized to the spatial resolution defined by GUS. During DPA training, the local perception encoders, GUS, and unified detection head remain frozen, and only DPA is optimized using detection, feature-alignment, descriptor-classification, and prompt-regularization objectives.

Beyond adapting features from seen models, DPA also supports zero-shot collaboration with unseen local models. An unseen model may be independently pretrained, but none of its features or parameters is used during GeoDP training. At inference, its channel-compatible BEV feature is processed by the frozen projection branches learned from seen models and aggregated according to their descriptor distances to the stored geometry anchor. No GeoDP parameter is introduced or optimized for the unseen model. The complete procedure is provided in Appendix B.2.

3.4 Adaptive Spatial Fusion

Reliable collaborative detection requires effective aggregation of the adapted features to exploit complementary information from different agents. As illustrated in Fig. 2, DPA maps heterogeneous agent features into the unified feature space, producing $\{\widehat{\mathbf{F}}_i\}_{i=1}^N$. However, the reliability of these features may vary across spatial locations because of differences in sensor coverage, occlusion, and observation distance. Before fusion, all adapted features are transformed into the ego coordinate system and resized to the spatial resolution defined by GUS. For simplicity, we retain the notation $\widehat{\mathbf{F}}_i$ after spatial alignment.

Adaptive Spatial Fusion (ASF) concatenates the aligned features along the channel dimension and predicts a spatial score map for each agent:

$$\mathbf{S} = \mathcal{G}_f\left(\mathrm{Concat}\left[\widehat{\mathbf{F}}_1, \ldots, \widehat{\mathbf{F}}_N\right]\right), \tag{8}$$

where $\mathcal{G}_f(\cdot)$ is a lightweight convolutional weight predictor and $\mathbf{S} \in \mathbb{R}^{N \times H \times W}$. The predicted scores are normalized across the agent dimension using a temperature-controlled softmax:

$$\mathbf{W} = \mathrm{Softmax}_{\mathrm{agent}}\left(\frac{\mathbf{S}}{\tau}\right), \tag{9}$$

where τ is the temperature and $\mathbf{W}_i \in \mathbb{R}^{H \times W}$ is the spatial weight map of the i-th agent. At each spatial location, the weights of all available agents sum to one. Unavailable input slots are masked before softmax normalization.

The unified collaborative feature is obtained through position-wise weighted aggregation:

$$\mathbf{F}_{\text{fuse}} = \sum_{i=1}^{N} \mathbf{W}_i \odot \widehat{\mathbf{F}}_i. \tag{10}$$

Unlike assigning a single global weight to each agent, ASF adjusts agent contributions independently at different spatial locations. It can therefore emphasize complementary observations while suppressing unreliable or redundant responses.

Only ASF is optimized at this stage, while GUS, DPA, the local perception encoders, and the unified detection head remain frozen. The reference feature $\mathbf{F}^G$ is used only for training supervision and is not provided to the weight predictor. Thus, neither VGGT nor the GUS reference-feature branch is required during online inference.

3.5 Training Objective

During GeoDP training, the independently pretrained local perception encoders and the geometry foundation model remain frozen, allowing the framework to learn heterogeneous collaboration without modifying individual perception models. For each scene, the local encoders produce heterogeneous BEV features, while the geometry reference branch constructs the reference feature $\mathbf{F}^G$ and accumulates its descriptors through EMA to form the stored geometry anchor. Guided by this geometric reference, DPA adapts each heterogeneous feature into GUS, ASF aggregates the adapted features according to their spatial contributions, and the unified detection head produces the final predictions. The modules are optimized progressively according to their dependencies, while the established geometric reference remains fixed during downstream optimization. The overall training objective is summarized as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{GUS}} + \mathcal{L}_{\text{DPA}} + \mathcal{L}_{\text{ASF}}. \tag{11}$$

Here, the three terms supervise unified-space construction, dynamic feature adaptation, and adaptive spatial fusion, respectively. Their detailed definitions, loss weights, and module-wise optimization settings are provided in Appendix B.4. During inference, GeoDP retains the local perception encoders, DPA, ASF, the stored geometry anchor, and the unified detection head.

Table 1: Quantitative comparison under seen-model settings on OPV2V. We report AP@0.5 and AP@0.7, with the best results highlighted in bold and the second-best underlined.

Method	**AP@0.5**				**AP@0.7**			
	m1m2	m1m3	m2m4	all	m1m2	m1m3	m2m4	all
No Fusion	48.2	79.4	22.1	48.0	35.0	68.7	10.6	34.2
MPDA	81.5	92.2	52.0	51.2	69.2	85.0	33.1	43.5
MPDA-P	56.1	81.1	35.4	46.5	40.9	69.7	17.3	35.3
PnPDA	86.5	**94.9**	<u>53.2</u>	49.4	75.5	**90.3**	35.1	42.4
PnPDA-P	55.2	87.5	36.5	43.4	44.7	80.5	21.6	34.6
HEAL	86.0	92.0	49.1	<u>81.3</u>	<u>78.9</u>	88.2	31.3	<u>73.9</u>
STAMP	54.5	77.0	26.4	38.2	44.8	70.8	13.4	28.6
PolyInter	85.1	87.5	47.8	52.4	70.1	77.5	<u>37.5</u>	39.9
NegoCollab	<u>87.2</u>	91.1	51.2	74.5	76.5	85.4	31.9	55.5
GeoPA (**Ours**)	**88.9**	<u>92.4</u>	**60.3**	**85.9**	**81.9**	**90.3**	**40.9**	**79.5**

4 Experiments

4.1 Experimental Settings

Datasets and metrics. We evaluate GeoDP on OPV2V (Xu et al., 2022c) and DAIR-V2X (Yu et al., 2022). OPV2V is a simulated dataset for vehicle-to-vehicle collaborative perception, while DAIR-V2X contains real-world vehicle-infrastructure scenarios. We follow the standard data splits and preprocessing protocols provided by OpenCOOD. Following prior collaborative perception

Table 2: Quantitative comparison under seen-model settings on DAIR-V2X.

Method	AP@0.5			AP@0.7		
	m1m2	m1m3	m2m1	m1m2	m1m3	m2m1
No Fusion	67.2	68.9	1.5	<u>55.3</u>	57.8	0.4
MPDA	65.7	69.1	12.7	54.4	57.2	1.8
MPDA-P	62.4	64.3	2.1	53.2	53.9	0.4
PnPDA	62.5	69.7	22.4	45.3	54.7	6.1
PnPDA-P	65.4	70.9	20.3	52.2	59.6	6.8
HEAL	<u>69.6</u>	<u>74.7</u>	20.1	55.1	59.7	5.3
STAMP	63.2	64.4	2.3	54.1	54.3	0.5
PolyInter	69.2	73.8	<u>23.2</u>	54.7	<u>60.8</u>	<u>7.0</u>
NegoCollab	68.1	70.4	9.7	53.6	54.2	2.8
GeoPA (**Ours**)	**71.8**	**77.9**	**25.6**	**57.3**	**63.5**	**10.3**

Table 3: Zero-shot adaptation performance on OPV2V. Models m5 and m6 are unseen during training and evaluated without additional training or fine-tuning.

Method	AP@0.5			AP@0.7		
	m1m6	m1m5	m2m6	m1m6	m1m5	m2m6
No Fusion	45.8	<u>74.9</u>	21.8	30.6	66.8	10.9
MPDA	59.4	65.6	32.6	40.9	56.4	12.9
MPDA-P	49.2	65.3	26.7	35.6	57.3	11.9
PnPDA	62.4	71.9	<u>39.9</u>	47.0	<u>67.0</u>	20.8
PnPDA-P	42.5	68.2	25.7	33.5	58.8	16.3
HEAL	<u>75.9</u>	70.4	24.0	<u>66.7</u>	64.3	13.6
STAMP	33.9	54.9	21.7	20.8	40.2	10.1
PolyInter	46.1	71.4	28.1	35.2	60.3	19.5
NegoCollab	47.2	66.0	32.2	37.3	60.7	<u>21.2</u>
GeoPA (**Ours**)	**87.4**	**90.2**	**54.2**	**75.7**	**76.9**	**31.3**

studies, we evaluate vehicle detection using average precision at intersection-over-union thresholds of 0.5 and 0.7, denoted as AP@0.5 and AP@0.7.

Heterogeneous settings. We construct heterogeneous settings using six perception models on both datasets. Models m1–m4 participate in GeoDP training and are treated as seen models, whereas m5 and m6 are withheld for zero-shot evaluation. The LiDAR models are m1 with PointPillars (Lang et al., 2019), m3 with SECOND (Yan et al., 2018), and m5 with VoxelNet (Zhou & Tuzel, 2018). The camera models are m2 with LSS-EfficientNet (Tan & Le, 2019; Philion & Fidler, 2020), m4 with LSS-ResNet-101, and m6 with LSS-ResNet-34 (He et al., 2016). Complete configurations are provided in Appendix A.2.

Implementation details. GeoDP is implemented in PyTorch based on OpenCOOD. Each perception model is independently trained for 3D object detection and then frozen as a feature encoder. GeoDP is trained using the objectives described in Sec. 3.5, while the geometry foundation model remains frozen during geometric reference construction. We use the Adam optimizer (Kingma & Ba, 2014) throughout training. All compared methods use the same inputs, detection range, and base perception models. The optimization schedule, parameter-freezing settings, and other training configurations are detailed in Appendix A.3.

4.2 Main Results

Seen-model settings. Tables 1 and 2 compare seen-model performance on OPV2V and DAIR-V2X, respectively. On OPV2V, "all" denotes joint collaboration among m1–m4; on DAIR-V2X, the pair order specifies the vehicle-side ego and infrastructure-side models, respectively. We compare

Table 4: Zero-shot adaptation performance on DAIR-V2X.

Method	AP@0.5			AP@0.7		
	m1m6	m1m5	m2m5	m1m6	m1m5	m2m5
No Fusion	62.5	65.1	1.2	44.8	46.9	0.3
MPDA	58.9	65.4	1.5	48.2	<u>55.7</u>	0.3
MPDA-P	56.4	60.3	1.9	44.9	52.4	0.2
PnPDA	54.1	<u>66.8</u>	1.7	35.1	53.9	0.4
PnPDA-P	58.6	61.0	1.3	49.6	51.9	0.2
HEAL	59.6	61.1	1.3	49.1	51.6	0.2
STAMP	53.5	62.6	1.1	44.1	50.3	0.2
PolyInter	61.7	62.9	<u>2.7</u>	50.8	51.6	<u>0.6</u>
NegoCollab	<u>63.3</u>	63.5	2.4	<u>52.9</u>	53.1	0.4
GeoPA (**Ours**)	**65.7**	**67.6**	**6.8**	**57.3**	**56.4**	**1.4**

Table 5: Components ablation of GeoDP.

Components			Detection performance	
GUS	DPA	ASF	AP@0.5	AP@0.7
–	–	–	79.4	69.8
✓	–	–	82.0	75.4
–	✓	–	83.2	76.1
✓	✓	–	86.8	79.3
✓	✓	✓	**88.9**	**81.9**

Table 6: Prompt-strategy ablation study of Dynamic Prompt Adapter (DPA).

DPA design			Detection performance	
Dynamic	Gap	Interaction	AP@0.5	AP@0.7
–	–	–	82.9	76.0
✓	–	–	84.4	77.9
✓	✓	–	86.1	79.1
✓	✓	✓	**88.9**	**81.9**

GeoDP with No Fusion, MPDA (Xu et al., 2022a), PnPDA (Luo et al., 2024), HEAL (Lu et al., 2024), STAMP (Gao et al., 2025), PolyInter (Xia et al., 2025), and NegoCollab (Shao et al., 2026). Following NegoCollab, we also evaluate the protocol-space variants MPDA-P and PnPDA-P, where P denotes a protocol agent that adopts the same perception configuration as m1 and provides the common representation space.

As shown in Table 1, GeoDP achieves the best performance under the m1m2, m2m4, and all-model settings on OPV2V, while remaining competitive under m1m3. As shown in Table 2, GeoDP achieves the best performance across all model combinations and evaluation metrics on DAIR-V2X. These results demonstrate the consistent effectiveness of GeoDP across datasets and heterogeneous model combinations.

Zero-shot adaptation. Tables 3 and 4 evaluate zero-shot model adaptation on OPV2V and DAIR-V2X, respectively. Models m5 and m6 are excluded from all GeoDP training stages. At test time, their frozen local features are adapted by reusing the learned projection branches through geometry-anchor-based routing, without optimizing any GeoDP parameter.

As shown in Tables 3 and 4, GeoDP consistently outperforms these heterogeneous collaboration methods across all unseen model combinations on both datasets. These results demonstrate effective generalization across unseen models, while the cross-dataset transferability of GeoDP is further evaluated in Appendix C.5.

4.3 Ablation studies

Component-wise ablation. We evaluate the contributions of GUS, DPA, and ASF on OPV2V under the m1m2 setting. The baseline maps heterogeneous features into a predefined reference space using a static adapter, followed by fixed weighted fusion. For controlled comparison, GUS replaces the predefined reference space with the geometry-anchored unified space, whereas DPA replaces static mapping with dynamic prompt adaptation. As shown in Table 5, either GUS or DPA improves the baseline, and their combination yields further gains, demonstrating their complementary roles in feature unification and adaptation. Adding ASF achieves the best performance, validating the complete GeoDP design.

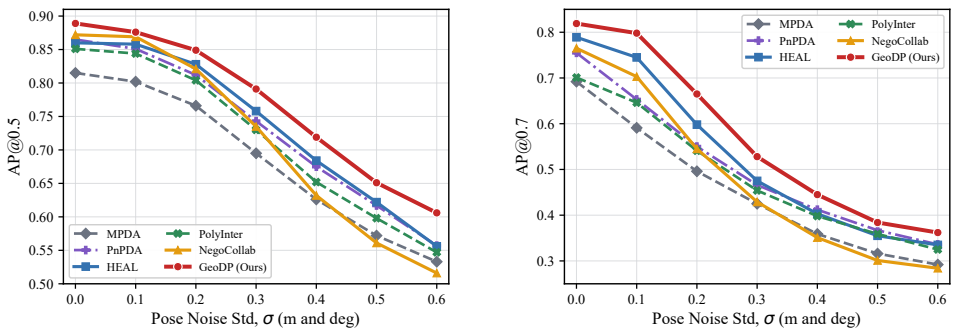

Figure 4: Robustness comparison under different pose-noise levels on OPV2V in the m1m2 setting.

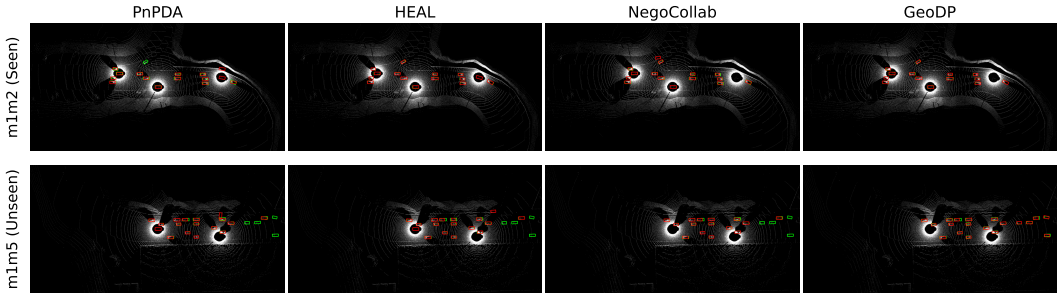

Figure 5: Qualitative detection results for m1m2 and m1m5 on OPV2V. Green boxes denote ground-truth objects, and red boxes denote predictions.

Prompt strategy ablation. We evaluate DPA variants on OPV2V under the m1m2 setting, with all other settings fixed. In Table 6, the all-dash row denotes the static-prompt baseline. Input-conditioned prompts improve performance; geometry-gap modeling brings further gains, and element-wise interaction yields the best results, highlighting the complementary roles of difference and interaction cues.

Pose-noise robustness. We introduce Gaussian noise of varying magnitudes into agent poses under the m1m2 setting on OPV2V to evaluate the robustness of different methods. As shown in Fig. 4, GeoDP consistently achieves the best performance on both AP@0.5 and AP@0.7 across all evaluated noise levels, demonstrating its robustness to pose perturbations.

4.4 Visualization

As shown in Fig. 5, we present qualitative detection results on OPV2V under the seen-model m1m2 and unseen-model m1m5 settings. Compared with existing methods, GeoDP's predicted boxes more closely match the ground-truth objects. In particular, GeoDP maintains good localization even when collaborating with a perception model unseen during training.

5 Conclusion

In this paper, we presented GeoDP for heterogeneous collaborative perception. GUS provides a model-independent geometric reference, DPA adapts heterogeneous features through input-conditioned prompts, and ASF aggregates them according to their spatial reliability. The stored geometry anchor supports adaptation to unseen models without running the geometry foundation model or fine-tuning GeoDP. Across simulated and real-world datasets, GeoDP achieves strong overall performance relative to existing methods in seen and unseen model settings; it also remains robust to pose noise and message dropout, with promising cross-dataset transfer.

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 4.1 and Appendix A.1 describe the datasets, data splits, detection ranges, and evaluation protocols. Table 7 specifies the local perception models, while Appendix A.3 provides the hardware, training settings, hyperparameters, and inference procedure. The module designs and optimization objectives are detailed in Sections 3.2–3.5 and Appendix B.1–B.4, including the procedure for unseen-model evaluation. Together, these details support independent reproduction of the reported results. 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.

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

Xiangbo Gao, Runsheng Xu, Jiachen Li, Ziran Wang, Zhiwen Fan, and Zhengzhong Tu. Stamp: Scalable task-and model-agnostic collaborative perception. In *International Conference on Learning Representations*, volume 2025, pp. 54656–54676, 2025.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

Yushan Han, Hui Zhang, Honglei Zhang, Chuntao Ding, Yuanzhouhan Cao, and Yidong Li. Cods: Enhancing collaborative perception in heterogeneous scenarios via domain separation. *IEEE Transactions on Mobile Computing*, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pp. 709–727. Springer, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 12689–12697. IEEE, 2019.

Aoxue Li, Peng Yuan, and Zhenguo Li. Semi-supervised object detection via multi-instance alignment with global class prototypes. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9799–9808. IEEE, 2022a.

Yiming Li, Shunli Ren, Pengxiang Wu, Siheng Chen, Chen Feng, and Wenjun Zhang. Learning distilled collaboration graph for multi-agent perception. *Advances in Neural Information Processing Systems*, 34:29541–29552, 2021.

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, 2022b.

Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE international conference on robotics and automation (ICRA)*, pp. 2774–2781. ieee, 2023.

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*, volume 2024, pp. 44576–44593, 2024.

Tianyou Luo, Quan Yuan, Guiyang Luo, Yuchen Xia, Yujia Yang, and Jinglin Li. Plug and play: A representation enhanced domain adapter for collaborative perception. In *European Conference on Computer Vision*, pp. 287–303. Springer, 2024.

Armin Maleki and Hayder Radha. Pearl: A lightweight prompt-based feature interpreter framework for real-time, anonymous, and heterogeneous collaborative perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1138–1147, 2026.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European conference on computer vision*, pp. 194–210. Springer, 2020.

Congzhang Shao, Quan Yuan, Guiyang Luo, Yue Hu, Danni Wang, Liu Yilin, Rui Pan, Bo Chen, and Jinglin Li. Negocollab: A common representation negotiation approach for heterogeneous collaborative perception. *Advances in Neural Information Processing Systems*, 38:87196–87217, 2026.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PmLR, 2019.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Sven Teufel, Jörg Gamerdinger, Georg Volk, and Oliver Bringmann. Collective pv-rcnn: A novel fusion technique using collective detections for enhanced local lidar-based perception. In *2023 IEEE 26th International conference on intelligent transportation systems (ITSC)*, pp. 1828–1834. IEEE, 2023.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 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. Springer, 2020.

Wentao Wang, Haoran Xu, and Guang Tan. Gt-space: Enhancing heterogeneous collaborative perception with ground truth feature space. *arXiv preprint arXiv:2603.19308*, 2026.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Yuchen Xia, Quan Yuan, Guiyang Luo, Xiaoyuan Fu, Yang Li, Xuanhan Zhu, Tianyou Luo, Siheng Chen, and Jinglin Li. One is plenty: A polymorphic feature interpreter for immutable heterogeneous collaborative perception. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1592–1601. IEEE, 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, Yi Guo, Xu Han, Xin Xia, Hao Xiang, and Jiaqi Ma. Opencda: an open cooperative driving automation framework integrated with co-simulation. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 1155–1162. IEEE, 2021.

Runsheng Xu, Jinlong Li, Xiaoyu Dong, Hongkai Yu, and Jiaqi Ma. Bridging the domain gap for multi-agent perception. *arXiv preprint arXiv:2210.08451*, 2022a.

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. Springer, 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, Weizhe Chen, Hao Xiang, Xin Xia, Lantao Liu, and Jiaqi Ma. Model-agnostic multi-agent perception framework. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1471–1478. IEEE, 2023a.

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, 2023b.

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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21329–21338. IEEE, 2022.

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

A More Implementation Details

A.1 Datasets and Evaluation Protocols

We conduct experiments on OPV2V (Xu et al., 2022c) and DAIR-V2X (Yu et al., 2022). OPV2V is a large-scale simulated V2V collaborative perception dataset generated using CARLA (Dosovitskiy et al., 2017) and OpenCDA (Xu et al., 2021). It contains 73 diverse scenes, 11,464 synchronized frames of LiDAR point clouds and multi-view images, and 232,913 annotated 3D vehicle bounding boxes. Each scene involves 2–7 collaborative agents. We follow the standard training, validation, and test splits provided by OpenCOOD and retain up to five agents within a communication range of $70\,\mathrm{m}$ for each sample. The detection range on OPV2V is set to $x \in [-102.4, 102.4]\,\mathrm{m}$, $y \in [-51.2, 51.2]\,\mathrm{m}$, and $z \in [-3, 1]\,\mathrm{m}$.

DAIR-V2X is a real-world multimodal dataset for vehicle–infrastructure collaborative perception. We use its cooperative subset, DAIR-V2X-C, in which each sample provides calibrated vehicle-side and infrastructure-side LiDAR and camera observations together with cooperative 3D annotations. For experiments involving camera agents, we retain samples from the official training and validation splits for which the images, point clouds, calibration parameters, and annotations are complete. The same filtered splits are used for all compared methods. Each sample contains one vehicle-side agent and one infrastructure-side agent, with the communication range set to $100\,\mathrm{m}$. The detection range is $x \in [-102.4, 102.4]\,\mathrm{m}$, $y \in [-51.2, 51.2]\,\mathrm{m}$, and $z \in [-3.5, 1.5]\,\mathrm{m}$.

Following the OpenCOOD evaluation protocol, we compute average precision for vehicle detection by matching predicted and ground-truth boxes according to their BEV intersection over union. We report AP at IoU thresholds of 0.5 and 0.7, denoted as $\mathrm{AP@0.5}$ and $\mathrm{AP@0.7}$, respectively. Under the seen-model setting, perception models participating in training are evaluated at test time. Under the zero-shot adaptation setting, unseen models are excluded from GeoDP training and evaluated without additional training or fine-tuning. For the pose-noise experiment, zero-mean Gaussian noise is added to the x- and y-translations and yaw angle of the agent poses, with standard deviations of $\sigma\,\mathrm{m}$ and σ°, respectively.

A.2 Model Configurations

As a complement to the heterogeneous model settings introduced in Section 4.1, Table 7 summarizes the detailed configurations of the six perception models used on OPV2V and DAIR-V2X, together with the protocol agent P. Models m1–m4 are treated as seen models and participate in GeoDP training, whereas m5 and m6 are used only for zero-shot evaluation. For MPDA-P and PnPDA-P, P denotes the protocol agent, whose perception configuration is identical to m1 and whose representation serves as the common representation space. The local perception networks of m5 and m6 are independently trained on the corresponding datasets, while their features are never used to optimize GeoDP, and no additional training or fine-tuning is performed during evaluation.

Table 7: Configurations of heterogeneous perception models on OPV2V and DAIR-V2X. Voxel and pillar sizes are reported in the order of x, y, and z. For PointPillars, the vertical pillar size spans the entire detection height. Models m1–m4 are seen during GeoDP training, whereas m5 and m6 are used only for zero-shot adaptation evaluation.

Model ID	**Sensor**	**Encoder/Backbone**	**Voxel size on OPV2V (m)**	**Voxel size on DAIR-V2X (m)**
P	LiDAR	PointPillars	$[0.4, 0.4, 4.0]$	$[0.4, 0.4, 5.0]$
m1	LiDAR	PointPillars	$[0.4, 0.4, 4.0]$	$[0.4, 0.4, 5.0]$
m2	Camera	LSS (EfficientNet-B0)	–	–
m3	LiDAR	SECOND	$[0.1, 0.1, 0.1]$	$[0.1, 0.1, 0.1]$
m4	Camera	LSS (ResNet-101)	–	–
m5	LiDAR	VoxelNet	$[0.4, 0.4, 0.4]$	$[0.4, 0.4, 0.5]$
m6	Camera	LSS (ResNet-34)	–	–

A.3 Training and Inference Settings

GeoDP is implemented in PyTorch based on OpenCOOD, and all experiments are conducted on two NVIDIA A40 GPUs. Before training GeoDP, each local perception model listed in Table 7 is independently trained on the corresponding dataset and subsequently used as a frozen feature encoder. GeoDP is trained in three stages using the same schedule on OPV2V and DAIR-V2X. In the first stage, a frozen VGGT-1B is used to train GUS and the unified detection head while updating the stored geometry anchor through EMA for 20 epochs. In the second stage, the local perception encoders, GUS, and unified detection head are frozen, and only DPA is optimized for 12 epochs. In the third stage, DPA is additionally frozen, and only ASF is optimized for 4 epochs.

We use the Adam optimizer with a per-GPU batch size of 1, an initial learning rate of 5×10^{-4}, and a weight decay of 10^{-4} throughout training. A multi-step learning-rate schedule is adopted with a decay factor of 0.1. The EMA momentum for the geometry anchor is set to 0.99, and the softmax temperature in ASF is set to 1.5. During DPA training, the weights of the detection, feature alignment, descriptor classification, and prompt regularization losses are set to 1.0, 1.0, 0.1, and 0.001, respectively. During ASF training, the weights of the detection, feature alignment, weight balance, and entropy regularization losses are set to 1.0, 0.5, 0.01, and 0.01, respectively.

During online inference, only the local perception encoders, DPA, ASF, stored geometry anchor, and unified detection head are retained. Features produced by different agents are mapped into the unified space by DPA, spatially aggregated by ASF, and decoded by the unified detection head. VGGT and the GUS construction branch are used only for offline anchor construction and training supervision and are not executed during online inference. For zero-shot collaboration settings involving m5 or m6, all GeoDP parameters remain frozen, without additional training or fine-tuning.

B Additional Method Details

B.1 Detailed Construction of GUS

As illustrated in Fig. 3(a), GUS is constructed offline from calibrated multi-view images. Its geometry-aware encoding and anchor update follow Eqs. 1–3 in the main text. Specifically, the depth predicted by frozen VGGT is converted to metric scale and divided into D non-uniform bins using linear-increasing discretization. The representative depth of the k-th bin is

$$d_k = d_{\min} + \frac{d_{\max} - d_{\min}}{D(D+1)} k(k+1), \qquad k = 0, \ldots, D-1.$$

The discretized depth lifts image features into the camera frustums, which are projected onto a shared BEV grid using the camera parameters. Voxel pooling and BEV encoding then produce the reference feature $\mathbf{F}^G$, as formulated in Eq. 2. The unified detection head supervises $\mathbf{F}^G$ to preserve stable spatial structures and task-relevant semantics.

The geometry descriptor encoder subsequently encodes and normalizes $\mathbf{F}^G$ into $\mathbf{z}_G$. As defined in Eq. 3, the batch descriptors update the stored geometry anchor $\bar{\mathbf{z}}_G$ through EMA with a momentum of 0.99, initialized by the first batch descriptor. During DPA training, the current descriptor $\mathbf{z}_G$ provides sample-specific geometric guidance. At inference, only $\bar{\mathbf{z}}_G$ is retained, without executing VGGT or the GUS construction branch.

B.2 Detailed Design of DPA

DPA architecture. The overall architecture of the Dynamic Prompt Adapter (DPA) is illustrated in Fig. 3(b), while its descriptor extraction, prompt generation, and feature modulation follow Eqs. 4–7 in the main text. Given an agent feature $\mathbf{F}_i$, DPA first applies an input projection comprising a 1×1 convolution, group normalization, and ReLU to unify its channel dimension to 256. Except for these lightweight input projections, the descriptor encoder, Prompt Router, and feature modulation network are shared across agents.

The descriptor encoder extracts the channel-wise mean, standard deviation, global response, and cross-channel correlation from the projected feature $\mathbf{X}_i$. To compute the correlation, $\mathbf{X}_i$ is pooled to

a fixed spatial size and reshaped as $\widetilde{\mathbf{X}}_i \in \mathbb{R}^{C\times S}$, whose Gram representation is

$$\mathbf{G}_i = \frac{1}{S}\widetilde{\mathbf{X}}_i\widetilde{\mathbf{X}}_i^\top. \tag{12}$$

These statistics are aggregated and normalized into the 16-dimensional descriptor $\mathbf{z}_i$ following Eq. 4. During training, an auxiliary descriptor-classification head improves its discrimination among different feature distributions and is discarded at inference. The Prompt Router then models the difference and element-wise interaction between $\mathbf{z}_i$ and the reference descriptor as defined in Eq. 5, and generates a 64-dimensional prompt $\mathbf{p}_i$ through a two-layer MLP. Following Eqs. 6 and 7, the prompt produces the FiLM parameters for channel-wise modulation, yielding the unified-space feature $\widehat{\mathbf{F}}_i$.

Zero-shot inference for unseen models. The local perception network of an unseen model is trained independently on the corresponding dataset, but neither its features nor labels participate in GeoDP training. Given its feature $\mathbf{F}_u$, DPA selects the input projections learned from seen models whose input channel dimensions are compatible with $\mathbf{F}_u$, denoted by $\mathcal{K}_u$. The k-th branch follows Eqs. 4–7 to generate a descriptor $\mathbf{z}_u^{(k)}$ and an adapted candidate $\widehat{\mathbf{F}}_u^{(k)}$. Its distance from the stored geometry anchor is defined as

$$q_u^{(k)} = \frac{1}{d_z}\left\|\mathbf{z}_u^{(k)} - \bar{\mathbf{z}}_G\right\|_2^2, \tag{13}$$

where d_z is the descriptor dimension. The routing weights and aggregated feature are computed as

$$\alpha_u^{(k)} = \frac{\exp\left(-q_u^{(k)}/\tau_u\right)}{\sum_{j\in\mathcal{K}_u}\exp\left(-q_u^{(j)}/\tau_u\right)}, \qquad \widehat{\mathbf{F}}_u = \sum_{k\in\mathcal{K}_u}\alpha_u^{(k)}\widehat{\mathbf{F}}_u^{(k)}. \tag{14}$$

Candidate routing is activated only during zero-shot inference, so no routing temperature is involved in training. At inference, τ_u is fixed to 1.0, and both the local perception networks and all GeoDP parameters remain frozen. The aggregated feature $\widehat{\mathbf{F}}_u$ is assigned to an available ASF input slot, requiring neither a model-specific adapter nor any parameter update for the unseen model.

The current zero-shot mechanism assumes that the unseen feature is compatible with at least one learned input projection, i.e., $\mathcal{K}_u \neq \varnothing$. If an unseen model produces a channel dimension unsupported during training, an additional channel transformation or input projection is required, increasing the integration and deployment complexity for new models. Direct adaptation across arbitrary channel dimensions therefore remains a limitation of the current design and motivates future dimension-agnostic projection mechanisms.

B.3 Detailed Design of ASF

Adaptive Spatial Fusion (ASF) receives the unified-space features produced by DPA, following the overall formulation in Eqs. 8–10 of the main text. Before fusion, all agent features are transformed into the ego coordinate system and resized to the spatial resolution defined by GUS. The aligned features are then concatenated along the channel dimension and passed to the spatial weight predictor.

The weight predictor contains two 3×3 convolutional layers with 128 channels, each followed by group normalization and ReLU. A final 1×1 convolution generates a spatial score map S_i for each agent. Following Eq. 9, these scores are normalized across agents using a softmax temperature τ, which is set to 1.5 during both training and inference.

To support a varying number of collaborative agents, ASF uses a binary variable $m_i \in \{0,1\}$ to indicate whether the i-th input slot is available. An unavailable slot is filled with zeros and masked before softmax normalization. The resulting weight is

$$W_i(h,w) = \frac{m_i\exp\left(S_i(h,w)/\tau\right)}{\sum_{j=1}^{N} m_j\exp\left(S_j(h,w)/\tau\right)}. \tag{15}$$

The weights are therefore distributed only among the available agents without changing the ASF architecture. The fused feature is computed using Eq. 10 and passed to the unified detection head. During this stage, GUS, the local perception encoders, DPA, and the unified detection head remain frozen, and only ASF is optimized. Its training objective is provided in Appendix B.4. At online inference, ASF operates solely on the adapted DPA outputs and requires neither VGGT, the reference feature $\mathbf{F}^G$, nor the GUS construction branch.

B.4 Complete Optimization Objectives

GeoDP is trained using the three-stage strategy described in Sec. 3.5. Each stage employs the 3D detection loss $\mathcal{L}_{\text{det}}$, which consists of classification, bounding-box regression, and direction-classification terms. The corresponding loss weights are provided in Appendix A.3.

In the first stage, GUS and the unified detection head are optimized using detection supervision together with a multi-scale occupancy objective:

$$\mathcal{L}_{\text{GUS}} = \mathcal{L}_{\text{det}} + \mathcal{L}_{\text{occ}}. \tag{16}$$

Here, $\mathcal{L}_{\text{occ}}$ strengthens foreground responses through occupancy supervision at multiple feature scales. Since the metric depth predicted by VGGT is directly discretized and used in the LSS view transformation, no additional depth-classification loss is applied.

In the second stage, only DPA is optimized. For the M input feature types observed during training, its objective is

$$\mathcal{L}_{\text{DPA}} = \frac{1}{M}\sum_{i=1}^{M}\left(\lambda_{\text{det}}\mathcal{L}_{\text{det}}^{(i)} + \lambda_{\text{feat}}\mathcal{L}_{\text{feat}}^{(i)} + \lambda_{\text{desc}}\mathcal{L}_{\text{desc}}^{(i)} + \lambda_{\text{prompt}}\mathcal{L}_{\text{prompt}}^{(i)}\right). \tag{17}$$

The feature-alignment term is defined as

$$\mathcal{L}_{\text{feat}}^{(i)} = \text{SmoothL1}\left(\widehat{\mathbf{F}}_i, \text{sg}\big(\mathbf{F}^G\big)\right), \tag{18}$$

where $\text{sg}(\cdot)$ denotes stop-gradient. The auxiliary descriptor-classification loss $\mathcal{L}_{\text{desc}}$ distinguishes the feature distributions observed during training, while $\mathcal{L}_{\text{prompt}}$ applies mean-squared regularization to the dynamic prompt $\mathbf{p}_i$. Both auxiliary objectives are used only during training and introduce no online inference overhead.

In the third stage, only ASF is optimized using

$$\mathcal{L}_{\text{ASF}} = \lambda_{\text{det}}\mathcal{L}_{\text{det}} + \lambda_{\text{feat}}\mathcal{L}_{\text{feat}} + \lambda_{\text{bal}}\mathcal{L}_{\text{bal}} + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}}. \tag{19}$$

Here, $\mathcal{L}_{\text{feat}}$ aligns the fused feature $\mathbf{F}_{\text{fuse}}$ with the stop-gradient reference feature $\mathbf{F}^G$ using Smooth-L1 loss. The balance term $\mathcal{L}_{\text{bal}}$ prevents a single input from dominating the average fusion weights, while the entropy term $\mathcal{L}_{\text{ent}}$ discourages premature concentration of spatial weights during training. All reference features used for feature supervision are detached, preserving the established unified feature space throughout DPA and ASF training.

C Additional Experimental Results

C.1 Efficiency Analysis

We compare the online computational efficiency of different methods under the DAIR-V2X m1m2 setting. All methods are evaluated on a single NVIDIA A40 GPU. Online parameters include only those executed during inference. Inference latency is measured from GPU-resident preprocessed sensor inputs to the raw predictions of the detection head, excluding data loading, device transfer, and post-processing. The latency is averaged over 200 runs after 50 warm-up iterations. We additionally evaluate GeoDP-Online, which executes VGGT and the GUS reference branch during inference, to examine the overhead of online geometry modeling.

As shown in Table 8, the complete GeoDP contains only 20.053M online parameters, which is comparable to existing heterogeneous collaborative perception methods, while achieving the lowest inference latency. In contrast, GeoDP-Online requires 1241.329M parameters because it executes

Table 8: Comparison of online parameter count and inference latency under the DAIR-V2X m1m2 setting. GeoDP-Online executes VGGT and the GUS reference branch during inference, whereas GeoDP uses the stored geometry anchor.

Method	Online params (M)	Inference latency (ms)	AP@0.5
MPDA	**18.763**	118.189	65.7
PnPDA	19.001	147.483	62.5
HEAL	18.809	119.023	69.6
STAMP	19.053	125.538	63.2
PolyInter	21.397	173.948	69.2
NegoCollab	19.078	144.679	68.1
GeoDP-Online	1241.329	317.571	**72.2**
GeoDP (**Ours**)	20.053	**116.842**	71.8

VGGT and the GUS reference branch online. By storing the geometry anchor and removing the online reference branch, GeoDP reduces the online parameter count by 1221.276M, corresponding to a reduction of approximately 98.4%, while substantially decreasing inference latency. These results show that DPA and ASF introduce only limited overhead, enabling both geometry-guided heterogeneous feature adaptation and efficient online inference.

C.2 Effectiveness of VGGT Geometry Guidance

To evaluate the quality of the geometric information used to construct GUS, we compare VGGT depth with the expected depth produced by the LSS depth distribution. On OPV2V, we select 50 scenes from the training data used for GUS construction, covering 200 camera views, and use the dataset-provided depth as reference. On DAIR-V2X, we evaluate 50 vehicle-side camera views using projected LiDAR depth as reference. All metrics are computed only at valid reference-depth locations. Fixed depth scales of 12 and 20.25 are used for OPV2V and DAIR-V2X, respectively. We report absolute relative error (AbsRel), mean absolute error (MAE), root mean square error (RMSE), and threshold accuracy δ_1.

As shown in Table 9, VGGT produces consistently lower depth errors on both datasets. On OPV2V, it reduces AbsRel from 0.805 to 0.351 and MAE from 6.928 to 4.113. On DAIR-V2X, AbsRel and MAE decrease from 3.743 and 24.844 to 0.389 and 8.062, respectively. VGGT also achieves higher δ_1 on both datasets, indicating better consistency with the reference geometry.

Table 9: Quantitative comparison of LSS and VGGT depth estimates used for GUS construction.

Dataset	Depth source	AbsRel ↓	MAE ↓	RMSE ↓	δ_1 ↑
OPV2V	LSS Depth	0.805	6.928	11.030	0.164
	VGGT Depth	**0.351**	**4.113**	**6.853**	**0.263**
DAIR-V2X	LSS Depth	3.743	24.844	33.649	0.192
	VGGT Depth	**0.389**	**8.062**	**13.318**	**0.314**

Fig. 6 presents qualitative results on OPV2V. Compared with LSS depth, VGGT depth preserves clearer road structures, object boundaries, and foreground–background relationships, while exhibiting better spatial agreement with the reference depth. These results demonstrate that VGGT provides more reliable geometric information for GUS construction. Together with the component-wise ablation results in the main paper, the comparison shows that accurate geometry guidance contributes to establishing a stable unified feature space.

C.3 Feature-Space Analysis

To further examine the alignment capability of DPA, we compare the input-projected and DPA-adapted features with the unified-space reference feature on OPV2V using Smooth-L1 distance and cosine similarity. Table 10 reports the results for the seen models m2 and m3, as well as the unseen

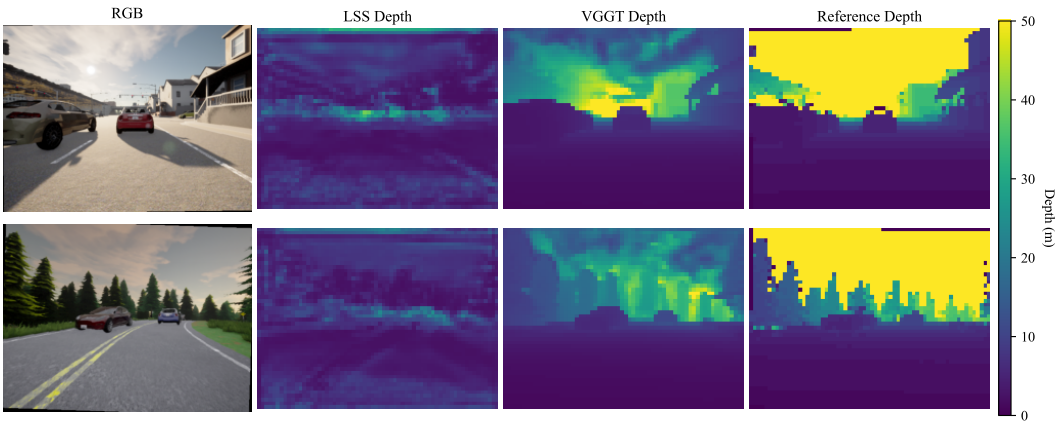

Figure 6: Qualitative comparison of depth estimates used for GUS construction on OPV2V. From left to right: RGB image, LSS depth, VGGT depth, and reference depth.

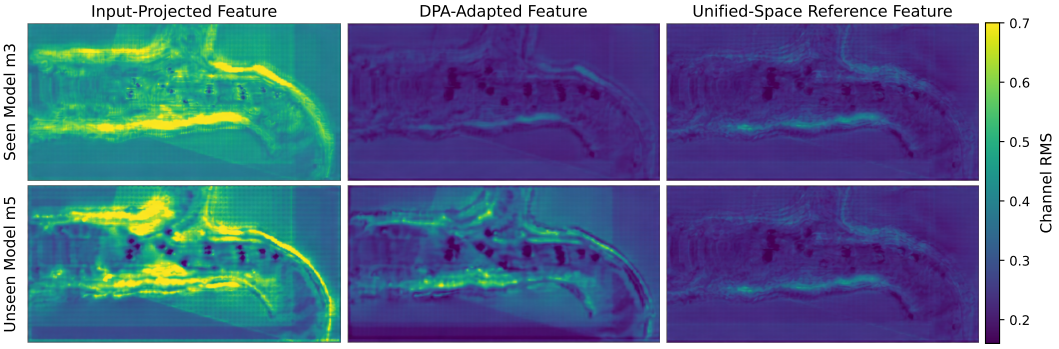

Figure 7: Visualization of BEV feature responses before and after DPA on the same OPV2V scene. Input-projected, DPA-adapted, and unified-space reference features are shown for a representative seen model m3 and unseen model m5.

models m5 and m6. After DPA, all four models achieve consistently lower feature distances and higher cosine similarities. The adapted features of the seen models exhibit strong agreement with the reference feature. DPA also substantially reduces the feature gaps of m5 and m6, demonstrating its ability to adapt previously unseen models.

Table 10: Feature-space alignment results before and after DPA. m2 and m3 are seen models, while m5 and m6 are unseen models.

Model		**Smooth-L_1 distance**		**Cosine similarity**	
Type	**ID**	**Before DPA**	**After DPA**	**Before DPA**	**After DPA**
Seen	m2	0.203	**0.004**	0.101	**0.938**
	m3	0.107	**0.003**	0.024	**0.942**
Unseen	m5	0.098	**0.015**	0.045	**0.802**
	m6	0.161	**0.036**	0.108	**0.741**

Fig. 7 further visualizes the BEV feature responses of a representative seen model m3 and unseen model m5. Before adaptation, their response magnitudes and spatial structures differ noticeably from the unified-space reference feature. After DPA, the responses over major road structures and foreground regions become more consistent with the reference feature. Together, the quantitative and

qualitative results demonstrate that DPA effectively maps heterogeneous features from both seen and unseen models into the geometry-anchored unified space.

C.4 Sensitivity Analysis of ASF Temperature

During ASF training, the temperature is set to $\tau = 1.5$. To evaluate its sensitivity at inference, we keep all trained model parameters fixed and vary τ among 0.5, 1.5, and 3.0 under the m1m2 setting on OPV2V and DAIR-V2X. As shown in Table 11, a lower temperature produces more discriminative spatial weights and generally yields better detection performance, whereas a larger temperature results in overly smooth weight distributions. These results indicate that the temperature controls the selectivity of ASF without changing its learned parameters. For consistency between training and inference, all main experiments use $\tau = 1.5$.

Table 11: Sensitivity analysis of the ASF temperature on OPV2V and DAIR-V2X under the m1m2 setting.

Dataset	Temperature	AP@0.5	AP@0.7
OPV2V	0.5	89.4	84.0
	1.5	88.9	81.9
	3.0	86.8	82.4
DAIR-V2X	0.5	72.3	59.0
	1.5	71.8	57.3
	3.0	68.2	56.5

C.5 Cross-Dataset Direct Transfer

To further evaluate the robustness of heterogeneous collaborative perception models to dataset distribution shifts, we conduct a cross-dataset direct transfer experiment from DAIR-V2X to OPV2V. All experiments are performed under the m1m3 collaboration setting, with OPV2V fixed as the test dataset. For each method, we first report the in-domain performance obtained by training and testing on OPV2V as a reference. We then directly evaluate the model trained on DAIR-V2X using the OPV2V test set, without any additional training or fine-tuning on OPV2V.

During cross-dataset evaluation, the complete online inference parameters and feature-space configuration learned on DAIR-V2X are retained to avoid modifying the model due to incompatible perception ranges or feature grids. Specifically, the model follows the perception range, feature-grid configuration, and learned parameters associated with its training dataset, whereas the test samples, agent calibration, modality assignment, detection annotations, and evaluation pipeline are provided by OPV2V. For GeoDP, we directly load the complete model trained on DAIR-V2X, including the local perception branches, DPA, ASF, unified detection head, and stored geometry anchor. Inference is performed in the online mode without running VGGT. The compared methods similarly use their complete DAIR-V2X-trained models, with no target-dataset parameter adaptation.

Table 12: Cross-dataset direct transfer results from DAIR-V2X to OPV2V under the m1m3 setting. In-domain results obtained by training and testing on OPV2V are provided as references. Relative Drop measures the performance degradation from in-domain to cross-dataset evaluation, and the best cross-dataset results are highlighted in bold.

Method	AP@0.5			AP@0.7		
	In-domain	Cross-dataset	Drop (%) ↓	In-domain	Cross-dataset	Drop (%) ↓
MPDA	92.2	25.6	72.23	85.0	12.5	85.29
PnPDA	**94.9**	4.5	95.26	**90.3**	1.3	98.56
HEAL	92.0	<u>29.6</u>	<u>67.83</u>	88.2	<u>17.3</u>	<u>80.39</u>
STAMP	77.0	3.3	95.71	70.8	1.3	98.16
PolyInter	87.5	24.1	72.46	77.5	11.3	85.42
NegoCollab	91.1	2.4	97.37	85.4	1.2	98.59
GeoDP (**Ours**)	<u>92.4</u>	**80.5**	**12.88**	**90.3**	**63.9**	**29.24**

Table 12 reports both the in-domain and cross-dataset results. For each method, the relative performance drop is calculated as

$$\text{Relative Drop} = \frac{\text{AP}_{\text{in-domain}} - \text{AP}_{\text{cross-dataset}}}{\text{AP}_{\text{in-domain}}} \times 100\%. \tag{20}$$

The results show that existing heterogeneous collaboration methods suffer substantial degradation when transferred from DAIR-V2X to OPV2V, with relative drops of 67.83%–97.37% on AP@0.5 and 80.39%–98.59% on AP@0.7. This indicates that their learned feature mappings and collaboration mechanisms are strongly tied to the training distribution, resulting in limited cross-dataset generalization. In contrast, GeoDP achieves the best cross-dataset performance, with substantially smaller relative drops of 12.88% and 29.24%, respectively. These results provide strong evidence that the complete GeoDP framework preserves substantially better detection performance under the evaluated dataset shift, without target-dataset fine-tuning.

C.6 Robustness to Collaborator-Message Dropout

To evaluate robustness under unstable communication, we conduct a collaborator-message dropout experiment on OPV2V under the m2m4 setting. For each test sample, the complete feature message transmitted by the collaborating agent is randomly discarded with probability p, while the ego agent's local feature remains available. We consider $p \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$, where $p = 0$ represents normal collaboration and $p = 1$ indicates that all collaboration messages are unavailable. This experiment isolates message loss without introducing pose noise or other perturbations. All reported results are averaged over three random seeds.

As shown in Fig. 8, the performance of all methods decreases as the dropout probability increases because progressively less collaborative information is available. Nevertheless, GeoDP consistently achieves the highest AP@0.5 and AP@0.7 across all evaluated dropout probabilities. Even when all collaborator messages are removed at $p = 1.0$, GeoDP maintains the best performance among the compared methods in the resulting ego-only inference setting. These results demonstrate that GeoDP maintains reliable detection performance under intermittent or completely unavailable collaborative communication.

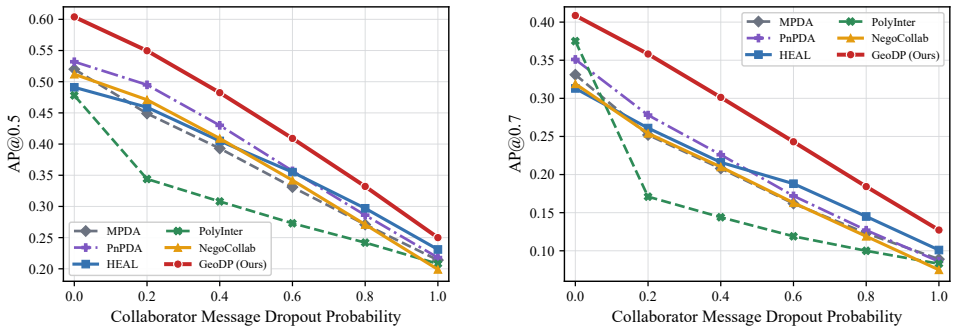

Figure 8: Robustness comparison under different collaborator-message dropout probabilities on OPV2V under the m2m4 setting. The left and right panels report AP@0.5 and AP@0.7, respectively. All results are averaged over three random seeds.

C.7 More Visualization

Fig. 9 and 10 present qualitative comparisons on OPV2V and DAIR-V2X under both seen- and unseen-model settings. On OPV2V, GeoDP produces accurate detections under the seen m1m3 and m2m4 configurations and the unseen m1m6 and m2m6 configurations. Similar results are observed on DAIR-V2X under the seen m1m2 and m1m3 configurations and the unseen m1m5 and m1m6 configurations. Compared with existing methods, GeoDP provides more consistent object localization and fewer spurious predictions, demonstrating its effectiveness across different heterogeneous configurations and its generalization to unseen perception models.

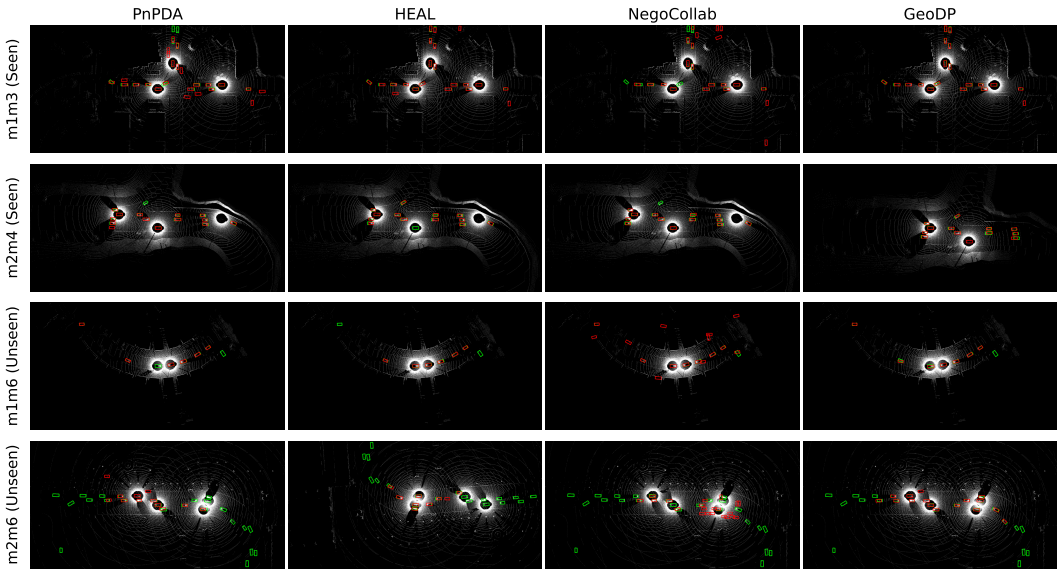

Figure 9: Qualitative detection results on OPV2V under seen- and unseen-model configurations. The four rows correspond to m1m3, m2m4, m1m6, and m2m6, respectively. Green boxes denote ground-truth objects, and red boxes denote predictions.

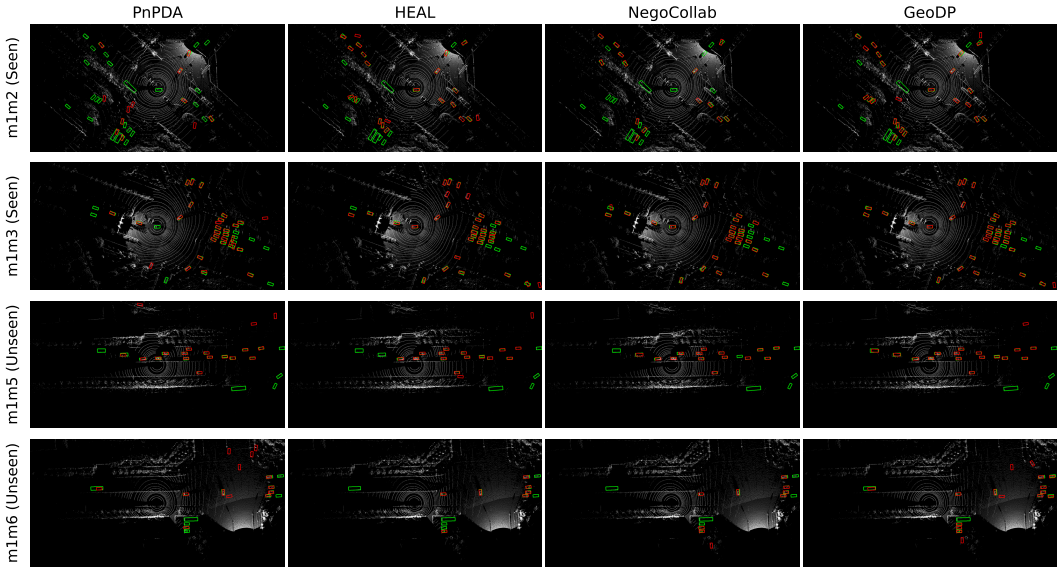

Figure 10: Qualitative detection results on DAIR-V2X under seen- and unseen-model settings. The four rows correspond to m1m2, m1m3, m1m5, and m1m6, respectively. Green boxes denote ground-truth objects, and red boxes denote predictions.