ADAPTATION-FREE COLLABORATIVE PERCEPTION WITH UNSEEN AGENT CONFIGURATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Collaborative perception improves 3D object detection by enabling agents to share complementary observations, but most existing methods assume fixed or known collaborator encoder configurations, limiting deployment in practice. In this work, we consider an open-world setting in which auxiliary agents with unseen configurations may appear after deployment, such as different LiDAR beam counts or encoder architectures. To address this challenge, we propose ALF, a collaborative perception framework that enables adaptation-free collaboration with unseen agent configurations by lifting lightweight box-level messages into ego-compatible auxiliary features. ALF converts auxiliary box-level messages into pseudo-BEV maps and synthesizes ego-compatible latent features by combining object-centric cues with scene context from the ego feature. On V2X-Real, under a zero-shot evaluation across 64 case studies, ALF outperforms the strongest prior baseline by 35.91% in relative mAP@0.7. Controlled comparisons further show that, in the zero-shot setting, ALF is more accurate and robust than direct decision-level late fusion, while requiring only 120 bytes per agent per frame.

1 INTRODUCTION

Collaborative perception enables multiple agents to exchange information and jointly reason about a scene beyond the sensing limits of any individual agent. By aggregating complementary observations, it improves spatial coverage and reduces uncertainty caused by occlusion and limited sensing range (Huang et al., 2025). Autonomous driving is a prominent application of this paradigm, where vehicles and infrastructure communicate to jointly perceive the environment (Wang et al., 2025; Bae et al., 2024b; Gao et al., 2025a).

Collaborative perception frameworks are categorized as early, intermediate, and late fusion, depending on the type of information exchanged: raw sensor data, latent feature representations, and object-level predictions. Among these, intermediate fusion strikes a balance between communication cost and representational fidelity (Xu et al., 2022a;b; Hu et al., 2022; Lu et al., 2023; Zhang et al., 2024). It preserves richer scene context than late fusion while avoiding the bandwidth overhead of early fusion. However, its central challenge is representation heterogeneity: features from auxiliary agents may be incompatible with the ego representation due to differences in *agent configurations*. We define an agent configuration as the combination of within-modality sensor specifications and encoder configurations, covering hardware-dependent properties such as LiDAR beam count and model-dependent choices such as backbone, voxelization, and resolution settings.

In realistic deployments, heterogeneity must be handled in an open-world setting, where an ego agent must collaborate with previously unseen auxiliary agents whose agent configurations were not observed during training. We refer to this scenario as *configuration-agnostic heterogeneity*. This setting is challenging because independently developed agents do not, in general, share a common latent interface. In contrast, most prior work addresses heterogeneity only within a closed set of agent configurations observed during training or fine-tuning, which we refer to as *configuration-specific heterogeneity*.

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.

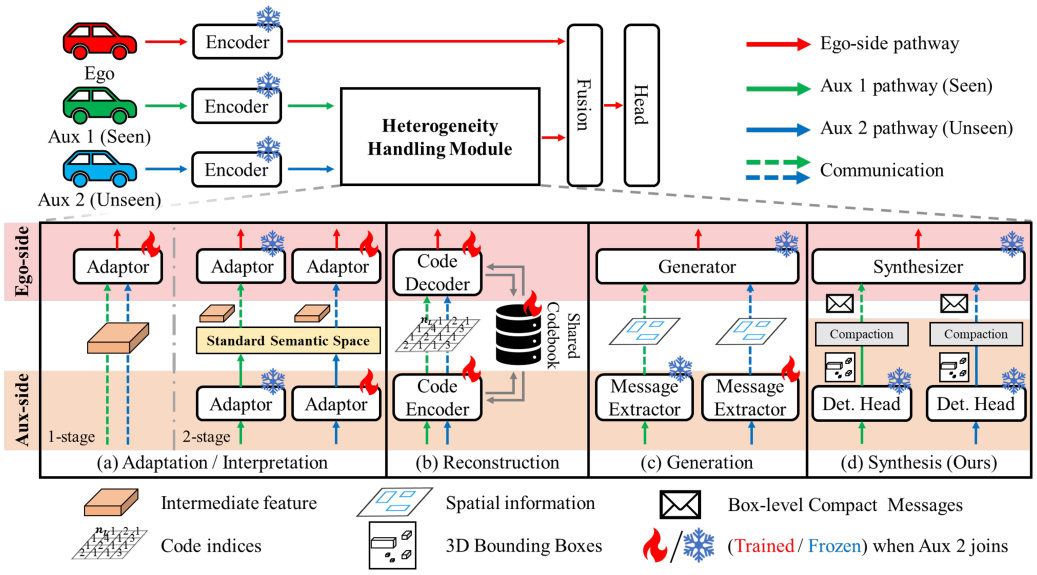


Figure 1: Comparison of strategies for integrating an unseen auxiliary agent (AUX2) in heterogeneous collaborative perception, after fine-tuning with a seen auxiliary agent (AUX1): (a) requires training additional adaptors for AUX2, (b) requires updating the shared codebook and code encoder/decoder, (c) requires training the message extractor of AUX2, while (d) requires no additional training.

Prior heterogeneous intermediate-fusion methods mainly rely on adaptation Lu et al. (2024); Xu et al. (2023); Xia et al. (2025); Luo et al. (2024); Gao et al. (2025b), reconstruction Hu et al. (2024), or generation Zhou et al. (2025), as illustrated in Fig. 1. These approaches typically require additional training to accommodate previously unseen auxiliary encoder configurations. This need for repeated per-agent retraining limits scalability and can degrade fusion performance when received representations fall outside the training distribution. Late fusion offers another route to handling heterogeneity. Because it exchanges only detected boxes, it requires no adaptation to new collaborators. However, it merges auxiliary boxes with finalized ego detections at the decision stage, so auxiliary information cannot interact with ego features.

We propose Adaptation-Free Late-to-Intermediate Fusion (ALF), which starts from the configuration-agnostic interface of late fusion and moves toward integrating auxiliary information at the intermediate stage. Each auxiliary agent sends a compact box-level message derived from its predicted 3D boxes, and the ego agent lifts this message into an ego-compatible latent feature before fusion (Fig. 1(d)). During this lifting, the ego latent feature provides scene context, allowing auxiliary cues to interact with ego features before the detection decision. Compared with heterogeneous intermediate fusion, ALF neither exchanges configuration-dependent latent features nor reconstructs the auxiliary agent's native feature space, and therefore does not require alignment or adaptation to unseen auxiliary feature spaces. As a result, ALF enables plug-and-play collaboration with heterogeneous auxiliary agents under a sub-1KB communication budget.

We summarize our contributions as follows.

- We propose ALF, an adaptation-free late-to-intermediate fusion framework that lifts box-level messages into ego-compatible latent features conditioned on the ego representation, enabling plug-and-play collaboration with heterogeneous auxiliary agents.

- We study configuration-agnostic heterogeneity, an open-world setting in which unseen auxiliary agent configurations may appear at inference time.
- We show that ALF achieves strong zero-shot generalization to unseen auxiliary configurations and outperforms direct late fusion under a sub-1KB communication budget.

2 Related Work

Collaborative Perception Methods Intermediate fusion has been the main focus in collaborative perception due to its favorable accuracy-bandwidth trade-off. F-Cooper Chen et al. (2019) uses maxout-based feature aggregation. V2X-ViT Xu et al. (2022b) introduces heterogeneous multi-agent attention and multi-scale window attention. CoBEVT Xu et al. (2022a) adopts fused axial attention for efficient local-global aggregation. Where2comm Hu et al. (2022) reduces communication cost by transmitting sparse foreground regions. MRCNet Hong et al. (2024) further improves robustness in noisy environments through motion-aware communication. A related line uses bounding boxes together with intermediate fusion in two ways. Some methods use boxes as geometric anchors for feature alignment, as in CoAlign Lu et al. (2023) and RoCo Huang et al. (2024), which correct relative poses from matched object detections before feature fusion. Others use boxes as auxiliary signals, as in CoSDH Xu et al. (2025), which applies confidence-aware late fusion on detection boxes to refine intermediate-fusion results under low bandwidth. In contrast, ALF uses compact box-level messages as the primary collaboration interface and lifts them into ego-compatible latent representations, enabling feature-level fusion without relying on configuration-dependent intermediate features.

Heterogeneity Handling Methods Heterogeneous encoders introduce a domain gap between ego and auxiliary features, making direct intermediate fusion challenging. Existing approaches can be broadly categorized into three types. The first is the adaptation- or interpretation-based approach, as shown in Fig. 1(a). This approach learns mappings either to the ego feature domain or to a shared semantic space. For example, MPDA Xu et al. (2023) learns a feature resizer and a sparse cross-domain transformer. HEAL Lu et al. (2024) builds a unified collaboration space and performs backward alignment by freezing the ego-side fusion and detection head. PnPDA Luo et al. (2024) and STAMP Gao et al. (2025b) emphasize non-intrusive and scalable alignment. PolyInter Xia et al. (2025) uses a prompt-driven interpreter. The second is the reconstruction-based approach, as shown in Fig. 1(b). Rather than transmitting latent features, methods such as CodeFilling Hu et al. (2024) send compact code indices and reconstruct latent features from a shared codebook. The third is the generation-based approach, as shown in Fig. 1(c). Methods such as GenComm Zhou et al. (2025) generate collaborator-aligned features from spatial cues, requiring only lightweight extractor tuning for newly introduced agents. All three approaches require additional training when a new auxiliary configuration appears. Late fusion avoids such training by exchanging configuration-agnostic boxes, but integrates them only at the decision stage. ALF builds on this interface and lifts box-level messages into ego-compatible latent features, enabling feature-level fusion without retraining.

3 Methodology

3.1 Problem Formulation

We consider collaborative 3D object detection with one ego agent and a time-varying set of auxiliary agents. Throughout this paper, we reserve the index 1 for the ego agent. The ego agent observes sensor data $\mathcal{P}_1$ and extracts an ego latent feature $\mathcal{F}_1$. Each auxiliary agent $i \in \mathcal{A}_{\text{aux}}$ has its own agent configuration $c_i \in \mathcal{C}$ and sends a message $\mathcal{M}_{i\to 1}$ to the ego agent. Here, an agent configuration refers to the attributes that determine the latent feature domain, including within-modality sensor specifications and encoder configurations.

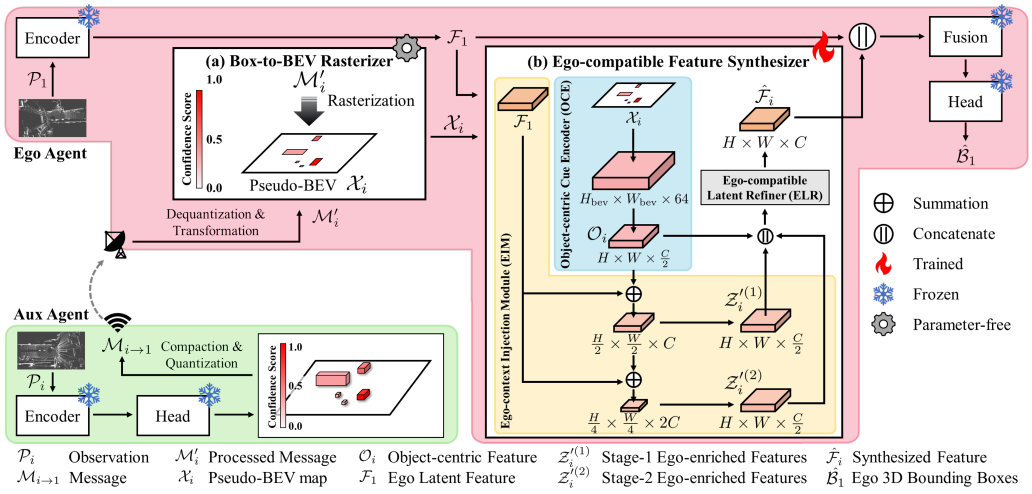


Figure 2: Overall architecture of ALF. (a) The box-level message is converted into a pseudo-BEV map by the Box-to-BEV Rasterizer (B2BR). (b) The Ego-compatible Feature Synthesizer (EFS) lifts the sparse pseudo-BEV cue into an ego-compatible latent feature using the ego latent feature.

Let $\mathcal{C}_{\text{seen}} \subset \mathcal{C}$ denote the auxiliary agent configurations available during training. We denote by $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ the training and deployment distributions over collaborative scenes, each consisting of the ego observation $\mathcal{P}_1$, the auxiliary agents $\{(\mathcal{P}_i, c_i)\}_{i\in\mathcal{A}_{\text{aux}}}$, and the ground-truth 3D bounding boxes $\mathcal{B}_1^\star$. All auxiliary configurations under $\mathcal{D}_{\text{train}}$ lie in $\mathcal{C}_{\text{seen}}$, whereas $\mathcal{D}_{\text{test}}$ may contain agents with $c_i \notin \mathcal{C}_{\text{seen}}$. Our goal is to learn a heterogeneity-handling module $\mathcal{H}_\theta$ on $\mathcal{D}_{\text{train}}$ and deploy it on $\mathcal{D}_{\text{test}}$ without any parameter update. Specifically, we aim to solve

$$\theta^\star = \arg\max_\theta \ \mathbb{E}_{\mathcal{D}_{\text{test}}}\Big[\mathcal{J}\big(\mathcal{B}_1^\star,\ \mathcal{CP}\big(\mathcal{F}_1, \mathcal{H}_\theta(\{\mathcal{M}_{i\to1}\}_{i\in\mathcal{A}_{\text{aux}}})\big)\big)\Big]$$
$$\text{subject to} \quad \sum_{i\in\mathcal{A}_{\text{aux}}} b(\mathcal{M}_{i\to1}) \le B,$$

using only samples from $\mathcal{D}_{\text{train}}$. Here, $\mathcal{J}$ denotes a 3D object detection evaluation function such as average precision (AP), $\mathcal{CP}$ denotes the collaborative perception detector, which consists of a feature fusion module and a detection head and produces the final prediction, $b(\cdot)$ measures communication cost, and B is the communication budget. This problem cannot be solved directly because $\mathcal{D}_{\text{test}}$ is unavailable during training. If $\mathcal{M}_{i\to1}$ were a latent feature, its representation would depend on c_i, and $\mathcal{H}_\theta$ would receive inputs outside its training support whenever $c_i \notin \mathcal{C}_{\text{seen}}$. ALF instead chooses a message space whose format is independent of c_i, so that $\mathcal{H}_\theta$ learned on $\mathcal{D}_{\text{train}}$ can be directly applied to $\mathcal{D}_{\text{test}}$.

3.2 Method Overview

ALF combines the configuration-agnostic interface of late fusion with feature-level integration without requiring compatibility between auxiliary and ego latent spaces. The overall pipeline has three stages. First, each auxiliary agent converts its local detections into a compact box-level message $\mathcal{M}_{i\to1}$ through field selection and quantization. Second, the ego agent dequantizes and transforms the received message into the ego frame using the relative pose $T_{i\to1}$ and converts it into a high-resolution pseudo-BEV cue $\mathcal{X}_i$ with the Box-to-BEV Rasterizer (B2BR). Third, the Ego-compatible Feature Synthesizer (EFS) integrates $\mathcal{X}_i$ with the ego latent feature $\mathcal{F}_1 \in \mathbb{R}^{H\times W\times C}$ to lift an ego-compatible latent feature $\hat{\mathcal{F}}_i \in \mathbb{R}^{H\times W\times C}$.

The heterogeneity-handling module $\mathcal{H}_\theta$ introduced in Sec. 3.1 is instantiated as a sequential composition of dequantization, coordinate transformation, B2BR, and EFS:

$$\hat{\mathcal{F}}_i = \mathcal{H}_\theta(\mathcal{M}_{i\to 1};\, T_{i\to 1},\, \mathcal{F}_1) = f_{\mathrm{EFS},\theta}(f_{\mathrm{B2BR}}(\mathcal{T}(\mathcal{D}(\mathcal{M}_{i\to 1})\,, T_{i\to 1}))\,, \mathcal{F}_1)\,,$$

where $f_{\mathrm{EFS},\theta}$ denotes the EFS, f_{B2BR} denotes the B2BR, $\mathcal{T}$ denotes transformation into the ego frame using $T_{i\to 1}$, and $\mathcal{D}$ denotes dequantization. After synthesizing the ego-compatible latent feature $\hat{\mathcal{F}}_i$, the final 3D bounding boxes $\hat{\mathcal{B}}_1$are predicted through the collaborative perception model $\mathcal{CP}$:

$$\hat{\mathcal{B}}_1 = \mathcal{CP}\left(\mathcal{F}_1, \{\hat{\mathcal{F}}_i\}_{i\in\mathcal{A}_{\mathrm{aux}}}\right).$$

Fig. 2 summarizes the full pipeline.

3.3 Box-Level Message Compaction

ALF communicates through box-level messages rather than configuration-dependent latent features. This choice is central to our open-world setting, where auxiliary agents may use unseen agent configurations and therefore cannot be assumed to share a compatible latent space with the ego agent.

Each auxiliary agent $i \in \mathcal{A}_{\mathrm{aux}}$ sends a message defined as $\mathcal{M}_{i\to 1} = \{m_{i,k}\}_{k=1}^{K_{\max}}$ where $K_{\max}$ denotes the maximum number of prediction boxes and $m_{i,k}$ contains the BEV-relevant fields of a prediction box. In particular, we define $m_{i,k} = (\tilde{x}_{i,k}, \tilde{y}_{i,k}, \tilde{w}_{i,k}, \tilde{\ell}_{i,k}, \tilde{\psi}_{i,k}, \tilde{s}_{i,k})$ where $\tilde{x}_{i,k}$ and $\tilde{y}_{i,k}$ denote the box center in the local BEV plane, $\tilde{w}_{i,k}$ and $\tilde{\ell}_{i,k}$ denote the box width and length, $\tilde{\psi}_{i,k}$ denotes the yaw angle, and $\tilde{s}_{i,k}$ denotes the confidence score. Here, $\tilde{(\cdot)}$ indicates quantized values. Quantization details are provided in Appendix A.1.

If the detector produces fewer than $K_{\max}$ boxes, the remaining entries are zero-padded. If it produces more, we retain the top-$K_{\max}$ boxes ranked by confidence. This compact representation preserves essential object evidence for collaboration while enforcing a bounded, low-bandwidth communication interface.

3.4 Box-to-BEV Rasterizer

We now convert the compact box-level message into a pseudo-BEV representation. After transmission, the ego agent first dequantizes the received message and transforms the spatial fields (position, size, orientation) into the ego frame using relative pose $T_{i\to 1}$, yielding a processed message $\mathcal{M}'_i$. We then convert $\mathcal{M}'_i$ into a pseudo-BEV map $\mathcal{X}_i$ with the B2BR:

$$\mathcal{X}_i = f_{\mathrm{B2BR}}(\mathcal{M}'_i) \in \mathbb{R}^{H_{\mathrm{bev}}\times W_{\mathrm{bev}}\times 1},$$

where H_{bev} and W_{bev} denote the height and width of the BEV grid.

This pseudo-BEV representation $\mathcal{X}_i$ serves two purposes. First, it preserves the spatial footprint of auxiliary detections in a shared representation that is independent of the auxiliary agent configuration. Second, it provides a high-resolution object-centric cue that can be lifted into the ego latent space by the subsequent synthesizer. Additional details of B2BR are provided in Appendix A.2.

3.5 Ego-compatible Feature Synthesizer

The pseudo-BEV map $\mathcal{X}_i$ is a sparse representation of object-level information. To enable feature-level integration before the final detection decision, EFS lifts this cue into the ego latent space:

$$\hat{\mathcal{F}}_i = f_{\mathrm{EFS},\theta}(\mathcal{X}_i, \mathcal{F}_1) \in \mathbb{R}^{H\times W\times C}.$$

Here, "ego-compatible" means compatibility with the latent space expected by the frozen ego fusion and detection modules, rather than reconstruction of the auxiliary agent's native feature space.

EFS consists of three blocks:

$$\mathcal{O}_i = \Phi_{\mathrm{OCE}}(\mathcal{X}_i), \qquad (\mathcal{Z}_i'^{(1)}, \mathcal{Z}_i'^{(2)}) = \Phi_{\mathrm{EIM}}(\mathcal{O}_i, \mathcal{F}_1), \qquad \hat{\mathcal{F}}_i = \Phi_{\mathrm{ELR}}\Big([\mathcal{O}_i, \mathcal{Z}_i'^{(1)}, \mathcal{Z}_i'^{(2)}]\Big).$$

The **Object-centric Cue Encoder** (OCE) converts $\mathcal{X}_i$ into a compact object-centric feature $\mathcal{O}_i$ suitable for latent-space synthesis. The **Ego-context Injection Module** (EIM) injects multi-scale scene context from the ego latent feature $\mathcal{F}_1$ to complement the sparse object-level cue with scene geometry and semantics. The **Ego-compatible Latent Refiner** (ELR) aggregates the object-centric and ego-enriched features and refines them into the latent space expected by the frozen ego-side fusion module, producing the final ego-compatible latent feature $\hat{\mathcal{F}}_i$. Detailed architectural configurations are provided in Appendix A.3.

3.6 Loss

We train ALF using a standard ego detection loss $\mathcal{L}_{\mathrm{det}}$ and an auxiliary feature-alignment loss $\mathcal{L}_{\mathrm{align}}$:

$$\mathcal{L} = \alpha_{\mathrm{det}}\mathcal{L}_{\mathrm{det}} + \alpha_{\mathrm{align}}\frac{1}{|\mathcal{A}_{\mathrm{aux}}|}\sum_{i\in\mathcal{A}_{\mathrm{aux}}}\mathcal{L}_{\mathrm{align}}^{(i)}.$$

The detection loss is $\mathcal{L}_{\mathrm{det}} = \alpha_{\mathrm{cls}}\mathcal{L}_{\mathrm{cls}} + \alpha_{\mathrm{reg}}\mathcal{L}_{\mathrm{reg}}$, where $\mathcal{L}_{\mathrm{cls}}$ is the focal classification loss Lin et al. (2017) and $\mathcal{L}_{\mathrm{reg}}$ is the Smooth L1 box regression loss Girshick (2015).

For each auxiliary agent i, we align the synthesized feature $\hat{\mathcal{F}}_i$ to an ego-domain teacher feature $\mathcal{F}_i^\star$ so that it matches the latent representation expected by the frozen ego-side fusion and detection modules. The teacher is obtained by transforming the raw auxiliary observation $\mathcal{P}_i$ into the ego frame using $T_{i\to1}$ and encoding it with the frozen ego encoder. It thus represents what the ego encoder would produce from the auxiliary viewpoint. To prevent background regions from dominating the alignment, we separately supervise object and background regions:

$$\mathcal{L}_{\mathrm{align}}^{(i)} = \alpha_{\mathrm{obj}}\,\mathcal{L}_{\cos}(\hat{\mathcal{F}}_i, \mathcal{F}_i^\star; \Omega_i^{\mathrm{obj}}) + \alpha_{\mathrm{bg}}\,\mathcal{L}_{\cos}(\hat{\mathcal{F}}_i, \mathcal{F}_i^\star; \Omega_i^{\mathrm{bg}}), \tag{1}$$

where Ω_i^{obj} and Ω_i^{bg} denote the object and background masks, respectively, and $\mathcal{L}_{\cos}$ denotes the region-weighted cosine alignment loss. Detailed constructions of the teacher feature, masks, and alignment loss are provided in Appendix A.4 and A.5.

3.7 Training Strategy and Adaptation-Free Deployment

ALF is trained once by optimizing only the EFS parameters θ, while the ego encoder, fusion module, and detection head remain frozen. Training requires raw auxiliary observations, which are used both to construct the teacher feature and to generate box messages. Specifically, we apply the pretrained ego detector to the raw auxiliary observations and construct box-level messages from its predictions. At deployment, EFS remains fixed and raw auxiliary observations are not required. Auxiliary agents act as black-box detectors that provide $\mathcal{M}_{i\to1}$ and $T_{i\to1}$, so any agent following this message format can participate without fine-tuning or retraining.

4 Experiments

4.1 Experimental Setup

Dataset, metrics, and baselines. We evaluate ALF on V2X-Real Xiang et al. (2024), a large-scale real-world V2X cooperative perception dataset, and V2XSet Xu et al. (2022b), a simulated V2X cooperative perception dataset. For V2X-Real, performance is measured by Average Precision (AP) for each superclass, and we report the mean Average Precision (mAP) across all superclasses. We report mAP@0.5 and mAP@0.7

at IoU thresholds of 0.5 and 0.7, respectively. For V2XSet, we follow the standard protocol and report vehicle AP at the same IoU thresholds. Unless otherwise stated, all results in the main paper are reported on V2X-Real, while the V2XSet results are provided in Appendix C.6.

To evaluate ALF under heterogeneous settings, we compare it with representative baselines, including E2E, MPDA Xu et al. (2023), CodeFilling Hu et al. (2024), and GenComm Zhou et al. (2025). Here, E2E denotes an end-to-end baseline that jointly trains all parameters, including the ego encoder, auxiliary encoders, fusion module, and detection head. To assess generality across fusion architectures, we evaluate each method across various fusion backbones, including AttnFuse Xu et al. (2022c), CoBEVT Xu et al. (2022a), V2X-ViT Xu et al. (2022b), and ParCon Bae et al. (2024a). Detailed descriptions of the dataset, metrics, and implementation details are provided in Appendix B.1, B.2, and B.4, respectively.

Experiment Design. We define the ego and auxiliary agent configurations separately. The ego agent is fixed to using a 128-beam LiDAR and a PointPillar backbone with voxel size $[0.4, 0.4, 30]$ along the x, y, and z axes, respectively. For auxiliary agents, we consider two LiDAR specifications, 128-beam and 64-beam LiDARs, combined with four encoder configurations: two voxel-resolution variants of PointPillar Lang et al. (2019) and two voxel-resolution variants of SECOND Yan et al. (2018), as summarized in Appendix B.3. This yields eight auxiliary agent configurations.

We define three evaluation protocols, where EGO denotes the ego agent configuration (PP4 with a 128-beam LiDAR), AUX1 denotes a *seen* auxiliary agent configuration, and AUX2 denotes an *unseen* auxiliary agent configuration:

(1) **Seen-pair evaluation.** The model is trained and evaluated on the same (EGO+AUX1) pair.
(2) **Zero-shot unseen evaluation.** The model is trained on (EGO+AUX1) but evaluated on (EGO+AUX2) without additional training.
(3) **Unseen adaptation evaluation.** The model is further fine-tuned on (EGO+AUX1+AUX2) and evaluated on (EGO+AUX2). Results are reported in Appendix C.3.

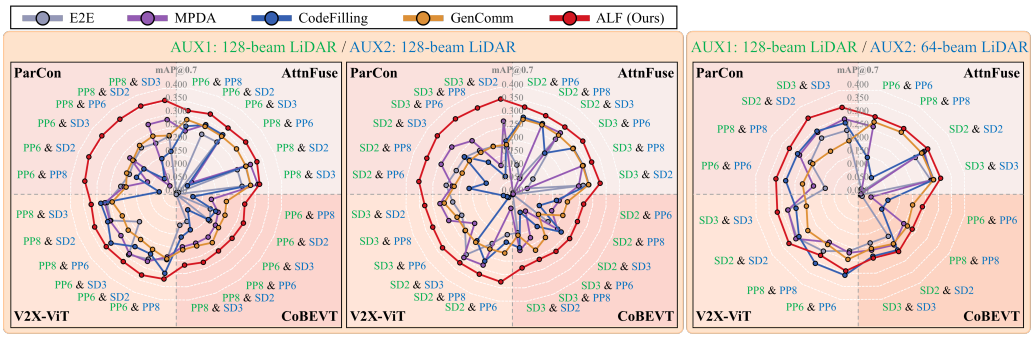
These protocols define how the baselines are trained. ALF is trained once with messages generated by the ego detector, whose configuration differs from all auxiliary configurations considered. Thus, every auxiliary configuration is unseen by ALF under all protocols. Protocol-specific details are provided in Appendix B.5.

4.2 QUANTITATIVE ANALYSIS

Zero-Shot Unseen Evaluation. Fig. 3 reports zero-shot unseen evaluation results for ALF and the baselines under two sources of heterogeneity. Averaged over all 64 cases in Fig. 3, ALF achieves 0.311 mAP@0.7, a 35.91% relative improvement over the strongest baseline, GenComm (0.229). As shown in Fig. 3(a), ALF outperforms all baselines in every case under encoder-configuration heterogeneity. Fig. 3(b) further shows that ALF achieves the best mAP@0.7 in 14 of 16 LiDAR-specification heterogeneity cases. These results demonstrate ALF's ability to handle unseen auxiliary configurations without additional adaptation.

Moreover, ALF maintains stable performance across both heterogeneity types and individual unseen cases. By contrast, E2E, MPDA, and CodeFilling show severe degradation in some unseen cases, with mAP@0.7 dropping below 0.1. GenComm avoids such a dramatic drop but remains sensitive to the fusion method; under V2X-ViT and ParCon, it performs substantially below ALF.

Seen-Pair Evaluation. Tab. 1 reports seen-pair results, where each baseline is trained on the same auxiliary configuration it is evaluated on. Even in this setting, ALF, which is never trained on these configurations, achieves the highest average mAP@0.5 and mAP@0.7, while requiring only 0.0096 Mbps (120 bytes per frame at 10 Hz).



(a) Encoder Configuration (b) LiDAR Specification

Figure 3: Comparison of detection accuracy under **zero-shot unseen evaluation** with two sources of heterogeneity. (a) Encoder configuration: AUX1 and AUX2 use different encoders with 128-beam LiDARs. (b) LiDAR specification: AUX1 and AUX2 use the same encoder with 128-beam and 64-beam LiDARs, respectively. Full results are provided in Appendix C.2.

Table 1: **Seen-pair evaluation**. Results are averaged over four fusion backbones and four auxiliary encoder configurations (in total, 16 cases). Full results are provided in Tab. 4 in Appendix C.1.

	E2E	MPDA	CodeFilling	GenComm	ALF (Ours)
Average mAP@0.5 / 0.7 ↑	0.463 / 0.287	0.474 / 0.298	0.432 / 0.267	0.448 / 0.269	**0.485 / 0.317**
# Best cases @0.5 / 0.7 ↑	2 / 1	5 / 2	0 / 0	1 / 2	**8 / 11**
Bandwidth@10Hz (Mbps) ↓	503–1845	503–1845	1.475–2.703	3.932–7.209	**0.0096**

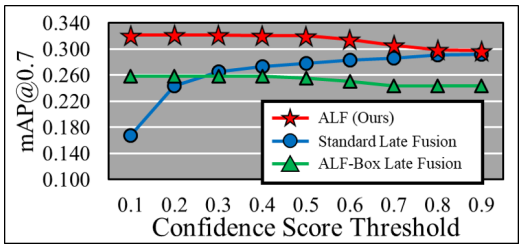


Figure 4: Confidence-threshold sensitivity. Results are averaged over four unseen encoder pairs.

Table 2: EFS ablation in the zero-shot setting. Results are averaged over four unseen encoder pairs.

OCE	EIM	ELR	mAP@0.5	mAP@0.7
			0.453	0.295
✓			0.465	0.300
✓	✓		0.476	0.306
✓		✓	0.468	0.303
✓	✓	✓	**0.481**	**0.322**

4.3 Comparison with Decision-Level Late Fusion

To isolate the benefit of feature synthesis, we compare ALF with two decision-level fusion baselines. *Standard Late Fusion* directly fuses auxiliary detections above the confidence threshold with ego predictions, whereas *ALF-Box Late Fusion* directly fuses the same top-K boxes used by ALF. In contrast, *ALF* lifts these boxes into ego-compatible latent features through B2BR and EFS before feature-level fusion.

Sensitivity to Confidence Thresholds. Fig. 4 reports the average mAP@0.7 over four unseen encoder pairs as the auxiliary confidence threshold varies. Standard Late Fusion is highly sensitive to the threshold, improving from 0.168 to 0.292, whereas ALF-Box Late Fusion remains relatively stable but substantially below ALF. In contrast, ALF maintains 0.320–0.322 mAP@0.7 over thresholds from 0.1 to 0.5 and outperforms both

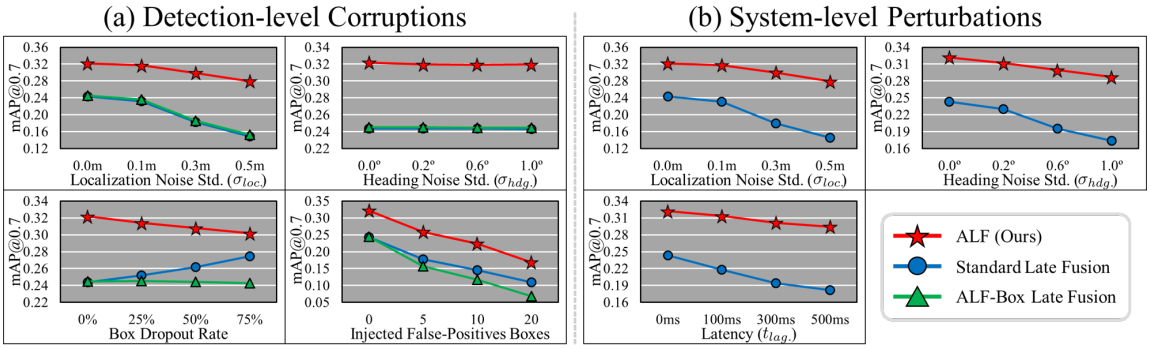


Figure 5: Detection accuracy under (a) detection-level corruptions applied to auxiliary boxes and (b) system-level perturbations applied to the relative pose and communication. Results are averaged over four unseen encoder pairs.

baselines across the full range. This stability is important in open-world deployment, where the threshold cannot be tuned in advance for the detectors of unseen auxiliary agents.

Robustness to Corruptions and Perturbations. We perturb auxiliary boxes using box dropout, false-positive injection, and zero-mean Gaussian noise on box centers (σ_{loc}) and yaw angles (σ_{hdg}), and further apply the same noise to the agent pose used for coordinate transformation, together with communication latency sampled uniformly up to t_{lag}. As shown in Fig. 5, ALF outperforms both late-fusion baselines at all detection-level corruptions and degrades gradually under system-level perturbations, indicating that lifting imperfect box cues into ego-compatible features is more robust than directly using them at the decision level.

4.4 Ablation Study

As shown in Tab. 2, directly using the pseudo-BEV representation without EFS yields the lowest performance, and each EFS component contributes additional gains, with the full EFS performing best. These results support both the benefit of feature synthesis beyond the pseudo-BEV cue itself and the complementary roles of OCE, EIM, and ELR under unseen auxiliary configurations. Additional ablations on message quantization, box budget, and EFS components are provided in Appendix C.5.

5 Conclusion

We study configuration-agnostic heterogeneity in cooperative 3D object detection, where agents with unseen LiDAR specifications or encoder configurations may appear after training and must collaborate without further adaptation. To address this challenge, we propose ALF, a late-to-intermediate fusion framework that communicates compact box-level messages, converts them into pseudo-BEV maps, and synthesizes ego-compatible features using ego-scene context. Experiments on V2X-Real show zero-shot generalization to unseen configurations, improved accuracy and robustness over direct decision-level late fusion, and low communication cost, supporting ego-side synthesis as a scalable interface that retains late-fusion interoperability while enabling pre-decision feature-level integration.

Limitations and Future Work. We focus on within-modality configuration heterogeneity and leave cross-modal collaboration, raw sensor noise, packet loss, and transmission errors to future work.

REFERENCES

Hyunchul Bae, Minhee Kang, and Heejin Ahn. ParCon: Noise-robust collaborative perception via multi-module parallel connection. *arXiv preprint arXiv:2407.11546*, 2024a.

Hyunchul Bae, Minhee Kang, Minwoo Song, and Heejin Ahn. Rethinking the role of infrastructure in collaborative perception. In *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, 2024b.

Qi Chen, Xu Ma, Sihai Tang, Jingda Guo, Qing Yang, and Song Fu. F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing (SEC)*, pp. 88–100, 2019.

Xiangbo Gao, Zhengzhong Tu, et al. Airv2x: Unified air–ground/vehicle-to-everything collaboration for perception. *arXiv preprint arXiv:2506.19283*, 2025a.

Xiangbo Gao, Runsheng Xu, Jiachen Li, Ziran Wang, Zhiwen Fan, and Zhengzhong Tu. STAMP: Scalable task- and model-agnostic collaborative perception. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025b.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015.

Shixin Hong, Yu Liu, Zhi Li, Shaohui Li, and You He. Multi-agent collaborative perception via motion-aware robust communication network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15301–15310, 2024.

Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 4874–4886, 2022.

Yue Hu, Juntong Peng, Sifei Liu, Junhao Ge, Si Liu, and Siheng Chen. Communication-efficient collaborative perception via information filling with codebook. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15481–15490, 2024.

Tao Huang, Jianan Liu, Xi Zhou, Dinh C Nguyen, Mostafa Rahimi Azghadi, Yuxuan Xia, Qing-Long Han, and Sumei Sun. Vehicle-to-everything cooperative perception for autonomous driving. *Proceedings of the IEEE*, 2025.

Zhe Huang, Shuo Wang, Yongcai Wang, Wanting Li, Deying Li, and Lei Wang. Roco: Robust cooperative perception by iterative object matching and pose adjustment. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, pp. 7833–7842, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

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

Yifan Lu, Yue Hu, Yiqi Zhong, Dequan Wang, Siheng Chen, and Yanfeng Wang. An extensible framework for open heterogeneous collaborative perception. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 287–303, 2024.

Benwu Wang, Xu Li, Qimin Xu, Wenkai Zhu, Yinan Du, Haoyang Che, Zhiyuan Xu, and Baidan Li. Vicooper: Communication-efficient vehicle-infrastructure cooperative 3d object detection leveraging roadside hd point cloud background map priors. *IEEE Internet of Things Journal*, 2025.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1592–1601, 2025.

Hao Xiang, Zhaoliang Zheng, Xin Xia, Runsheng Xu, Letian Gao, Zewei Zhou, Xu Han, Xinkai Ji, Mingxi Li, Zonglin Meng, et al. V2x-real: a large-scale dataset for vehicle-to-everything cooperative perception. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 455–470, 2024.

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

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 *Proceedings of the Conference on Robot Learning (CoRL)*, 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 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 *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2583–2589, 2022c.

Runsheng Xu, Jinlong Li, Xiaoyu Dong, Hongkai Yu, and Jiaqi Ma. Bridging the domain gap for multi-agent perception. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6035–6042, 2023.

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

Jingyu Zhang, Kun Yang, Yilei Wang, Hanqi Wang, Peng Sun, and Liang Song. Ermvp: Communication-efficient and collaboration-robust multi-vehicle perception in challenging environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12575–12584, 2024.

Junfei Zhou, Penglin Dai, Quanmin Wei, Bingyi Liu, Xiao Wu, and Jianping Wang. Pragmatic heterogeneous collaborative perception via generative communication mechanism. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

A METHOD DETAILS

A.1 QUANTIZATION DETAILS

For each scalar field v, we apply b-bit zero-point quantization:

$$\tilde{v} = \mathrm{clip}\left(\left\lfloor \frac{v}{s_v} \right\rceil + z_v,\ q_{\min}, q_{\max}\right), \quad s_v = \frac{v_{\max} - v_{\min}}{q_{\max} - q_{\min}}, \quad z_v = \left\lfloor q_{\min} - \frac{v_{\min}}{s_v} \right\rceil, \tag{2}$$

where $q_{\min} = 0$ and $q_{\max} = 2^b - 1$.

A.2 DETAILS OF THE BOX-TO-BEV RASTERIZER

Dequantization and Ego-Frame Transformation. Upon reception, the ego agent dequantizes each field via

$$v^{\mathrm{dq}} = s_v(\tilde{v} - z_v), \tag{3}$$

and transforms $(x^{\mathrm{dq}}, y^{\mathrm{dq}}, \psi^{\mathrm{dq}})$ to the ego frame using $T_{i\to 1}$.

Rasterization Details. B2BR converts the processed box message into a single-channel pseudo-BEV map by rasterizing box footprints on the BEV grid. Each box contributes a confidence-weighted occupancy pattern, and the final map stores the occupied cells after rasterization.

The grid resolution is determined by the ego agent's BEV detection range $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ and the voxel size (v_x, v_y):

$$H_{\mathrm{bev}} = \left\lfloor \frac{x_{\max} - x_{\min}}{v_x} \right\rfloor, \qquad W_{\mathrm{bev}} = \left\lfloor \frac{y_{\max} - y_{\min}}{v_y} \right\rfloor.$$

For a transformed box parameterized by (x, y, ℓ, w, ψ) on the BEV plane, we test whether each grid cell center lies inside the rotated box footprint. Let $(x_{g,u}, y_{g,v})$ denote the center of grid cell (u, v). We first translate the cell center to the box center and then rotate it into the box-local coordinate frame:

$$\delta x = x_{g,u} - x, \qquad \delta y = y_{g,v} - y, \tag{4}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}. \tag{5}$$

The cell is marked as occupied if

$$|x'| \le \frac{\ell}{2}, \qquad |y'| \le \frac{w}{2}. \tag{6}$$

If multiple boxes overlap in the same cell, we assign the maximum confidence score among them. This produces a sparse pseudo-BEV map that preserves the spatial footprint of auxiliary detections while remaining independent of the auxiliary encoder configuration.

A.3 DETAILED ARCHITECTURE OF THE EGO-COMPATIBLE FEATURE SYNTHESIZER

Basic operators. We denote a 2D convolution with kernel size k and stride s by $\mathrm{Conv}_{k,s}$, batch normalization by BN, and the Sigmoid Linear Unit by SiLU. We define

$$\mathrm{CBA}_{k,s}(\cdot) \triangleq \mathrm{Conv}_{k,s} \to \mathrm{BN} \to \mathrm{SiLU}, \tag{7}$$

$$\mathrm{Bilinear}_{\times m}(\cdot) \triangleq \text{bilinear interpolation by a factor of } m. \tag{8}$$

We further use a residual block

$$\mathrm{RB}(\mathbf{x}) \triangleq \mathbf{x} + \mathrm{BN}(\mathrm{Conv}_{3,1}(\mathrm{SiLU}(\mathrm{BN}(\mathrm{Conv}_{3,1}(\mathbf{x}))))). \tag{9}$$

Object-centric Cue Encoder (OCE). The operator $\mathrm{Expand}(\cdot)$ performs channel-wise expansion by replication:

$$\mathrm{Expand}(\mathcal{X}_i) = \underbrace{[\mathcal{X}_i, \ldots, \mathcal{X}_i]}_{C_0 \text{ times}} \in \mathbb{R}^{H_{\mathrm{bev}} \times W_{\mathrm{bev}} \times C_0}, \tag{10}$$

where $C_0 = 64$ in our implementation. This operation introduces no learnable parameters and simply duplicates the single-channel input along the channel dimension.

The encoder $\mathrm{Enc}_L(\cdot)$ consists of L stages:

$$\mathcal{O}_i^{(\ell)} = \mathrm{CBA}_{3,1}\Big(\mathrm{CBA}_{3,2}\Big(\mathcal{O}_i^{(\ell-1)}\Big)\Big), \quad \ell = 1, \ldots, L, \tag{11}$$

with $\mathcal{O}_i^{(0)} = \mathrm{Expand}(\mathcal{X}_i)$. Here, $\mathrm{CBA}_{3,2}$ reduces spatial resolution by a factor of 2, while $\mathrm{CBA}_{3,1}$ refines features at the same scale. The final OCE output is

$$\mathcal{O}_i = \mathcal{O}_i^{(L)} = \mathrm{Enc}_L(\mathrm{Expand}(\mathcal{X}_i)) \in \mathbb{R}^{H \times W \times C/2}. \tag{12}$$

Across the L stages, the spatial resolution is reduced by a factor of 2^L, while the channel dimension is progressively increased to reach the output dimension of $\mathcal{O}_i$.

Ego-context Injection Module (EIM). EIM enriches the object-centric feature $\mathcal{O}_i$ with scene geometry and semantic context from the ego latent feature $\mathcal{F}_1$. It follows a two-stage pyramid-style design in which ego context is injected before successive downsampling operations. The ego latent feature is spatially aligned to each target resolution using adaptive average pooling.

The downsampling operators are defined as

$$\mathrm{Down}_k(\mathbf{x}) = \mathrm{CBA}_{3,2}(\mathbf{x}), \quad k \in \{1, 2\}. \tag{13}$$

The upsampling operators restore resolution via bilinear interpolation followed by refinement:

$$\mathrm{Up}_k(\mathbf{x}) = \mathrm{CBA}_{3,1}(\mathrm{Bilinear}_{\times s_k}(\mathbf{x})), \tag{14}$$

where $s_1 = 2$ and $s_2 = 4$.

Thus, the two ego-enriched features are instantiated as

$$\mathcal{Z}_i'^{(1)} = \mathrm{Up}_1(\mathrm{Down}_1(\mathcal{O}_i + \mathrm{AvgPool}(\mathcal{F}_1))), \tag{15}$$

$$\mathcal{Z}_i'^{(2)} = \mathrm{Up}_2(\mathrm{Down}_2(\mathrm{Down}_1(\mathcal{O}_i) + \mathrm{AvgPool}(\mathcal{F}_1))). \tag{16}$$

Ego-compatible Latent Refiner (ELR). The projection operator is defined as

$$\mathrm{Proj}(\mathbf{x}) = \mathrm{CBA}_{3,1}\big(\mathrm{CBA}_{3,1}(\mathbf{x})\big), \tag{17}$$

which maps the concatenated feature to channel dimension C.

The refinement operator is defined as

$$\mathrm{ResBlock}_3(\mathbf{x}) = (\mathrm{RB} \circ \mathrm{RB} \circ \mathrm{RB})(\mathbf{x}), \tag{18}$$

which applies three residual blocks while preserving spatial resolution.

Overall EFS output. The object-centric feature $\mathcal{O}_i$ and the ego-enriched features $\mathcal{Z}_i'^{(1)}$ and $\mathcal{Z}_i'^{(2)}$ provide complementary object-level information and scene context. ELR aggregates these features and projects them into the latent space expected by the ego-side fusion module:

$$\tilde{\mathcal{F}}_i = \mathrm{Proj}\Big([\mathcal{O}_i, \mathcal{Z}_i'^{(1)}, \mathcal{Z}_i'^{(2)}]\Big) \in \mathbb{R}^{H\times W\times C}. \tag{19}$$

The projected representation is then refined as

$$\hat{\mathcal{F}}_i = \mathrm{ResBlock}_3\Big(\tilde{\mathcal{F}}_i\Big) \in \mathbb{R}^{H\times W\times C}. \tag{20}$$

The resulting $\hat{\mathcal{F}}_i$ is the ego-compatible latent feature used for subsequent feature-level fusion.

A.4 Teacher Feature Construction

For each auxiliary agent i, we define a teacher feature $\mathcal{F}_i^\star$ as the latent feature produced by the frozen ego encoder when the auxiliary observation is expressed in the ego coordinate frame. Specifically, let $\mathcal{P}_i$ denote the auxiliary observation and let $T_{i\to 1}$ denote the relative pose from agent i to the ego agent. We first transform $\mathcal{P}_i$ into the ego frame and denote the transformed observation by $\tilde{\mathcal{P}}_{i\to 1}$. The teacher feature is then obtained by encoding $\tilde{\mathcal{P}}_{i\to 1}$ with the frozen ego encoder.

This construction differs from using the auxiliary agent's native latent feature. Instead, it defines the supervisory target directly in the latent space expected by the ego fusion and detection stack. As a result, the alignment objective encourages the synthesized feature to be compatible with the ego latent domain while preserving the foreground structure induced by the auxiliary observation.

A.5 Pseudo-BEV-Based Mask Construction for Region-Aware Alignment

We provide implementation details for constructing masks derived from the pseudo-BEV map. Let $\mathcal{X}_i \in \mathbb{R}^{H_{\mathrm{bev}}\times W_{\mathrm{bev}}\times 1}$ denote the pseudo-BEV map for auxiliary agent i. We downsample $\mathcal{X}_i$ to the target spatial resolution (H, W) of the ego latent feature using max pooling:

$$\bar{\mathcal{X}}_i = \mathrm{MaxPool}(\mathcal{X}_i; H, W). \tag{21}$$

We first obtain a foreground occupancy mask by thresholding the downsampled map:

$$\mathbf{\Omega}_i^{\mathrm{fg}} = \mathbf{1}[\bar{\mathcal{X}}_i > \tau]. \tag{22}$$

Because this mask is often too sparse, we further expand it using morphological dilation to obtain the object-region mask used in the main loss:

$$\mathbf{\Omega}_i^{\mathrm{obj}} = \mathrm{Dilate}(\mathbf{\Omega}_i^{\mathrm{fg}}; d_r), \tag{23}$$

where d_r denotes the dilation radius, and the corresponding kernel size is $2d_r + 1$. The background mask is then defined as the complement of the object-region mask:

$$\mathbf{\Omega}_i^{\mathrm{bg}} = \mathbf{1} - \mathbf{\Omega}_i^{\mathrm{obj}}. \tag{24}$$

This construction allows the alignment loss to emphasize not only object-occupied regions but also their surrounding context. In all experiments, we set $\tau = 0$, use max pooling for spatial downsampling, and set the dilation radius to $d_r = 2$.

Table 3: Encoder configurations.

Backbone Architecture	Encoder Configuration	Voxel Size (v_x, v_y, v_z)	2D / 3D CNN Layers	Detection Range $([x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}])$	Feature Size $(H \times W \times C)$
PointPillar	PP4	0.4, 0.4, 30	21 / 0	[-102.4, 102.4] × [-38.4, 38.4]	$48 \times 128 \times 256$
	PP6	0.6, 0.6, 30	21 / 0	[-105.6, 105.6] × [-38.4, 38.4]	$64 \times 176 \times 256$
	PP8	0.8, 0.8, 30	21 / 0	[-102.4, 102.4] × [-38.4, 38.4]	$48 \times 128 \times 256$
SECOND	SD2	0.2, 0.2, 0.1	12 / 12	[-102.4, 102.4] × [-38.4, 38.4]	$48 \times 128 \times 512$
	SD3	0.3, 0.3, 0.1	12 / 12	[-105.6, 105.6] × [-38.4, 38.4]	$64 \times 176 \times 512$

Region-weighted cosine alignment. Given a region mask $\mathbf{\Omega}$, the region-weighted cosine alignment loss used in Sec. 3.6 is defined as

$$\mathcal{L}_{\cos}(\hat{\mathcal{F}}_i, \mathcal{F}_i^\star; \mathbf{\Omega}) = \frac{1}{\max(1, \|\mathbf{\Omega}\|_1)} \sum_{u=1}^{H} \sum_{v=1}^{W} \mathbf{\Omega}(u,v) \left(1 - \cos\Big(\hat{\mathcal{F}}_i(u,v), \mathcal{F}_i^\star(u,v)\Big)\right). \tag{25}$$

This loss computes feature alignment over the selected spatial region, with normalization by the number of selected cells and a safeguard against division by zero for an empty mask.

B Experimental details

B.1 Dataset

We evaluate all methods on V2X-Real, a large-scale real-world V2X cooperative perception dataset collected with two connected automated vehicles and two smart infrastructure units, each equipped with LiDAR and multi-view cameras. It contains 33K LiDAR frames and 171K camera images with over 1.2M annotated 3D bounding boxes across 10 categories, and provides four subsets (vehicle-centric, infrastructure-centric, V2V, and I2I) to evaluate diverse collaboration modes.

We further evaluate multiple heterogeneity-handling methods, including ALF, on V2XSet using two representative fusion backbones, AttnFuse and V2X-ViT, to assess whether the observed zero-shot behavior generalizes beyond V2X-Real.

B.2 Metrics

For V2X-Real, we group the 10 annotated categories into three superclasses: car, pedestrian, and truck. For each IoU threshold, we compute the Average Precision (AP) for each superclass and report the mean Average Precision (mAP) as $\mathrm{mAP} = \frac{1}{3}\left(\mathrm{AP}_{\text{car}} + \mathrm{AP}_{\text{ped}} + \mathrm{AP}_{\text{truck}}\right)$. We report mAP@0.5 and mAP@0.7 at IoU thresholds of 0.5 and 0.7, respectively. For V2XSet, we report vehicle AP@0.5 and AP@0.7 following its evaluation setting.

B.3 Encoder Configurations

We define five encoder configurations based on backbone architecture and voxel size, as summarized in Tab. 3. PP4 is used only for the ego agent, while the remaining four encoder configurations are used for auxiliary agents.

B.4 Training Details

Our training procedure consists of encoder pretraining followed by separate fine-tuning protocols for ALF and baselines. All models are trained and evaluated on an RTX 4090 GPU.

Encoder Pretraining. For each agent configuration, we pretrain three types of checkpoints: (1) single-agent detectors, which are used for late fusion; (2) homogeneous intermediate-fusion models, where all agents share the same encoder configuration; and (3) GenComm Zhou et al. (2025) stage-1 models.

For the standalone single-agent detectors used in late fusion, we train each model for 200 epochs using AdamW Loshchilov & Hutter (2019) with an initial learning rate of 3×10^{-4} and weight decay of 1×10^{-2}. We adopt cosine annealing with warm-up. The warm-up lasts for the first 40 epochs, starts from a learning rate of 2×10^{-4}, and the minimum learning rate is set to 2×10^{-5}.

For the homogeneous intermediate-fusion models and GenComm stage-1 models, we use the same optimizer and scheduler, but train for 40 epochs with warm-up during the first 10 epochs.

ALF Fine-tuning. For ALF, we fine-tune the EFS parameters only once, using box-level messages generated by applying the pretrained PP4 single-agent detector to raw auxiliary observations (Sec. 3.7), while keeping all other components frozen. We set the maximum number of transmitted bounding boxes $K_{\max} = 60$ during training, and use $K_{\max} = 20$ and 30 during inference on V2X-Real and V2XSet, respectively. All transmitted data are quantized from FP32 to INT8. The loss weights are set to $\alpha_{\text{cls}} = 1.0$, $\alpha_{\text{reg}} = 2.0$, $\alpha_{\text{obj}} = 1.0$, $\alpha_{\text{bg}} = 0.5$, $\alpha_{\text{det}} = 1.0$, and $\alpha_{\text{align}} = 1.0$.

Baseline-specific Fine-tuning. During baseline fine-tuning, E2E, MPDA, and CodeFilling initialize the ego and auxiliary branches from the homogeneous intermediate-fusion checkpoints of their corresponding agent configurations, while GenComm initializes from the corresponding stage-1 checkpoints. Each baseline is fine-tuned for 10 epochs using Adam Kingma & Ba (2015) with an initial learning rate of 1×10^{-3}, $\epsilon{=}1\times10^{-10}$, and weight decay of 1×10^{-4}. We use a multi-step decay schedule with decay factor $\gamma{=}0.1$ and a milestone at epoch 5.

During baseline-specific fine-tuning, we follow the freezing strategy of each method. E2E is trained end-to-end without freezing any layers. MPDA and CodeFilling keep the ego and auxiliary encoders frozen, while GenComm updates only the DME and freezes all remaining layers.

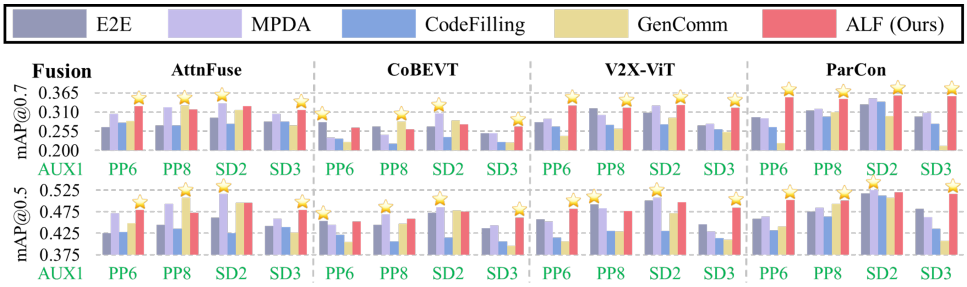
Unlike ALF, which fine-tunes the EFS only once and reuses it across agent configurations, the baselines must be fine-tuned separately for each agent configuration.

B.5 Protocol-Specific Evaluation Details

Below, we describe how the evaluated models are defined and applied under each protocol.

- **Baselines.** For the baselines, protocols (1)–(3) in Sec. 4.1 correspond to protocol-dependent training and evaluation under different auxiliary agent configurations. In the zero-shot unseen setting, to apply a model trained with AUX1 to AUX2, we align the output representation of AUX2 to the auxiliary feature shape used with AUX1 (Tab. 3). Specifically, we match the channel dimension using a frozen 1×1 convolution with orthogonally initialized weights, and resample the BEV spatial resolution with bilinear interpolation.

- **ALF.** ALF does not require protocol-specific training or feature-shape alignment when the auxiliary encoder changes. Therefore, its results are determined only by the auxiliary agent configuration used at inference and are invariant to the protocol label.



(a) Grouped bar chart for seen-pair evaluation

Method	E2E	MPDA	CodeFilling	GenComm	**ALF (Ours)**
# Best@0.7	1	2	0	2	**11**
# Best@0.5	2	5	0	1	**8**
Bandwidth@10Hz (Mbps) ↓	503–1845	503–1845	1.475–2.703	3.932–7.209	**0.0096**

(b) Summary of the number of best-performing cases in terms of mAP@0.7 and mAP@0.5, together with bandwidth. Bandwidth is reported as a range because it varies across PP6, PP8, SD2, and SD3.

Figure 6: Comparison of detection accuracy under seen-pair evaluation. In this setting, both EGO and AUX1 use 128-beam LiDARs.

C Additional Experiments

C.1 Full Results of Seen-Pair Evaluation

Seen-Pair Evaluation. Fig. 6 reports the results of the baseline methods under the seen-pair evaluation protocol and ALF for comparison. Despite its adaptation-free design, ALF remains competitive with SOTA methods that require pair-specific fine-tuning to handle heterogeneity. As shown in Fig. 6(b), ALF achieves the highest mAP@0.5 and mAP@0.7 in 8 and 11 out of 16 cases, respectively. These results are particularly significant considering ALF communicates sub-1KB box-level messages, whereas feature-based SOTAs rely on dense, high-bandwidth representations. Tab. 4 provides the full quantitative results corresponding to Fig. 6.

Communication Bandwidth. The last row of Fig. 6(b) reports the payload bitrate of each method at 10 Hz. The transmitted message size of the baselines varies with the feature size determined by each agent configuration, leading to variable and often high communication costs. In contrast, ALF achieves a fixed and minimal bitrate because its message size is determined only by the fixed box schema and $K_{\max}$, not by the auxiliary feature size. Each message contains $K_{\max} = 20$ boxes, with six INT8-quantized values per box, resulting in a fixed payload of $6 \times 20 \times 8 = 960$ bits, i.e., 120 bytes per frame. At a 10 Hz operating frequency, this corresponds to a bandwidth of only 9.6 kbps (0.0096 Mbps), representing several orders of magnitude reduction compared to feature-based collaboration. Although CodeFilling and GenComm transmit compact intermediate-derived messages, their communication cost remains tied to the BEV feature grid. In contrast, ALF communicates through fixed-size box-level messages, making its payload independent of the auxiliary feature-map resolution.

C.2 Full Results of Zero-Shot Unseen Evaluation

Tab. 5 presents the full quantitative results for the heterogeneous encoder configurations shown in Fig. 3(a).

Table 4: Seen-pair evaluation (EGO+AUX1). The best performance is highlighted in bold, and the second-best is underlined. In this setting, both EGO and AUX1 use 128-beam LiDARs.

Fusion Module	AUX1	E2E mAP@0.5/0.7	MPDA mAP@0.5/0.7	CodeFilling mAP@0.5/0.7	GenComm mAP@0.5/0.7	ALF (Ours) mAP@0.5/0.7
AttnFuse	PP6	0.425 / 0.266	0.471 / 0.305	0.428 / 0.279	0.448 / 0.284	**0.479 / 0.327**
	PP8	0.444 / 0.272	0.493 / 0.323	0.435 / 0.271	**0.507 / 0.329**	0.472 / 0.317
	SD2	0.461 / 0.293	**0.516 / 0.335**	0.425 / 0.276	0.496 / 0.316	0.496 / 0.327
	SD3	0.442 / 0.282	0.459 / 0.304	0.440 / 0.282	0.427 / 0.272	**0.479 / 0.316**
CoBEVT	PP6	**0.453 / 0.281**	0.445 / 0.236	0.421 / 0.234	0.405 / 0.224	0.452 / 0.266
	PP8	0.445 / 0.269	**0.469** / 0.245	0.406 / 0.219	0.447 / **0.282**	0.458 / 0.261
	SD2	0.473 / 0.268	**0.486 / 0.306**	0.415 / 0.238	0.478 / 0.286	0.475 / 0.275
	SD3	0.437 / 0.249	0.443 / 0.250	0.406 / 0.224	0.396 / 0.223	**0.461 / 0.268**
V2X-ViT	PP6	0.457 / 0.281	0.452 / 0.291	0.415 / 0.268	0.406 / 0.241	**0.482 / 0.328**
	PP8	**0.492** / 0.320	0.483 / 0.301	0.430 / 0.273	0.429 / 0.262	0.477 / **0.322**
	SD2	0.501 / 0.307	**0.507** / 0.329	0.430 / 0.275	0.471 / 0.293	0.498 / **0.330**
	SD3	0.446 / 0.271	0.429 / 0.276	0.414 / 0.261	0.410 / 0.252	**0.485 / 0.322**
ParCon	PP6	0.459 / 0.295	0.464 / 0.292	0.432 / 0.267	0.441 / 0.221	**0.503 / 0.352**
	PP8	0.475 / 0.314	0.484 / 0.319	0.464 / 0.297	0.493 / 0.309	**0.501 / 0.347**
	SD2	0.517 / 0.331	**0.525** / 0.348	0.513 / 0.339	0.508 / 0.299	0.520 / **0.357**
	SD3	0.482 / 0.296	0.462 / 0.308	0.436 / 0.276	0.407 / 0.213	**0.516 / 0.355**

C.3 FULL RESULTS OF UNSEEN ADAPTATION EVALUATION

Fig. 7 and Fig. 8 compare ALF with baseline methods in the unseen adaptation setting under encoder-configuration heterogeneity and LiDAR beam-count heterogeneity, respectively.

Under encoder-configuration heterogeneity, although ALF remains fixed after a single training stage, it still outperforms the adapted versions of E2E, MPDA, CodeFilling, and GenComm in 46, 43, 48, and 42 out of 48 AUX1–AUX2–fusion-module combinations, respectively.

We further examine whether adapting to AUX2 preserves performance on the original seen configuration, AUX1. Across the 48 evaluated AUX1–AUX2–fusion-module combinations, the unseen-adapted versions of E2E, MPDA, CodeFilling, and GenComm perform worse than their seen-pair counterparts in 33, 39, 29, and 32 cases, with maximum drops of 15.95%, 14.70%, 14.01%, and 6.84%, respectively.

Under LiDAR beam-count heterogeneity, ALF still outperforms the adapted versions of E2E, MPDA, CodeFilling, and GenComm in 14, 10, 15, and 12 out of 16 AUX1–AUX2–fusion-module combinations, respectively.

However, unlike the results under encoder-configuration heterogeneity, adaptation to the joint EGO+AUX1+AUX2 setting is often beneficial in this case. Across the 16 evaluated AUX1–AUX2–fusion-module combinations, the unseen-adapted versions of E2E, MPDA, CodeFilling, and GenComm perform worse than their seen-pair counterparts in only 2, 5, 9, and 11 cases, with maximum drops of 18.82%, 14.55%, 6.61%, and 18.69%, respectively. This trend suggests that, compared with the encoder-configuration heterogeneity setting, fine-tuning is more often beneficial under LiDAR beam-count heterogeneity, particularly for E2E and MPDA.

Tab. 7 and Tab. 8 report the full quantitative results for Fig. 7, while Tab. 9 and Tab. 10 report those for Fig. 8.

C.4 Qualitative Analysis

Fig. 9 visualizes the pseudo-BEV map $\mathcal{X}_i$, the synthesized feature $\hat{\mathcal{F}}_i$, the teacher feature $\mathcal{F}_i^\star$, and the ego-agent's feature $\mathcal{F}_1$. We also provide the single-agent detection results $\{\hat{\mathcal{B}}_i\}_{i\in\mathcal{A}_{\text{aux}}}$ and the final collaborated ego-centric detection $\hat{\mathcal{B}}_1$. Using EFS, the ego agent transforms object-centric cues from the pseudo-BEV map into ego-compatible latent features by leveraging the geometric and semantic context of $\mathcal{F}_1$. Consequently, the synthesized feature $\hat{\mathcal{F}}_i$ integrates the auxiliary object-centric cue with ego scene context, yielding a fusion-ready representation in the ego latent space.

C.5 Additional Ablation Studies

All ablation studies in this section are conducted under the homogeneous PP4 setting with a 128-beam LiDAR. Fig. 10 analyzes the effects of message quantization precision and box budget. The blue curve compares different quantization precisions (INT4, INT8, INT16, and FP32). INT4 quantization causes a clear performance drop in mAP@0.7, from 0.333 to 0.289, indicating substantial information loss. In contrast, increasing the precision beyond INT8 yields only marginal gains, suggesting that INT8 provides a favorable trade-off between communication efficiency and detection performance.

The yellow curve shows the effect of varying the maximum number of transmitted bounding boxes $K_{\max}$. Without collaboration ($K_{\max} = 0$), the model achieves an mAP@0.7 of 0.295, which increases to 0.333 at $K_{\max} = 20$. Increasing $K_{\max}$ beyond 20 degrades performance, suggesting that transmitting additional low-confidence detections provides limited useful information and may introduce noise.

Tab. 11 presents an ablation study of the EFS components. Directly using the pseudo-BEV map without EFS results in the lowest performance, indicating that the object-centric cue cannot be effectively consumed by the ego latent space as-is. The complete EFS achieves the highest mAP@0.7 while remaining comparable to the best variant at mAP@0.5, supporting the complementary roles of OCE, EIM, and ELR in synthesizing ego-compatible latent features.

C.6 Results on V2XSet

Fig. 11 reports the results on V2XSet using AttnFuse and V2X-ViT. As shown in Fig. 11(a), ALF is less competitive than configuration-specific baselines in the seen-pair setting, where each baseline is trained and evaluated with the same auxiliary configuration. For example, ALF achieves an average AP@0.7 of 0.742 and 0.762 with AttnFuse and V2X-ViT, respectively, compared with 0.837 and 0.840 for E2E.

In contrast, Fig. 11(b) shows a substantially different trend under zero-shot unseen evaluation. ALF achieves the best performance in 20 out of 24 unseen configuration pairs and ties for the best performance in two additional cases. Averaged across the unseen configurations, ALF achieves AP@0.7 of 0.742 with AttnFuse and 0.762 with V2X-ViT, outperforming the strongest competing baseline, GenComm, which achieves 0.719 and 0.724, respectively. These results expose the distinction between pair-specific accuracy and adaptation-free generalization: while ALF does not target the highest accuracy for a fixed, seen agent pair, it maintains strong performance when previously unseen auxiliary configurations are introduced without additional adaptation.

Table 5: Zero-shot unseen evaluation on (EGO+AUX2) with encoder configuration heterogeneity. Models trained/fine-tuned on (EGO+AUX1) are directly evaluated on (EGO+AUX2) without additional training. To enable transfer from AUX1 to AUX2, the output representation of AUX2 is aligned to the auxiliary feature size used with AUX1. The best performance is highlighted in bold. In this setting, EGO, AUX1, and AUX2 are all equipped with 128-beam LiDARs.

Fusion Method	AUX1	AUX2	E2E mAP@0.5/0.7	MPDA mAP@0.5/0.7	CodeFilling mAP@0.5/0.7	GenComm mAP@0.5/0.7	ALF (Ours) mAP@0.5/0.7
AttnFuse	PP6	PP8	0.052 / 0.005	0.379 / 0.240	0.395 / 0.258	0.435 / 0.283	**0.472 / 0.317**
		SD2	0.407 / 0.245	0.443 / 0.284	0.433 / 0.283	0.429 / 0.277	**0.496 / 0.327**
		SD3	0.406 / 0.249	0.444 / 0.283	0.435 / 0.284	0.429 / 0.277	**0.479 / 0.316**
	PP8	PP6	0.079 / 0.011	0.103 / 0.073	0.079 / 0.053	0.437 / 0.284	**0.479 / 0.327**
		SD2	0.404 / 0.248	0.451 / 0.301	0.444 / 0.282	0.435 / 0.283	**0.496 / 0.327**
		SD3	0.399 / 0.250	0.457 / 0.310	0.444 / 0.282	0.434 / 0.282	**0.479 / 0.316**
	SD2	PP6	0.257 / 0.110	0.014 / 0.006	0.446 / 0.289	0.428 / 0.281	**0.479 / 0.327**
		PP8	0.182 / 0.064	0.444 / 0.282	0.446 / 0.285	0.426 / 0.280	**0.472 / 0.317**
		SD3	0.413 / 0.259	0.448 / 0.289	0.284 / 0.194	0.425 / 0.278	**0.479 / 0.316**
	SD3	PP6	0.284 / 0.094	0.004 / 0.002	0.441 / 0.283	0.426 / 0.281	**0.479 / 0.327**
		PP8	0.168 / 0.058	0.448 / 0.280	0.440 / 0.283	0.426 / 0.281	**0.472 / 0.317**
		SD2	0.409 / 0.264	0.374 / 0.247	0.444 / 0.285	0.428 / 0.284	**0.496 / 0.327**
CoBEVT	PP6	PP8	0.008 / 0.001	0.277 / 0.134	0.129 / 0.064	0.338 / 0.191	**0.458 / 0.261**
		SD2	0.266 / 0.140	0.352 / 0.172	0.325 / 0.162	0.342 / 0.200	**0.475 / 0.275**
		SD3	0.274 / 0.161	0.351 / 0.188	0.314 / 0.154	0.346 / 0.203	**0.461 / 0.268**
	PP8	PP6	0.007 / 0.001	0.354 / 0.201	0.130 / 0.073	0.376 / 0.227	**0.452 / 0.266**
		SD2	0.229 / 0.110	0.347 / 0.198	0.250 / 0.144	0.354 / 0.209	**0.475 / 0.275**
		SD3	0.215 / 0.117	0.349 / 0.199	0.280 / 0.162	0.352 / 0.210	**0.461 / 0.268**
	SD2	PP6	0.004 / 0.001	0.330 / 0.174	0.242 / 0.164	0.352 / 0.205	**0.452 / 0.266**
		PP8	0.009 / 0.001	0.303 / 0.157	0.187 / 0.107	0.350 / 0.202	**0.458 / 0.261**
		SD3	0.267 / 0.166	0.374 / 0.211	0.398 / 0.228	0.278 / 0.121	**0.461 / 0.268**
	SD3	PP6	0.007 / 0.001	0.263 / 0.078	0.269 / 0.150	0.346 / 0.201	**0.452 / 0.266**
		PP8	0.013 / 0.001	0.317 / 0.174	0.151 / 0.072	0.377 / 0.224	**0.458 / 0.261**
		SD2	0.244 / 0.144	0.367 / 0.210	0.389 / 0.200	0.314 / 0.153	**0.475 / 0.275**
V2X-ViT	PP6	PP8	0.395 / 0.248	0.372 / 0.238	0.435 / 0.300	0.387 / 0.245	**0.477 / 0.322**
		SD2	0.444 / 0.265	0.441 / 0.274	0.399 / 0.244	0.378 / 0.225	**0.498 / 0.330**
		SD3	0.428 / 0.289	0.450 / 0.284	0.409 / 0.250	0.379 / 0.216	**0.485 / 0.322**
	PP8	PP6	0.287 / 0.172	0.446 / 0.259	0.442 / 0.306	0.375 / 0.225	**0.482 / 0.328**
		SD2	0.425 / 0.265	0.450 / 0.268	0.414 / 0.277	0.368 / 0.218	**0.498 / 0.330**
		SD3	0.418 / 0.262	0.450 / 0.270	0.426 / 0.285	0.388 / 0.242	**0.485 / 0.322**
	SD2	PP6	0.234 / 0.151	0.129 / 0.052	0.273 / 0.179	0.357 / 0.220	**0.482 / 0.328**
		PP8	0.309 / 0.198	0.425 / 0.285	0.393 / 0.269	0.366 / 0.220	**0.477 / 0.322**
		SD3	0.407 / 0.283	0.435 / 0.297	0.344 / 0.218	0.380 / 0.225	**0.485 / 0.322**
	SD3	PP6	0.376 / 0.223	0.285 / 0.178	0.034 / 0.019	0.371 / 0.221	**0.482 / 0.328**
		PP8	0.344 / 0.205	0.419 / 0.257	0.042 / 0.023	0.372 / 0.224	**0.477 / 0.322**
		SD2	0.381 / 0.225	0.425 / 0.273	0.068 / 0.035	0.395 / 0.257	**0.498 / 0.330**
ParCon	PP6	PP8	0.333 / 0.210	0.310 / 0.197	0.129 / 0.064	0.336 / 0.190	**0.501 / 0.347**
		SD2	0.308 / 0.191	0.262 / 0.117	0.325 / 0.162	0.338 / 0.202	**0.520 / 0.357**
		SD3	0.315 / 0.201	0.333 / 0.221	0.314 / 0.154	0.339 / 0.207	**0.516 / 0.355**
	PP8	PP6	0.343 / 0.218	0.147 / 0.039	0.130 / 0.073	0.372 / 0.229	**0.503 / 0.352**
		SD2	0.317 / 0.206	0.403 / 0.282	0.250 / 0.144	0.370 / 0.235	**0.520 / 0.357**
		SD3	0.341 / 0.220	0.401 / 0.282	0.280 / 0.162	0.359 / 0.222	**0.516 / 0.355**
	SD2	PP6	0.387 / 0.250	0.358 / 0.227	0.242 / 0.164	0.358 / 0.220	**0.503 / 0.352**
		PP8	0.368 / 0.228	0.418 / 0.289	0.187 / 0.107	0.333 / 0.204	**0.501 / 0.347**
		SD3	0.344 / 0.225	0.440 / 0.287	0.398 / 0.228	0.328 / 0.204	**0.516 / 0.355**
	SD3	PP6	0.332 / 0.204	0.370 / 0.250	0.269 / 0.150	0.322 / 0.196	**0.503 / 0.352**
		PP8	0.341 / 0.193	0.203 / 0.116	0.151 / 0.072	0.328 / 0.192	**0.501 / 0.347**
		SD2	0.309 / 0.174	0.414 / 0.274	0.389 / 0.200	0.315 / 0.186	**0.520 / 0.357**

Table 6: Zero-shot unseen evaluation on (EGO+AUX2) with LiDAR beam-count heterogeneity. Models trained/fine-tuned on (EGO+AUX1) are directly evaluated on (EGO+AUX2) without additional training. To enable transfer from AUX1 to AUX2, the output representation of AUX2 is aligned to the auxiliary feature size used with AUX1. The best performance is highlighted in bold. In this setting, EGO and AUX1 use 128-beam LiDARs, whereas AUX2 uses 64-beam LiDAR.

Fusion Method	AUX1	AUX2	E2E mAP@0.5/0.7	MPDA mAP@0.5/0.7	CodeFilling mAP@0.5/0.7	GenComm mAP@0.5/0.7	ALF (Ours) mAP@0.5/0.7
AttnFuse	PP6	PP6	0.071 / 0.023	0.378 / 0.261	0.201 / 0.143	0.431 / 0.279	**0.446** / **0.299**
	PP8	PP8	0.047 / 0.020	0.008 / 0.005	0.116 / 0.077	0.432 / 0.282	**0.447** / **0.298**
	SD2	SD2	0.413 / 0.251	0.443 / 0.291	0.441 / 0.285	0.425 / 0.280	**0.466** / **0.307**
	SD3	SD3	0.414 / 0.272	0.433 / 0.276	0.440 / 0.282	0.426 / 0.281	**0.464** / **0.309**
CoBEVT	PP6	PP6	0.007 / 0.001	0.175 / 0.041	0.232 / 0.106	0.339 / 0.201	**0.426** / **0.238**
	PP8	PP8	0.030 / 0.009	0.352 / 0.196	0.172 / 0.097	0.361 / 0.213	**0.408** / **0.234**
	SD2	SD2	0.313 / 0.188	0.381 / 0.227	0.431 / 0.250	0.375 / 0.222	**0.473** / **0.257**
	SD3	SD3	0.348 / 0.184	0.368 / 0.212	0.409 / 0.235	0.363 / 0.205	**0.475** / **0.242**
V2X-ViT	PP6	PP6	0.369 / 0.214	0.364 / 0.221	**0.448** / **0.307**	0.386 / 0.246	0.412 / 0.291
	PP8	PP8	0.391 / 0.252	0.360 / 0.212	**0.443** / **0.308**	0.385 / 0.223	0.410 / 0.271
	SD2	SD2	0.431 / 0.287	0.437 / 0.293	0.424 / 0.278	0.389 / 0.238	**0.441** / **0.324**
	SD3	SD3	**0.437** / 0.278	0.430 / 0.277	0.398 / 0.246	0.352 / 0.206	0.433 / **0.318**
ParCon	PP6	PP6	0.229 / 0.134	0.276 / 0.179	0.426 / 0.285	0.354 / 0.221	**0.468** / **0.323**
	PP8	PP8	0.362 / 0.224	0.416 / 0.273	0.429 / 0.282	0.351 / 0.213	**0.464** / **0.319**
	SD2	SD2	0.409 / 0.258	0.442 / 0.284	0.440 / 0.290	0.332 / 0.199	**0.499** / **0.349**
	SD3	SD3	0.390 / 0.248	0.433 / 0.291	0.436 / 0.276	0.346 / 0.208	**0.488** / **0.336**

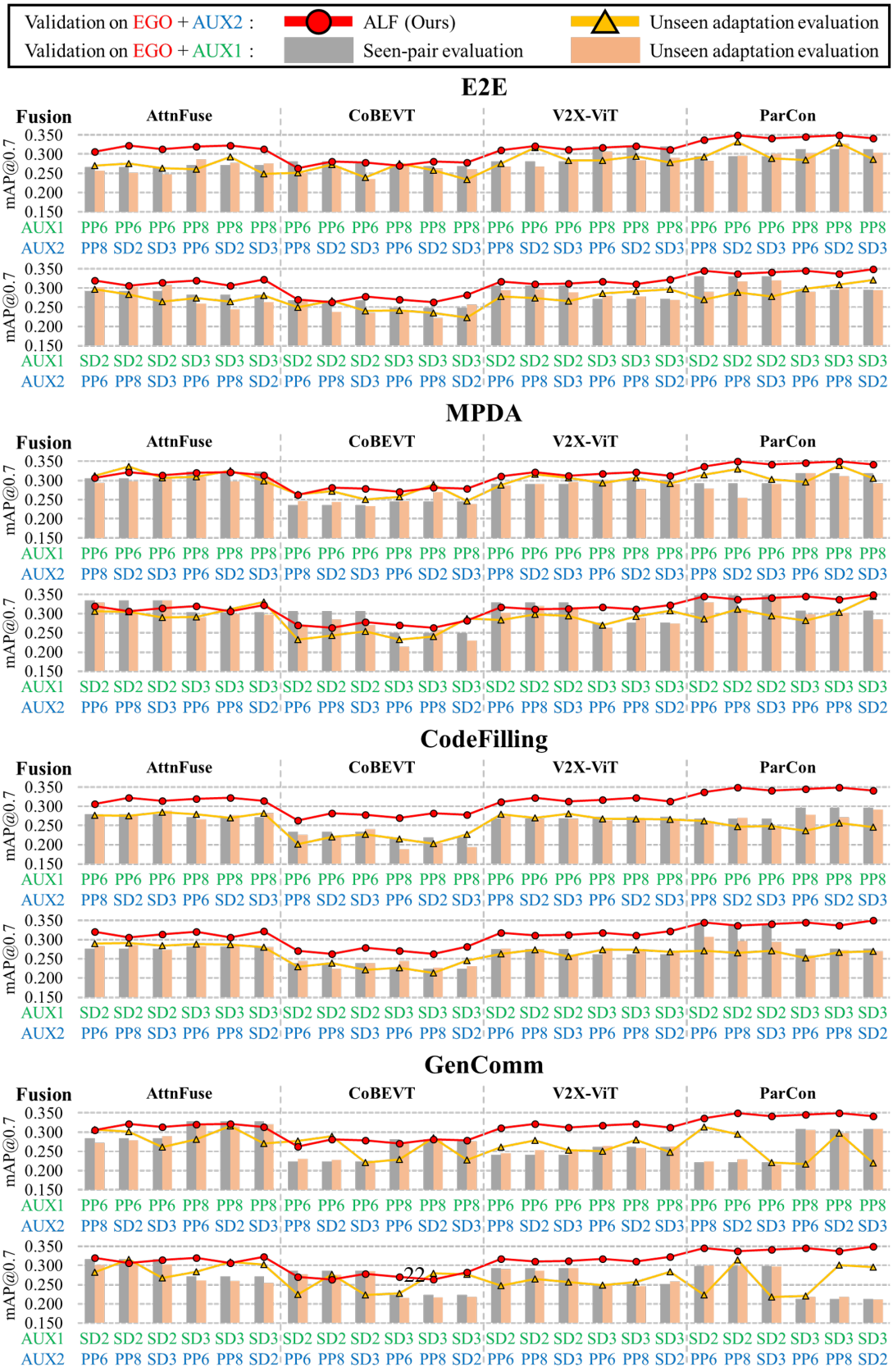


Figure 7: Comparison of mAP@0.7 under seen-pair and unseen-adaptation evaluation with encoder configuration heterogeneity. In this setting, EGO, AUX1, and AUX2 are all equipped with 128-beam LiDARs.

Table 7: Unseen adaptation evaluation on (EGO+AUX2) with encoder configuration heterogeneity. Models trained/fine-tuned on (EGO+AUX1) are further trained/fine-tuned on (EGO+AUX1+AUX2) and then evaluated on (EGO+AUX2). The best performance is highlighted in bold. In this setting, EGO, AUX1, and AUX2 are all equipped with 128-beam LiDARs.

Fusion Method	AUX1	AUX2	E2E mAP@0.5/0.7	MPDA mAP@0.5/0.7	CodeFilling mAP@0.5/0.7	GenComm mAP@0.5/0.7	ALF (Ours) mAP@0.5/0.7
AttnFuse	PP6	PP8	0.433 / 0.271	0.479 / 0.312	0.432 / 0.277	**0.489** / 0.307	0.472 / **0.317**
		SD2	0.460 / 0.276	**0.511** / **0.337**	0.426 / 0.275	0.482 / 0.301	0.496 / 0.327
		SD3	0.424 / 0.263	0.451 / 0.307	0.436 / 0.285	0.407 / 0.261	**0.479** / **0.316**
	PP8	PP6	0.427 / 0.261	0.455 / 0.310	0.435 / 0.279	0.440 / 0.281	**0.479** / **0.327**
		SD2	0.467 / 0.294	**0.500** / 0.326	0.421 / 0.270	0.484 / 0.317	0.496 / **0.327**
		SD3	0.407 / 0.249	0.441 / 0.299	0.443 / 0.282	0.422 / 0.270	**0.479** / **0.316**
	SD2	PP6	0.445 / 0.297	0.455 / 0.307	0.441 / 0.289	0.442 / 0.283	**0.479** / **0.327**
		PP8	0.445 / 0.284	0.483 / 0.304	0.446 / 0.291	**0.493** / 0.316	0.472 / **0.317**
		SD3	0.433 / 0.265	0.444 / 0.290	0.440 / 0.284	0.419 / 0.268	**0.479** / **0.316**
	SD3	PP6	0.427 / 0.274	0.458 / 0.291	0.442 / 0.289	0.446 / 0.283	**0.479** / **0.327**
		PP8	0.440 / 0.264	0.482 / 0.312	0.445 / 0.288	**0.488** / 0.310	0.472 / **0.317**
		SD2	0.457 / 0.281	**0.500** / **0.330**	0.434 / 0.280	0.467 / 0.302	0.496 / 0.327
CoBEVT	PP6	PP8	0.437 / 0.252	**0.477** / 0.264	0.405 / 0.201	0.437 / **0.277**	0.458 / 0.261
		SD2	**0.497** / 0.273	0.492 / 0.272	0.417 / 0.220	0.464 / **0.290**	0.475 / 0.275
		SD3	0.431 / 0.240	**0.464** / 0.250	0.400 / 0.226	0.392 / 0.221	0.461 / **0.268**
	PP8	PP6	**0.468** / **0.276**	0.437 / 0.258	0.405 / 0.215	0.402 / 0.230	0.452 / 0.266
		SD2	0.477 / 0.259	**0.486** / **0.289**	0.405 / 0.203	0.464 / 0.286	0.475 / 0.275
		SD3	0.432 / 0.234	0.427 / 0.247	0.420 / 0.227	0.399 / 0.228	**0.461** / **0.268**
	SD2	PP6	0.415 / 0.249	0.432 / 0.232	0.408 / 0.229	0.407 / 0.224	**0.452** / **0.266**
		PP8	0.450 / 0.267	0.440 / 0.244	0.421 / 0.239	0.444 / **0.277**	**0.458** / 0.261
		SD3	0.406 / 0.240	0.423 / 0.254	0.400 / 0.221	0.389 / 0.224	**0.461** / **0.268**
	SD3	PP6	0.426 / 0.243	0.425 / 0.232	0.413 / 0.226	0.399 / 0.227	**0.452** / **0.266**
		PP8	0.441 / 0.236	0.446 / 0.240	0.395 / 0.213	0.438 / **0.279**	**0.458** / 0.261
		SD2	0.428 / 0.224	**0.500** / **0.288**	0.421 / 0.245	0.467 / 0.277	0.475 / 0.275
V2X-ViT	PP6	PP8	0.472 / 0.276	0.460 / 0.288	0.438 / 0.277	0.432 / 0.261	**0.477** / **0.322**
		SD2	**0.502** / 0.317	0.492 / 0.316	0.416 / 0.269	0.455 / 0.279	0.498 / **0.330**
		SD3	0.451 / 0.284	0.457 / 0.307	0.421 / 0.269	0.410 / 0.253	**0.485** / **0.322**
	PP8	PP6	0.460 / 0.284	0.447 / 0.294	0.444 / 0.282	0.415 / 0.251	**0.482** / **0.328**
		SD2	0.475 / 0.295	**0.500** / 0.307	0.443 / 0.283	0.451 / 0.280	0.498 / **0.330**
		SD3	0.428 / 0.279	0.442 / 0.292	0.412 / 0.263	0.415 / 0.248	**0.485** / **0.322**
	SD2	PP6	0.449 / 0.278	0.452 / 0.283	0.430 / 0.271	0.412 / 0.247	**0.482** / **0.328**
		PP8	0.439 / 0.275	0.461 / 0.298	0.432 / 0.274	0.430 / 0.264	**0.477** / **0.322**
		SD3	0.418 / 0.266	0.446 / 0.294	0.406 / 0.259	0.421 / 0.256	**0.485** / **0.322**
	SD3	PP6	0.452 / 0.286	0.423 / 0.270	0.430 / 0.273	0.410 / 0.249	**0.482** / **0.328**
		PP8	0.462 / 0.292	0.470 / 0.292	0.428 / 0.269	0.427 / 0.257	**0.477** / **0.322**
		SD2	0.471 / 0.297	0.488 / 0.307	0.412 / 0.261	0.461 / 0.284	**0.498** / **0.330**
ParCon	PP6	PP8	0.450 / 0.294	0.479 / 0.315	0.440 / 0.261	0.489 / 0.314	**0.501** / **0.347**
		SD2	0.518 / 0.332	0.515 / 0.329	0.407 / 0.246	0.495 / 0.294	**0.520** / **0.357**
		SD3	0.475 / 0.289	0.470 / 0.302	0.437 / 0.249	0.420 / 0.221	**0.516** / **0.355**
	PP8	PP6	0.454 / 0.285	0.455 / 0.297	0.387 / 0.236	0.434 / 0.217	**0.503** / **0.352**
		SD2	**0.527** / 0.330	0.508 / 0.339	0.425 / 0.256	0.499 / 0.298	0.520 / **0.357**
		SD3	0.460 / 0.286	0.481 / 0.305	0.418 / 0.246	0.424 / 0.220	**0.516** / **0.355**
	SD2	PP6	0.452 / 0.270	0.465 / 0.286	0.435 / 0.271	0.435 / 0.223	**0.503** / **0.352**
		PP8	0.466 / 0.289	0.473 / 0.311	0.422 / 0.265	0.492 / 0.315	**0.501** / **0.347**
		SD3	0.455 / 0.278	0.454 / 0.294	0.425 / 0.271	0.417 / 0.218	**0.516** / **0.355**
	SD3	PP6	0.447 / 0.298	0.431 / 0.282	0.411 / 0.252	0.431 / 0.221	**0.503** / **0.352**
		PP8	0.476 / 0.309	0.471 / 0.303	0.430 / 0.267	0.478 / 0.301	**0.501** / **0.347**
		SD2	0.515 / 0.322	0.501 / 0.345	0.441 / 0.269	0.494 / 0.296	**0.520** / **0.357**

Table 8: Unseen adaptation evaluation on (EGO+AUX1) with encoder configuration heterogeneity. Models trained/fine-tuned on (EGO+AUX1) are further trained/fine-tuned on (EGO+AUX1+AUX2) and then evaluated on (EGO+AUX1). The best performance is highlighted in bold. In this setting, EGO, AUX1, and AUX2 are all equipped with 128-beam LiDARs.

Fusion Method	AUX1	AUX2	E2E mAP@0.5/0.7	MPDA mAP@0.5/0.7	CodeFilling mAP@0.5/0.7	GenComm mAP@0.5/0.7	ALF (Ours) mAP@0.5/0.7
AttnFuse	PP6	PP8	0.410 / 0.257	0.456 / 0.293	0.430 / 0.276	0.438 / 0.273	**0.472** / **0.317**
		SD2	0.400 / 0.252	0.452 / 0.298	0.433 / 0.277	0.448 / 0.279	**0.496** / **0.327**
		SD3	0.391 / 0.247	0.455 / 0.306	0.441 / 0.289	0.449 / 0.290	**0.479** / **0.316**
	PP8	PP6	0.454 / 0.287	0.486 / 0.310	0.419 / 0.266	**0.496** / 0.319	0.479 / **0.327**
		SD2	0.458 / 0.277	0.486 / 0.298	0.438 / 0.277	0.492 / 0.315	**0.496** / **0.327**
		SD3	0.428 / 0.275	0.495 / 0.305	0.445 / 0.282	**0.495** / **0.320**	0.479 / 0.316
	SD2	PP6	0.473 / 0.302	**0.515** / **0.329**	0.429 / 0.283	0.487 / 0.301	0.479 / 0.327
		PP8	0.476 / 0.304	**0.506** / 0.301	0.435 / 0.286	0.491 / 0.310	0.472 / **0.317**
		SD3	0.476 / 0.308	**0.511** / **0.334**	0.424 / 0.273	0.488 / 0.303	0.479 / 0.316
	SD3	PP6	0.445 / 0.260	0.454 / 0.291	0.434 / 0.283	0.418 / 0.260	**0.479** / **0.327**
		PP8	0.451 / 0.245	0.448 / 0.299	0.439 / 0.285	0.409 / 0.259	**0.472** / **0.317**
		SD2	0.434 / 0.263	0.452 / 0.296	0.434 / 0.280	0.404 / 0.253	**0.496** / **0.327**
CoBEVT	PP6	PP8	0.444 / 0.252	**0.459** / 0.247	0.424 / 0.225	0.403 / 0.231	0.458 / **0.261**
		SD2	0.442 / 0.268	0.428 / 0.244	0.417 / 0.222	0.401 / 0.227	**0.475** / **0.275**
		SD3	0.424 / 0.236	0.447 / 0.233	0.423 / 0.241	0.403 / 0.225	**0.461** / **0.268**
	PP8	PP6	**0.472** / 0.270	0.452 / 0.247	0.400 / 0.188	0.433 / **0.271**	0.452 / 0.266
		SD2	0.437 / 0.262	**0.475** / 0.269	0.419 / 0.204	0.445 / **0.278**	0.475 / 0.275
		SD3	**0.474** / 0.261	0.465 / 0.238	0.397 / 0.193	0.442 / **0.276**	0.461 / 0.268
	SD2	PP6	0.475 / 0.267	**0.502** / 0.269	0.443 / 0.245	0.470 / **0.277**	0.452 / 0.266
		PP8	0.465 / 0.238	**0.492** / **0.284**	0.405 / 0.224	0.476 / 0.275	0.458 / 0.261
		SD3	0.437 / 0.235	**0.474** / 0.270	0.423 / 0.239	0.470 / **0.285**	0.461 / 0.268
	SD3	PP6	0.447 / 0.238	0.423 / 0.213	0.407 / 0.244	0.390 / 0.215	**0.452** / **0.266**
		PP8	0.436 / 0.221	0.430 / 0.258	0.398 / 0.226	0.390 / 0.217	**0.458** / **0.261**
		SD2	0.450 / 0.256	0.445 / 0.230	0.414 / 0.230	0.395 / 0.217	**0.475** / **0.275**
V2X-ViT	PP6	PP8	0.464 / 0.268	0.433 / 0.287	0.415 / 0.273	0.406 / 0.245	**0.477** / **0.322**
		SD2	0.425 / 0.268	0.443 / 0.291	0.411 / 0.267	0.411 / 0.253	**0.498** / **0.330**
		SD3	0.445 / 0.285	0.471 / 0.296	0.415 / 0.268	0.411 / 0.251	**0.485** / **0.322**
	PP8	PP6	**0.486** / 0.307	0.472 / 0.301	0.425 / 0.263	0.430 / 0.264	0.482 / **0.328**
		SD2	0.444 / 0.284	0.453 / 0.278	0.430 / 0.265	0.432 / 0.258	**0.498** / **0.330**
		SD3	0.449 / 0.290	0.462 / 0.291	0.438 / 0.266	0.440 / 0.263	**0.485** / **0.322**
	SD2	PP6	0.483 / 0.295	**0.491** / 0.303	0.434 / 0.276	0.469 / 0.290	0.482 / **0.328**
		PP8	0.480 / 0.297	**0.499** / 0.320	0.435 / 0.275	0.465 / 0.287	0.477 / **0.322**
		SD3	0.467 / 0.288	**0.498** / 0.317	0.422 / 0.260	0.476 / 0.292	0.485 / **0.322**
	SD3	PP6	0.452 / 0.280	0.418 / 0.264	0.428 / 0.277	0.410 / 0.252	**0.482** / **0.328**
		PP8	0.462 / 0.278	0.439 / 0.289	0.420 / 0.271	0.414 / 0.246	**0.477** / **0.322**
		SD2	0.432 / 0.269	0.431 / 0.274	0.414 / 0.271	0.419 / 0.258	**0.498** / **0.330**
ParCon	PP6	PP8	0.444 / 0.282	0.436 / 0.278	0.442 / 0.257	0.437 / 0.223	**0.501** / **0.347**
		SD2	0.465 / 0.296	0.423 / 0.255	0.431 / 0.269	0.438 / 0.229	**0.520** / **0.357**
		SD3	0.477 / 0.301	0.457 / 0.291	0.443 / 0.243	0.428 / 0.216	**0.516** / **0.355**
	PP8	PP6	0.484 / 0.303	0.482 / 0.319	0.443 / 0.278	0.486 / 0.305	**0.503** / **0.352**
		SD2	0.495 / 0.326	0.475 / 0.311	0.450 / 0.271	0.489 / 0.297	**0.520** / **0.357**
		SD3	0.462 / 0.304	0.472 / 0.292	0.465 / 0.292	0.487 / 0.308	**0.516** / **0.355**
	SD2	PP6	0.492 / 0.290	**0.507** / 0.329	0.479 / 0.307	0.492 / 0.299	0.503 / **0.352**
		PP8	0.499 / 0.317	0.484 / 0.313	0.470 / 0.297	**0.503** / 0.298	0.501 / **0.347**
		SD3	0.501 / 0.320	0.509 / 0.345	0.463 / 0.293	0.496 / 0.296	**0.516** / **0.355**
	SD3	PP6	0.458 / 0.290	0.455 / 0.298	0.418 / 0.258	0.411 / 0.218	**0.503** / **0.352**
		PP8	0.471 / 0.301	0.458 / 0.303	0.432 / 0.273	0.418 / 0.218	**0.501** / **0.347**
		SD2	0.472 / 0.294	0.457 / 0.285	0.442 / 0.269	0.412 / 0.212	**0.520** / **0.357**

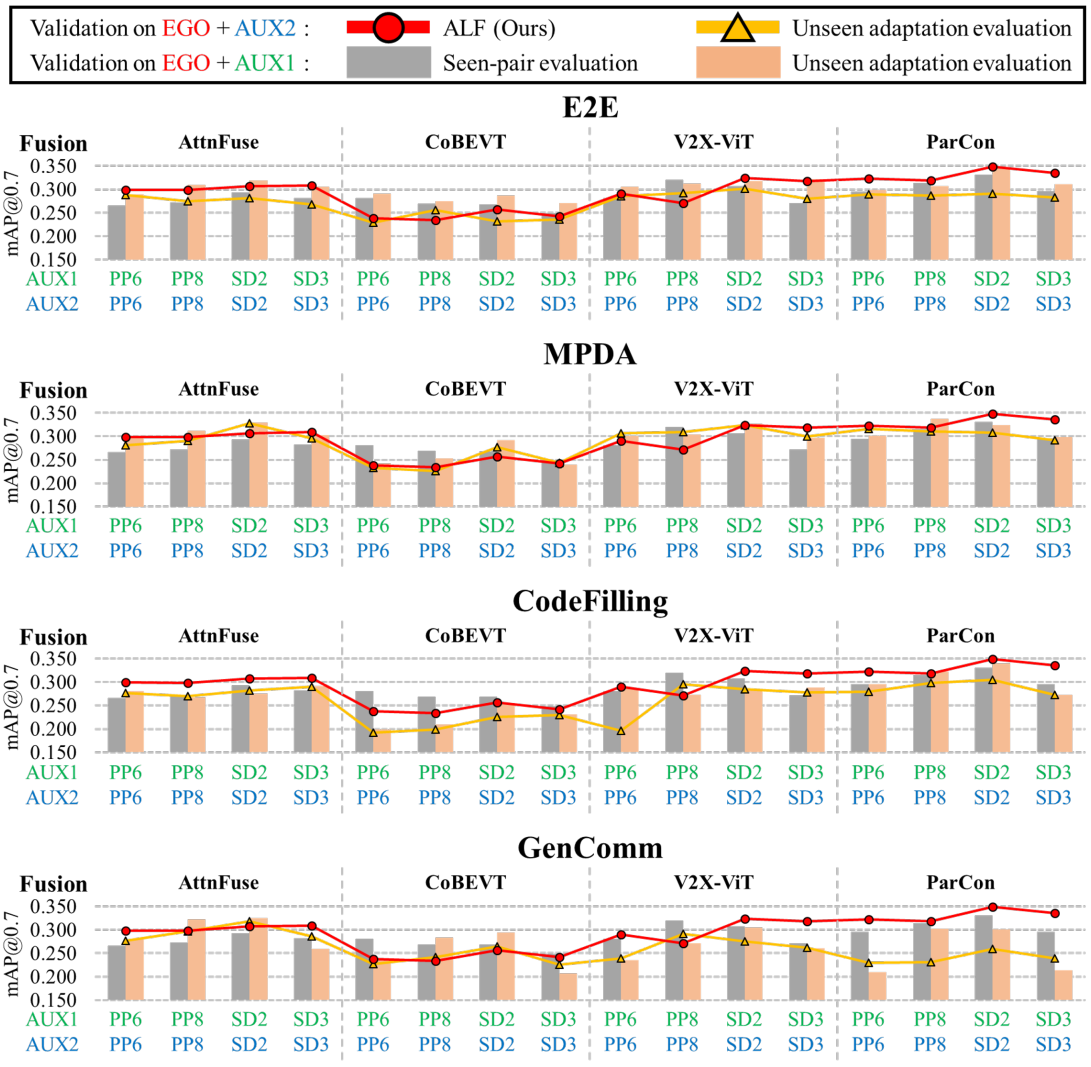


Figure 8: Comparison of mAP@0.7 under seen-pair and unseen-adaptation evaluation with LiDAR beam-count heterogeneity. In this setting, EGO and AUX1 use 128-beam LiDARs, whereas AUX2 uses 64-beam LiDAR.

Table 9: Unseen adaptation evaluation on (EGO+AUX2) with LiDAR beam-count heterogeneity. Models trained/fine-tuned on (EGO+AUX1) are further trained/fine-tuned on (EGO+AUX1+AUX2) and then evaluated on (EGO+AUX2). The best performance is highlighted in bold. In this setting, EGO and AUX1 use 128-beam LiDARs, whereas AUX2 uses 64-beam LiDAR.

Fusion Method	AUX1	AUX2	E2E mAP@0.5/0.7	MPDA mAP@0.5/0.7	CodeFilling mAP@0.5/0.7	GenComm mAP@0.5/0.7	ALF (Ours) mAP@0.5/0.7
AttnFuse	PP6	PP6	**0.459** / 0.288	0.450 / 0.281	0.429 / 0.277	0.439 / 0.276	0.446 / **0.299**
	PP8	PP8	0.427 / 0.274	0.456 / 0.290	0.437 / 0.270	**0.473** / 0.297	0.447 / **0.298**
	SD2	SD2	0.451 / 0.282	**0.492** / **0.328**	0.443 / 0.282	0.481 / 0.319	0.466 / 0.307
	SD3	SD3	0.419 / 0.267	**0.470** / 0.295	0.444 / 0.290	0.455 / 0.286	0.464 / **0.309**
CoBEVT	PP6	PP6	0.408 / 0.229	0.408 / 0.233	0.375 / 0.192	0.381 / 0.227	**0.412** / **0.238**
	PP8	PP8	**0.424** / **0.256**	0.402 / 0.226	0.369 / 0.199	0.416 / 0.241	0.410 / 0.234
	SD2	SD2	0.409 / 0.231	**0.484** / **0.276**	0.389 / 0.225	0.445 / 0.265	0.441 / 0.257
	SD3	SD3	0.377 / 0.235	0.414 / **0.244**	0.403 / 0.229	0.373 / 0.226	**0.433** / 0.242
V2X-ViT	PP6	PP6	0.436 / 0.285	**0.468** / **0.307**	0.338 / 0.196	0.386 / 0.239	0.426 / 0.291
	PP8	PP8	0.460 / 0.292	**0.462** / **0.309**	0.453 / 0.295	0.442 / 0.291	0.408 / 0.271
	SD2	SD2	0.479 / 0.302	**0.490** / **0.324**	0.435 / 0.285	0.437 / 0.275	0.473 / 0.324
	SD3	SD3	0.436 / 0.281	0.456 / 0.300	0.416 / 0.277	0.405 / 0.262	**0.475** / **0.318**
ParCon	PP6	PP6	0.439 / 0.289	**0.477** / 0.316	0.446 / 0.280	0.404 / 0.229	0.468 / **0.323**
	PP8	PP8	0.444 / 0.287	**0.475** / 0.310	0.456 / 0.298	0.444 / 0.231	0.464 / **0.319**
	SD2	SD2	0.467 / 0.291	0.485 / 0.307	0.475 / 0.305	0.451 / 0.259	**0.499** / **0.349**
	SD3	SD3	0.430 / 0.283	0.446 / 0.291	0.432 / 0.273	0.406 / 0.239	**0.488** / **0.336**

Table 10: Unseen adaptation evaluation on (EGO+AUX1) with LiDAR beam-count heterogeneity. Models trained/fine-tuned on (EGO+AUX1) are further trained/fine-tuned on (EGO+AUX1+AUX2) and then evaluated on (EGO+AUX1). The best performance is highlighted in bold. In this setting, EGO and AUX1 use 128-beam LiDARs, whereas AUX2 uses 64-beam LiDAR.

Fusion Method	AUX1	AUX2	E2E mAP@0.5/0.7	MPDA mAP@0.5/0.7	CodeFilling mAP@0.5/0.7	GenComm mAP@0.5/0.7	ALF (Ours) mAP@0.5/0.7
AttnFuse	PP6	PP6	**0.455** / 0.290	0.453 / 0.298	0.439 / 0.280	0.449 / 0.279	0.446 / **0.299**
	PP8	PP8	0.490 / 0.309	0.486 / 0.312	0.433 / 0.267	**0.495** / **0.323**	0.447 / 0.298
	SD2	SD2	0.502 / 0.319	**0.507** / **0.330**	0.429 / 0.275	0.503 / 0.325	0.466 / 0.307
	SD3	SD3	0.475 / 0.306	**0.481** / 0.302	0.445 / 0.291	0.431 / 0.259	0.464 / **0.309**
CoBEVT	PP6	PP6	**0.461** / **0.291**	0.412 / 0.242	0.389 / 0.199	0.402 / 0.239	0.412 / 0.238
	PP8	PP8	**0.466** / 0.275	0.451 / 0.253	0.409 / 0.210	0.445 / **0.283**	0.410 / 0.234
	SD2	SD2	0.487 / 0.286	**0.498** / 0.292	0.423 / 0.256	0.475 / **0.294**	0.441 / 0.257
	SD3	SD3	**0.454** / **0.270**	0.431 / 0.239	0.400 / 0.231	0.400 / 0.207	0.433 / 0.242
V2X-ViT	PP6	PP6	**0.463** / **0.305**	0.447 / 0.301	0.444 / 0.287	0.406 / 0.235	0.426 / 0.291
	PP8	PP8	**0.479** / **0.312**	0.470 / 0.304	0.436 / 0.273	0.440 / 0.271	0.408 / 0.271
	SD2	SD2	0.502 / 0.318	**0.507** / **0.326**	0.448 / 0.282	0.476 / 0.304	0.473 / 0.324
	SD3	SD3	**0.479** / **0.322**	0.454 / 0.297	0.435 / 0.289	0.412 / 0.260	0.475 / 0.318
ParCon	PP6	PP6	0.454 / 0.299	**0.477** / 0.301	0.476 / 0.296	0.432 / 0.210	0.468 / **0.323**
	PP8	PP8	0.469 / 0.307	**0.517** / **0.337**	0.497 / 0.322	0.489 / 0.303	0.464 / 0.319
	SD2	SD2	**0.543** / 0.346	0.508 / 0.324	0.512 / 0.340	0.506 / 0.301	0.499 / **0.349**
	SD3	SD3	**0.492** / 0.311	0.465 / 0.298	0.433 / 0.273	0.414 / 0.213	0.488 / **0.336**

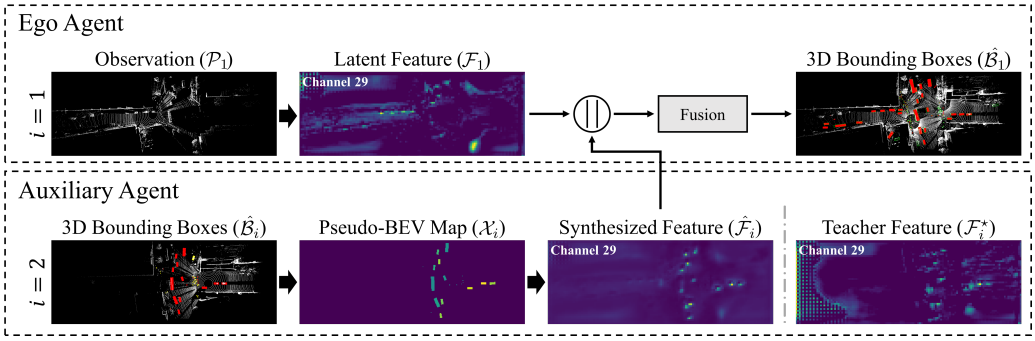


Figure 9: Visualization results of ALF using AttnFuse. The 3D bounding boxes shown in red, green and yellow denote the predicted, ground-truth, and filtered out bounding boxes, respectively.

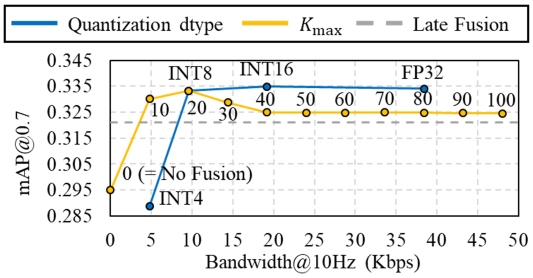
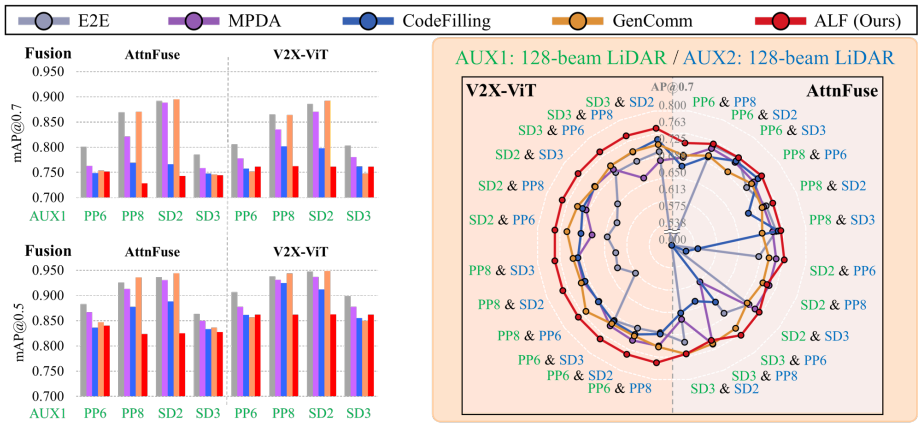


Figure 10: Ablation study on quantization schemes and maximum number of transmitted boxes $K_{\max}$.

Table 11: Ablation study on EFS using different combinations of its submodules.

OCE	EIM	ELR	mAP@0.5	mAP@0.7
			0.452	0.298
✓			0.470	0.305
✓	✓		0.483	0.313
✓		✓	**0.487**	0.318
✓	✓	✓	0.485	**0.333**



(a) Seen-Pair Evaluation (b) Zero-Shot Evaluation

Figure 11: Evaluation results on V2XSet using AttnFuse and V2X-ViT: (a) seen-pair evaluation and (b) zero-shot unseen evaluation.