# QuantV2X: A Fully Quantized Multi-Agent System for Cooperative Perception

Seth Z. Zhao[1⋆], Huizhi Zhang[1,2*†], Zhaowei Li[3], Juntong Peng[4], Anthony Chui[1], Zewei Zhou[1], Zonglin Meng[1‡], Hao Xiang[1], Zhiyu Huang[1], Fujia Wang[5], Ran Tian[5], Chenfeng Xu[5,6], Bolei Zhou[1], and Jiaqi Ma[1]

[1]UCLA [2]UW-Madison [3]NCSU [4]Purdue University [5]UC Berkeley [6]UT Austin

**Abstract.** Cooperative perception through Vehicle-to-Everything (V2X) communication offers significant potential for enhancing vehicle perception by mitigating occlusions and expanding the field of view. However, past research has predominantly focused on improving accuracy metrics without addressing the crucial system-level considerations of efficiency, latency, and real-world deployability. Noticeably, most existing systems rely on full-precision models, which incur high computational and transmission costs, making them impractical for real-time operation in resource-constrained environments. In this paper, we introduce **QuantV2X**, the first fully quantized multi-agent system designed specifically for efficient and scalable deployment of multi-modal, multi-agent V2X cooperative perception. QuantV2X introduces a unified end-to-end quantization strategy across both neural network models and transmitted message representations that simultaneously reduces computational load and transmission bandwidth. Remarkably, despite operating under low-bit constraints, QuantV2X achieves accuracy comparable to full-precision systems. More importantly, when evaluated under deployment-oriented metrics, QuantV2X reduces system-level latency by 3.2× and achieves a +9.5 improvement in mAP30 over full-precision baselines. Furthermore, QuantV2X scales more effectively, enabling larger and more capable models to fit within strict memory budgets. These results highlight the viability of a fully quantized multi-agent intermediate fusion system for real-world deployment. The system will be publicly released to promote research in this field: `https://github.com/ucla-mobility/QuantV2X`.

## 1 Introduction

Vehicle-to-Everything (V2X) cooperative perception has emerged as a promising paradigm for enabling safe and intelligent autonomous driving [7, 8, 26]. By allowing autonomous agents to share real-time sensor information, it creates a collective perception system that extends beyond the field of view of any single vehicle, significantly enhancing situational awareness for all agents [23,25]. Despite

⋆ Equal contribution. Project lead contact: `sethzhao506@g.ucla.edu`.
† Work done when Huizhi Zhang was a research intern at UCLA.
‡ Corresponding author: `meng925@g.ucla.edu`.

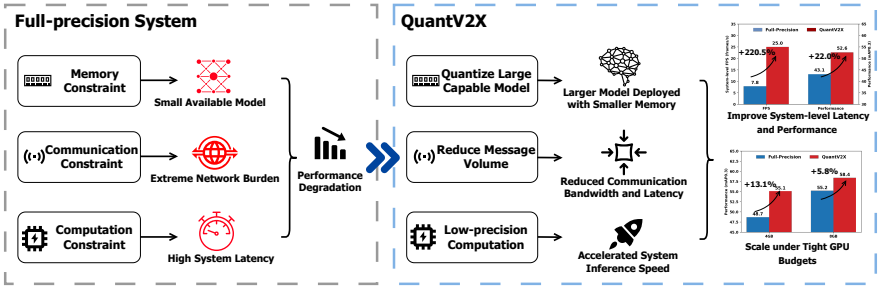


**Fig. 1:** Motivation. *Left:* Full-precision cooperative perception systems are ill-suited for real-world deployment. *Right:* QuantV2X presents an efficient and scalable solution for real-world cooperative driving systems.

remarkable progress in model design and accuracy improvements, most prior work has been developed under full-precision (FP32) assumptions, leading to prohibitive computational, memory, and communication costs. As illustrated in Fig. 1, full-precision systems cannot be accommodated within the tight memory budgets of in-vehicle GPUs without aggressive model size reduction, which inevitably sacrifices the model's expressiveness and causes substantial performance degradation. Moreover, even when such models are deployed, the high inference latency of full-precision networks, compounded by the transmission overhead of sharing FP32 BEV features, introduces prohibitive system-level delays. These intertwined bottlenecks highlight a critical gap between algorithmic advances in cooperative perception and their practical feasibility for real-world autonomous driving deployments.

To bridge this gap, we present **QuantV2X**, a fully quantized multi-agent cooperative system designed to holistically address the system-level latency and performance drop in resource-constrained V2X settings (shown in Fig. 1). Our core insight is that full-precision representations in both local computation and agent-to-agent communication dominate end-to-end latency and lead to downstream performance drop. Building on this, QuantV2X delivers a full-stack recipe encompassing both model-side and communication-side efficiency. On the model side, we propose a post-training quantization (PTQ) process that transforms pretrained full-precision models into compact low-bit networks while maintaining competitive accuracy. To mitigate the challenges brought by quantization-induced feature degradation, we introduce a novel alignment module that jointly corrects spatial misalignment and feature distribution shifts among heterogeneous agents. On the communication side, we replace costly FP32 BEV feature transmission with compact low-bit messages, where each agent transmits only the code indices of a shared codebook. These indices act as quantized representations of the full feature map, allowing the receiver to reconstruct high-fidelity features locally while significantly reducing communication bandwidth during collaborations. Together, under real-world latency constraints, QuantV2X reduces end-to-end system latency by **3.2**× compared to the full-precision system with +**9.5** mAP30 performance improvements in the V2X-Real dataset. These

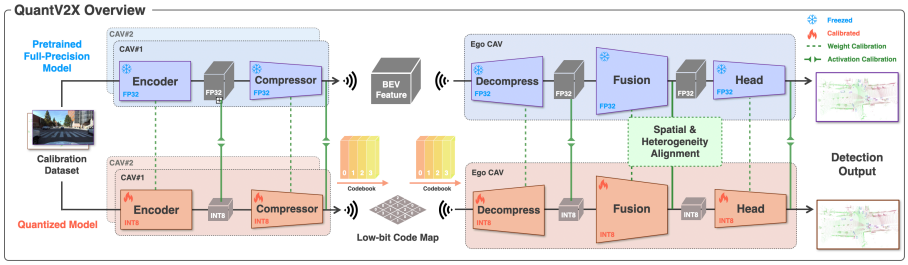


**Fig. 2:** QuantV2X overview. On the model side, QuantV2X transforms full-precision neural networks into compact low-bit representations, reducing computational overhead without sacrificing accuracy. On the communication side, it replaces bandwidth-heavy floating-point feature maps with quantized message representations, enabling efficient collaborations under strict transmission budgets.

results demonstrate that QuantV2X effectively overcomes system-level efficiency bottlenecks and enables real-time, high-performance V2X cooperative perception.

The experimental sections move beyond the conventional accuracy-centric paradigm, focusing on evaluating the holistic performance of the system in realistic deployment scenarios. In Section 3.1, we show that QuantV2X maintains the perception ability of full-precision systems, preserving up to 99.8% of their accuracy even under INT4 weight and INT8 activation quantization. In Section 3.2, we further show that QuantV2X consistently surpasses full-precision systems when evaluated under system-level latency, highlighting its real-world efficiency. Finally, in Section 3.3, we illustrate how quantized deployment enables larger and more capable models to run on edge platforms without exceeding resource budgets, thereby expanding both system capacity and performance. Additional real-world evaluation results (see Appendix E) on system's performance on real-world testbed [17] and power consumption further demonstrate the real-world efficiency impact of QuantV2X. Collectively, these results position QuantV2X as a practical and scalable pathway towards fully deployable multi-agent systems for V2X cooperative perception.

**Contribution.** In this work, we address the problems of inefficiency and performance degradation for cooperative perception in real-world resource-constrained scenarios. We illustrate the system-level latency bottleneck in full-precision systems and introduce **QuantV2X**, *a fully quantized multi-agent system for cooperative perception* that enables efficient model inference and multi-agent communication with maximum perception performance preservation while meeting the requirements of real-world deployment. To the best of our knowledge, this is the first work to demonstrate the viability and practicality of a fully quantized intermediate fusion system for future real-world deployment.

## 2 Methodology

### 2.1 QuantV2X: A System Overview

As shown in Fig. 2, QuantV2X presents a fully quantized system that unifies efficiency at both the model level and the communication level. QuantV2X consists of three stages: (i) **a full-precision pretraining stage**, where we train a full-precision (FP32) cooperative perception model that serves as the foundation for subsequent quantization, (ii) **a codebook learning stage**, where the model learns quantized transmission feature representations for communication-efficient collaboration, and (iii) **a post-training quantization (PTQ) stage**, where both full-precision models and features are converted into low-bit formats with minimal accuracy degradation.

### 2.2 Full-Precision Pretraining

In this stage, we adopt an intermediate fusion architecture. Let $b$ represent an agent type. For the $i$th agent in the set $S_{[b]}$, we define $f_{\text{encoder}}[b]$ as its perception encoder, input $\mathbf{O}_i$ as its raw input (RGB images or LiDAR point clouds) and $\mathbf{B}_i$ as its final detection output. The operation of the $i$th agent works as follows:

$$\mathbf{F}_i = f_{\text{encoder}[b]}\left(\mathbf{O}_i\right), \quad \mathbf{F}_{j\to i} = \Gamma_{j\to i}\left(\mathbf{F}_j\right), \quad \mathbf{H}_i = f_{\text{fusion}}\left(\left\{\mathbf{F}_{j\to i}\right\}_{j\in S_{[b]}}\right), \quad \mathbf{B}_i = f_{\text{head}}\left(\mathbf{H}_i\right),$$

where $\mathbf{F}_i$ is the initial BEV feature map produced by the encoder, $\Gamma_{j\to i}(\cdot)$ is an operator that transmits $j$th agent's feature to the $i$th agent and performs spatial transformation, $\mathbf{F}_{j\to i}$ is the spatially aligned BEV feature in $i$th's coordinate frame (note that $\mathbf{F}_{i\to i} = \mathbf{F}_i$), $\mathbf{H}_i$ is the fused feature and $\mathbf{B}_i$ is the final detection output obtained by a detection head $f_{\text{head}}(\cdot)$. This stage learns a complete FP32 model, parameterized by $f_{\text{encoder}[b]}, f_{\text{fusion}}, f_{\text{head}}$.

### 2.3 Codebook Learning

**Quantized Message Representation via Codebook** The transmission of FP32 BEV features poses major challenges for cooperative perception, incurring high bandwidth and computation costs on resource-limited hardware, as evidenced by recent deployments [17]. Motivated by the approaches proposed in [3,5], we employ a codebook-based messaging approach for collaborative agents and introduce a novel quantization-aware codebook learning method optimized for fully quantized systems. The codebook can be seen as a dictionary data structure represented by {codebook index : codebook feature}. Formally, we define the codebook as a learnable dictionary $\mathcal{D} = \{d_1, d_2, \ldots, d_{n_L}\} \in \mathbb{R}^{C\times n_L}$, where each entry $d_\ell \in \mathbb{R}^C$ represents a $C$-dimensional feature vector and $n_L$ represents the number of codes. The codebook is shared across all agents and serves as a compressed basis for BEV features. During collaboration, each agent transmits only the codebook indices to other agents instead of transmitting BEV feature maps. Regarding communication volume, for a BEV feature of dimensions

$H \times W \times C$, the original communication bandwidth requirement can be computed as $\log_2(H \times W \times C \times 32/8)$, where the number 32 indicates FP32 data representation, and the division by 8 converts bits to bytes. In contrast, when employing the codebook index representation with a codebook, the bandwidth is reduced as $\log_2(H \times W \times \log_2(n_L) \times n_R/8)$, where $\log_2(n_L)$ denotes the number of bits required to represent each code index integer, and $n_R$ indicates the number of codes used.

During transmission, given a BEV feature vector $F_{[h,w]} \in \mathbb{R}^C$ at spatial location $(h, w)$, the nearest code in the dictionary is selected via:

$$\text{index}_{[h,w]} = \arg \min_{\ell \in \{1,2,\ldots,n_L\}} \left\| F_{[h,w]} - d_\ell \right\|_2^2. \tag{1}$$

When using multiple codes per location ($n_R > 1$), the reconstructed feature vector $\hat{F}_{[h,w]}$ is computed as a weighted combination of selected codes:

$$\hat{F}_{[h,w]} = \sum_{r=1}^{n_R} \alpha_r \cdot d_{\text{index}_r}, \tag{2}$$

where $\alpha_r$ are the combination weights and $\{\text{index}_r\}_{r=1}^{n_R}$ are the corresponding selected code indices. The combination weight is generated by computing the distances between input feature segments and all codebook entries, converting these distances to logits, and applying Gumbel-Softmax to produce soft, differentiable weights during training (which become hard one-hot selections during inference), which are then used to reconstruct the quantized feature as a weighted combination of the selected codes from each codebook group.

**Codebook Training Strategy** We train the codebook in two stages. In the first stage, we randomly initialize $\mathcal{D}$ and freeze all other model parameters. Given frozen BEV features $F \in \mathbb{R}^{H \times W \times C}$ extracted from the encoder pretrained from full-precision models in Section 2.2, we assign each spatial position $(h, w)$ to one or more code indices via a nearest-neighbor assignment (argmin of squared Euclidean distance) within a product-quantized codebook. The learning objective becomes the following.

$$\min_{\Theta_{\text{cb}}} \sum_{(h,w)} \left\| F_{[h,w]} - \hat{F}_{[h,w]} \right\|_2^2, \tag{3}$$

where $\Theta_{\text{cb}}$ denotes all the parameters within the codebook module.

In the second stage, we unfreeze all model parameters and jointly optimize the encoder, fusion module, detection head, and codebook. The encoder is now trained to produce BEV features $F$ that are naturally quantizable with $\mathcal{D}$. At each forward pass, the BEV features are quantized to $\hat{F}$ using the current codebook, and the detection is performed on the quantized representation. The joint objective becomes:

$$\min_{\theta,\, \mathcal{D}} \; \mathcal{L}_{\text{det}}(\hat{B}, B^{\text{gt}}) + \lambda_{\text{rec}} \sum_{(h,w)} \left\| F_{[h,w]} - \hat{F}_{[h,w]} \right\|_2^2, \tag{4}$$

where $\theta$ denotes all the model parameters excluding $\mathcal{D}$, $\mathcal{L}_{\text{det}}$ denotes the standard detection loss, $\hat{B}$ denotes the detection output computed from quantized features $\hat{F}$, $B^{\text{gt}}$ denotes the ground-truth bounding boxes, and $\lambda_{\text{rec}}$ controls the weight of the reconstruction term.

### 2.4 Post-Training Quantization

The goal of the post-training quantization stage is to convert the full-precision model into a low-bit format while minimizing performance degradation. This stage only requires a small fraction of calibration data and does not need to retrain the whole model. We quantize both individual tensors and full network modules, leveraging deployment-friendly techniques compatible with inference engines like TensorRT [12]. Unlike prior work [24], which only partially quantizes the network, we apply end-to-end quantization across the entire pipeline (from the encoder and fusion module to the detection decoder) to achieve a fully quantized system.

**Preliminaries: Quantization for Tensors** Quantization maps floating-point (FP) values $x$ (e.g., weights or activations) to low-precision integer approximations $x_{\text{int}}$ following:

$$x_{\text{int}} = \text{clamp}\left(\left\lfloor \frac{x}{s} \right\rceil + z, q_{\min}, q_{\max}\right), \tag{5}$$

where $\lfloor \cdot \rceil$ denotes rounding-to-nearest integer, introducing rounding error $\Delta_r$; $z$ denotes the zero-point and $s$ denotes the scale factor defined as:

$$s = \frac{q_{\max} - q_{\min}}{2^b - 1}, \tag{6}$$

where $b$ is the target bit-width. The clamp operation ensures the result lies within the quantization range $[q_{\min}, q_{\max}]$, introducing a clipping error $\Delta_c$. The dequantized approximation of the original FP values is obtained via:

$$\hat{x} = s \cdot (x_{\text{int}} - z). \tag{7}$$

**QuantV2X Calibration Procedure** Calibration is essential in our fully quantized system to ensure that the transition from full-precision to quantized models does not hurt performance. The calibration procedure is outlined in Algorithm 1. We first construct a sampled subset from the training dataset as a calibration dataset. To address the variability in cooperative interactions, we introduce a multi-agent sampling strategy that randomly samples agent combinations and communication patterns for the construction of the calibration dataset. By exposing the model to varying numbers and configurations of interacting agents, we ensure that the quantization parameters are robustly calibrated to reflect the dynamic and heterogeneous nature of real-world cooperative perception.

As calibration begins, we initialize both weight and activation quantization using the Max-min calibration strategy, which defines the quantization range

based on the observed minimum and maximum values of the input tensor. This strategy aims to preserve the fine-grained structure of sparse point cloud features while remaining effective for RGB features [24]. Given an input tensor $X$, the quantization range is set to $[X_{\max}, X_{\min}]$ and the initial quantization scale $s_0$ is then computed using Eq. 6. To enable fine-grained calibration, we linearly discretize a range around the initial scale factor $s_0$, forming a set of candidate quantization scales $\{s_t\}_{t=1}^{T}$ within the interval $[\alpha s_0, \beta s_0]$. The hyperparameters $\alpha$, $\beta$, and $T$ control the search span and resolution. We then select the optimal quantization scale $s_{\text{opt}}$ by minimizing the reconstruction error between the original and quantized representations:

$$s_{\text{opt}} = \arg\min_{s_t} \|X - \hat{X}(s_t)\|_F^2, \tag{8}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\hat{X}(s_t)$ represents the quantized tensor under scale $s_t$. To further reduce the rounding error $\Delta_r$, we adopt a learnable rounding strategy inspired by AdaRound [13], introducing an auxiliary variable for each weight element to adaptively select rounding directions.

During calibration, we propose a block-wise reconstruction strategy that minimizes block-level discrepancies between full-precision and quantized outputs, reducing the interface errors that arise with per-layer calibration. This strategy is applied across the entire multi-agent system, including $f_{\text{encoder}[b]}, f_{\text{fusion}}, f_{\text{head}}$, to keep the quantized system aligned with the full-precision reference. For multi-agent fusion $f_{\text{fusion}}$, we add an alignment module within the intermediate fusion layers to preserve cross-agent feature consistency during fusion.

**Alignment Module** The fusion module serves as the central component of cooperative perception models, where features from all agents are aggregated into a unified representation. However, this process is particularly vulnerable to quantization noise. Directly applying conventional quantization techniques at this stage often leads to compounded feature misalignment that distorts the fused representation. For example, as illustrated in Fig. 3, naive linear quantization introduces a significant distribution shift relative to the full-precision model, ultimately harming downstream perception performance. To mitigate this quantization-induced degradation, we propose an alignment module that addresses two key sources of misalignment in cooperative perception scenarios: (i) sensor modality and architecture heterogeneity - differences in sensors (i.e., RGB and LiDAR point cloud) and encoder backbone (i.e., PointPillar [6] and SECOND [21]), and (ii) spatial discrepancies arising from real-world deployment issues such as transmission latency and pose noise due to temporal asynchrony. The alignment module mainly applies at the fusion stage with the following formulations:

**Heterogeneity Alignment Loss.** Heterogeneity among agents introduces ambiguity in activation range scaling during the calibration process. To encourage consistency between full-precision and quantized fused feature maps, we introduce

**Algorithm 1** QuantV2X Calibration

**Input**: Pretrained FP model with $N$ blocks, Calibration dataset $D^c$, Iteration $T$. Blocks denote the perception network components (e.g., backbone, fusion module, downstream head).

**Output**: Quantization parameters of both activation and weight in the network: weight scale $s_w$, weight zero-point $z_w$, activation scale $s_a$, and activation zero-point $z_a$.

**for** $B_n = \{B_i | i = 1, 2, ...N\}$ **do**
  Initialize weight parameters $s_w$ and $z_w$ of each layer in $B_n$ using Eq.( 6);
**end for**
Use weight quantization parameters to formulate a mirrored Quantized model with N blocks;
Input $D^c$ to FP model to collect final output prediction $O_{fp}$;
**for** $B_n$, $B_n^q = \{B_i, B_i^q | i = 1, 2, ...N\}$ where $B_n^q$ belongs to quantized model **do**
  Input $D^c$ into both FP and Quantized models and collect block input $I^q$ from $B_i^q$ and block output $A_i$ from $B_i$;
  Input $I^q$ into $B_i^q$ to initialize activation parameters $s_a$ and $z_a$ using Eq. ( 6);
  **for** all $j = 1, 2, ..., T$ iteration **do**
    Input $I^q$ to $B_i^q$ to get block output $\hat{A}_i$;
    Optimize parameters $s_w$, $z_w$, $s_a$, and $z_a$ of block $B_i^q$ using Eq.( 8);
    **if** $B_i$ belongs to the fusion module **then**
      Input $\hat{A}_i$ to the following FP network to get output $\hat{O}_{par}$ of partial-quantized network;
      Check $\hat{A}_i$ and $A_i$ to calculate $\mathcal{L}_{\text{hetero}}$ to perform heterogeneity alignment using Eq.( 9);
      Check $\hat{O}_{par}$ and $O_{fp}$ to calculate $\mathcal{L}_{\text{spatial}}$ to perform spatial alignment using Eq.( 10);
      Optimize parameters $s_w$, $z_w$, $s_a$, and $z_a$ of layer $B_i^q$ to minimize $\mathcal{L}_{\text{hetero}}$ and $\mathcal{L}_{\text{spatial}}$;
    **end if**
  **end for**
**end for**

a heterogeneity alignment loss based on the Kullback-Leibler (KL) divergence:

$$\mathcal{L}_{\text{hetero}} = D_{\text{KL}} \left( \mathbf{H}_i^{\text{fp}} \parallel \mathbf{H}_i^{\text{int}} \right), \tag{9}$$

where $\mathbf{H}_i^{\text{fp}}$ and $\mathbf{H}_i^{\text{int}}$ denote the respective fused BEV features from the full-precision and quantized models.

**Spatial Alignment Loss.** Precision loss introduced by the quantization process increases the sensitivity of the final detection output to real-world noises. To reduce the discrepancy in detection outputs due to spatial misalignment, we define a spatial alignment loss using L2 loss over the predicted bounding box distributions:

$$\mathcal{L}_{\text{spatial}} = \left\| \mathcal{B}_i^{\text{fp}} - \mathcal{B}_i^{\text{int}} \right\|_2^2, \tag{10}$$

where $\mathcal{B}_i^{\text{fp}}$ and $\mathcal{B}_i^{\text{int}}$ denote the respective bounding box representations from the full-precision and quantized models.

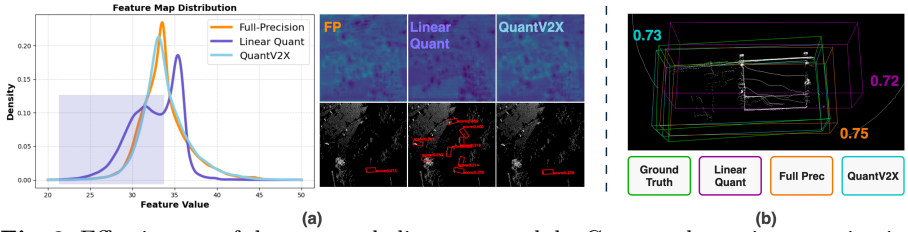


**Fig. 3:** Effectiveness of the proposed alignment module. Compared to naive quantization, (a) $\mathcal{L}_{\text{hetero}}$ leads to fewer false positive detections, and (b) $\mathcal{L}_{\text{spatial}}$ enables the quantized model to output 3D bounding boxes with more precise coordinates and higher confidence score.

## 3 Experiments

We evaluate QuantV2X on a suite of tasks to answer the following research questions: 1) Can QuantV2X preserve perception accuracy under aggressive low-bit quantization? 2) Does QuantV2X effectively reduce real-world system-level latency and improve overall performance? 3) Can QuantV2X enable larger and more capable models under constrained GPU memory budgets?

### 3.1 Model-level Experiments

**Experiment Protocols.** The main goal of the model-level experiments is to evaluate the performance of our PTQ process described in Section 2.4. To assess the effectiveness of our method in recovering spatial features, we exclude the compressor module in the cooperative perception model, which will be analyzed separately in the system-level experiments presented in Section 3.2.

**Datasets.** QuantV2X is evaluated with two real-world datasets, namely DAIR-V2X [22] and V2X-Real [16], and one simulation dataset OPV2V [20]. DAIR-V2X exhibits one vehicle and one infrastructure, both equipped with a LiDAR with different channel numbers and a $1920 \times 1080$ camera. V2X-Real is a large-scale, real-world V2X dataset that encompasses all V2X collaboration modes with two vehicles and two roadside units. Following previous protocols [11, 16, 17], the evaluation metric is presented as Average Precision (AP) with different intersection-over-union (IoU) thresholds. Additional evaluations on other datasets are presented in the Appendix D.2.

**Implementation Details.** We follow [11] and define the following notations for different agent modalities. $\mathbf{L_P}$ denotes an agent with LiDAR sensor using the PointPillar [6] backbone, and $\mathbf{L_S}$ denotes an agent with LiDAR sensor using the SECOND [21] backbone. $\mathbf{C_R}$ denotes a camera-based agent with Lift-Splat-Shoot [14] projection deployed and a ResNet50 model as the image

encoder. Pyramid Fusion [11] is the main intermediate fusion method for our experiments, as it has the best perception performance and fastest inference time. All experiments are calibrated using 0.5% of the original training data. Additional experiments on ablation study of calibration settings and comparisons with other SOTA methods [9, 24] are presented in the Appendix D.2.

**Generalizability across different fusion methods** Table 1 shows our PTQ method generalizes well across various fusion methods, including computation-based fusion methods [2, 20], CNN-based fusion methods [10, 11], and attention-based fusion methods [4, 19]. Detailed analysis of the quantization effect on each fusion method is discussed in Appendix D.3.

**Table 1:** Generalizability of QuantV2X across different fusion methods. Results displayed in terms of AP30/50 on DAIR-V2X dataset (collaboration mode: $\mathbf{L_P}$ + $\mathbf{C_R}$).

| Bits (W/A) | Pyramid Fusion | F-Cooper | AttFuse | V2X-ViT | Who2com | Where2comm |
|---|---|---|---|---|---|---|
| 32/32 | 75.1/68.2 | 64.5/56.0 | 68.8/63.1 | 57.4/49.5 | 63.2/57.3 | 62.1/53.7 |
| 8/8 | 74.6/67.8 | 62.9/55.4 | 67.0/61.9 | 40.0/11.0 | 59.1/54.2 | 59.5/51.8 |
| 4/8 | 74.2/66.7 | 57.4/49.5 | 66.6/60.8 | 29.9/8.8 | 57.2/52.8 | 60.4/51.5 |

**Component analysis** We begin by analyzing the individual components of QuantV2X to quantify their contributions in Table 2. It can be observed that the alignment module boosts the performance recovery from 97.4% to 98.8% and 95.2% to 99.8% for $\mathbf{L_P}$ + $\mathbf{C_R}$ and $\mathbf{L_P}$ + $\mathbf{L_S}$ settings in terms of AP30, respectively. This component-wise evaluation leads to two key observations:

**(i) QuantV2X preserves perception capability under heterogeneous settings.** Applying a basic channel-wise linear quantization method (as described in Eq. 5) leads to a significant drop in precision and results in blurred BEV feature boundaries, as shown in Fig. 3 (a). In contrast, our heterogeneity alignment loss aligns the activation range of BEV features from heterogeneous inputs, producing sharper BEV feature maps and reducing false positives.

**Table 2:** Component Analysis of QuantV2X in DAIR-V2X dataset. Bits (W/A) is set to INT4/8.

| Method | AP30/50 | |
|---|---|---|
| | $\mathbf{L_P}$ + $\mathbf{C_R}$ | $\mathbf{L_P}$ + $\mathbf{L_S}$ |
| Full-Precision | 75.1/68.2 | 80.3/76.1 |
| Max-min | 73.2/61.5 | 76.5/60.1 |
| +AdaRound [13] | 72.8/65.1 | 80.1/74.2 |
| +$\mathcal{L}_{\text{hetero}}$ | 74.0/66.4 | 80.8/75.3 |
| +$\mathcal{L}_{\text{spatial}}$ | 74.2/66.7 | 80.2/75.5 |

**(ii) QuantV2X demonstrates strong robustness under noisy environments.** As illustrated in Fig. 4, we evaluated the robustness of our method against localization error in the DAIR-V2X dataset. We follow the standard evaluation protocols [11, 20, 25] and use sample noises from Gaussian distribution added to the ground truth pose of each collaborating agent (positional or heading error). Under extreme settings, QuantV2X maintains performance comparable to full-precision models. Notably, it also preserves far-range detection capability. This finding highlights the importance of incorporating a spatial alignment loss during the calibration process. Without this design, the vanilla

linear quantization method fails significantly under noisy conditions. Fig. 3 (b) visualizes that the spatial alignment loss further refines the 3D bounding box predictions by correcting their coordinates.
More experimental results on the effectiveness of alignment module on higher-precision (e.g., AP70) and longer-range (e.g., >50m) are presented in Appendix D.5.

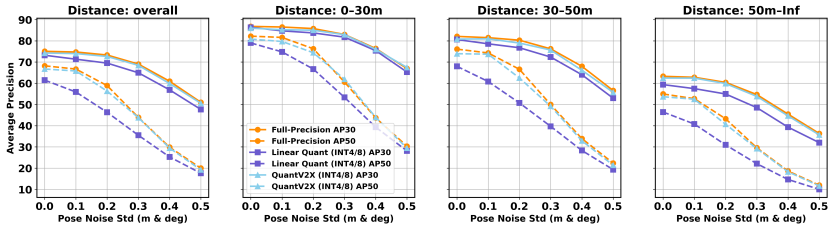


**Fig. 4:** Robustness under pose errors (Collaboration mode: $\mathbf{L_P}$ + $\mathbf{C_R}$, Bits (W/A) are set to INT4/8). Note that the vanilla quantization method suffers from considerable precision loss when the pose error is enlarged.

### 3.2 System-level Experiments

**Experiment Protocols.** The goal of system-level experiments is to examine the performance of the quantized system considering system-level latency, including local inference latency, multi-agent communication latency, and fusion inference latency. This differs from the model-level experiments in Section 3.1, which assume the multi-agent system is ideal and well-synchronized. In the system-level setting, the cooperative perception models incorporate a compressor module for transmission. For QuantV2X, we employ the quantized message representation described in Section 2, while full-precision baselines transmit compressed BEV features unless otherwise noted. Detailed testing settings and comparisons of power consumption are reported in the Appendix E.
**Implementation Details.** Pyramid Fusion [11] is our main fusion method as it has the best perception performance and the fastest inference time. We only consider PointPillar [6] models as each agent's backbone to be consistent with benchmarking in V2X-Real and V2X-ReaLO. The evaluations of the quantized system are conducted in terms of INT8 weight and INT8 activation to be consistent with the previous protocol [24]. For the codebook setting, we set $n_L$ to 128 and $n_R$ to 1. More details on the ablation study of $n_L$ and $n_R$ selections are presented in Appendix E.4.

**System-level Latency Measurement** To show the inference efficiency improvements of QuantV2X under real-world V2X testing environments, we evaluate the system-level latency of QuantV2X using the ROS and TensorRT platform [12, 17] and compare it against a full-precision baseline. The end-to-end system-level

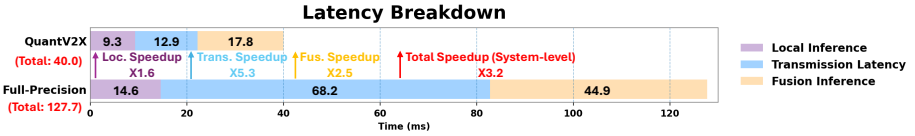


**Fig. 5:** System-level latency breakdown (unit: ms). Note that the numbers are obtained through averaging multiple runs in real-world deployment environment.

**Table 3:** System-level performance comparisons among different systems in V2X-Real Dataset. $\Delta$ denotes the difference with the Upper-bound, which assumes an ideal setting without considering system-level latency and transmission feature compression.

| System | Transmission Feature/Size | mAP30/50 | $\Delta$ |
|---|---|---|---|
| Upper-bound | - | 53.8/43.5 | - |
| Full-Precision [11] | BEV Feature/8.6 MB (No Compression) | 43.1/34.8 | -10.7/-8.7 |
| | BEV Feature/0.54 MB (×16 Compression) | 48.8/38.0 | -5.0/-5.5 |
| Where2Comm [4] | BEV Feature/0.30 MB (×16 Compression) | 49.7/39.0 | -4.1/-4.5 |
| CodeFilling [5] | Codebook/0.03 MB | 51.4/40.8 | -2.4/-2.7 |
| QuantV2X (Ours) | Codebook/0.03 MB | **52.6/42.2** | -1.2/-1.3 |

latency ($T_{\text{sys}}$) of a cooperative perception system can be decomposed into three primary components: (i) local inference latency ($T_{\text{local}}$), representing the time each agent takes to process its own sensor data; (ii) communication latency ($T_{\text{comm}}$), the time required to transmit information between agents; and (iii) fusion inference latency ($T_{\text{fus}}$), the time taken to process received data and generate a final perception output. A detailed testing environment is provided in Appendix E. As illustrated in Fig. 5, quantization significantly reduces latency across all components: $T_{\text{local}}$, $T_{\text{comm}}$, and $T_{\text{fus}}$. These gains stem from low-precision computation for model inference and reduced communication payload between agents. In the following sections, the impact of these improvements on system-level performance is further analyzed.

**System-level Performance Evaluations Evaluation Setting.** To simulate the impact of system-level latency under realistic conditions, we follow the protocols in [1, 15, 18, 19]. The total system latency is defined as $T_{\text{sys}} = T_{\text{local}} + T_{\text{comm}} + T_{\text{fus}}$, where $T_{\text{local}}$ and $T_{\text{fus}}$ are obtained from Fig. 5 and $T_{\text{comm}}$ is calculated according to the transmission delay formula established by previous protocols [1, 15, 18, 19]. The communication latency is calculated as $T_{\text{comm}} = f_s/v + \mathcal{U}(0, 200)$, whereas $f_s$ is the feature size and $v$ denotes the transmission rate (which is set to 27 Mbps according to [1, 19]) and $\mathcal{U}$ denotes the system-wise asynchronous delay following a uniform distribution between 0 and 200 ms. All experiments are conducted on the V2X-Real dataset to remain consistent with the measurements in Section 3.2.

**System-level experimental results.** We compare the system-level performance of full-precision systems, Where2Comm [4], CodeFilling [5], and QuantV2X. Notably, both full-precision systems and Where2Comm are affected by latency at both the model and communication levels, whereas CodeFilling is more heavily

impacted by model-level inefficiency. Table 3 presents that our method consistently outperforms the full-precision system due to the significant system-level latency reduction. Furthermore, the comparison with CodeFilling [5] emphasizes the critical role of reducing inference latency to the whole multi-agent system. These results demonstrate that in dynamic scenarios, the *information timeliness* advantage brought by low latency is sufficient to compensate for and even surpass the minor accuracy loss introduced by quantization, underscoring the importance of system-level optimization. Additional real-world experiments on ROS-based V2X-ReaLO platform [17] is presented in Appendix E..

**Decomposition analysis of system-level latency and performance.** To rigorously disentangle the impact of model-level and communication-level inefficiencies on system performance, we conduct a component-wise decomposition analysis as shown in Fig. 6. We decouple the total system-level latency into computation latency ($T_{\text{comp}} = T_{\text{local}} + T_{\text{fus}}$) and communication latency ($T_{\text{comm}}$) and study their corresponding impacts on final system-level performance:

**(i) Impact of model-level efficiency ($T_{\text{comp}}$).** As illustrated in Fig. 6(a), model quantization addresses the computational bottleneck. For Pyramid Fusion, this reduces computation overhead ($T_{\text{comp}}$) by approximately 55% (59.5ms $\rightarrow$ 27.1ms). Crucially, this step yields a secondary benefit: the bit-width reduction (FP32 $\rightarrow$ INT8) simultaneously lowers transmission latency ($T_{\text{comm}}$) from 286.8ms to 87.0ms. This aggregate latency reduction drives immediate accuracy gains (43.1 $\rightarrow$ 48.7 mAP30), validating that efficient low-precision computation enhances overall system performance.

**(ii) Impact of communication-level efficiency ($T_{\text{comm}}$).** Complementing quantization, our communication-level optimization further reduces $T_{\text{comm}}$ from 286.8ms to 12.9ms (Pyramid Fusion). When integrated with model quantization, the total system latency drops significantly from 346.3ms to 40.0ms. This efficiency enables QuantV2X to achieve 52.6 mAP30, approaching the upper-bound of 53.8 and demonstrating the necessity of optimizing both modules. In particular, these trends are observed across all evaluated architectures (Pyramid Fusion [11], F-Cooper [2], AttFuse [20]), where QuantV2X delivers the highest accuracy and lowest latency.

### 3.3 Scaling Behavior of QuantV2X under GPU Resource Budgets

We examine the scaling behavior of QuantV2X by varying the backbone capacity of both the full-precision baseline and QuantV2X under different GPU memory budgets for common in-vehicle GPUs, as shown in Fig. 7. For each memory budget, we allocate the largest feasible model that can fit within the available resources. Once memory limits are imposed, larger backbones in the full-precision systems cannot be accommodated without downsizing the model, which leads to noticeable performance degradation. In contrast, QuantV2X effectively bridges this gap by compressing larger models into compact low-bit representations that remain within device-level memory constraints while still achieving high perception accuracy. This capability demonstrates that QuantV2X not only alleviates efficiency bottlenecks but also fundamentally enables *scalability under*

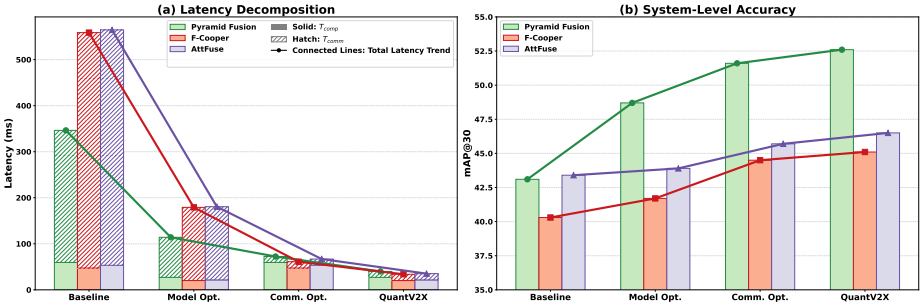


**Fig. 6: System-Level Efficiency and Accuracy Decomposition Analysis.** We evaluate four distinct system configurations: *Baseline* (FP32), *Model Opt.* (Quantization only), *Comm. Opt.* (Codebook only), and the fully unified *QuantV2X*. **(a) Latency Decomposition:** The stacked bars decompose total latency into computation ($T_{\text{comp}}$) and communication ($T_{\text{comm}}$). The trend illustrates the reduction in total latency. Note that Model Opt. specifically lowers the computation floor, while Comm. Opt. alleviates the transmission bottleneck. **(b) System-Level Accuracy:** The corresponding detection performance across three fusion architectures (Pyramid Fusion, F-Cooper, AttFuse). These trends highlight that QuantV2X delivers the highest accuracy with lowest total latency and is generalizable to different fusion architectures.

*resource constraints*. By unlocking the potential to deploy larger and more accurate cooperative perception models on edge devices, QuantV2X provides a practical pathway to scaling state-of-the-art cooperative perception in real-world resource-constrained settings.

## 4 Conclusion

In this work, we introduce QuantV2X, a fully quantized multi-agent system designed to tackle the system-level inefficiencies prevalent in cooperative perception. QuantV2X achieves substantial reductions in cumulative system latency and communication overhead while maintaining competitive perception performance relative to ideal full-precision baselines. Our findings highlight the potential of quantized multi-agent systems as a practical and scalable solution for resource-constrained deployment for V2X cooperative perception. We advocate for reframing cooperative perception research around system-level efficiency, latency, and deployability, a perspective we show is critical for transitioning V2X from research prototypes to scalable real-world deployment. As future directions, we aim to deploy QuantV2X in real-world Cellular-V2X testbeds to conduct comprehensive evaluations under practical deployment conditions.

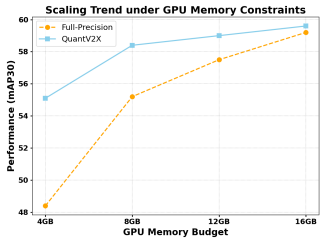


**Fig. 7:** Scaling trend under different GPU memory constraints. QuantV2X enables deployment of larger models under tight budgets while maintaining high perception performance.

## Acknowledgements

This work was supported by the Federal Highway Administration Center of Excellence on New Mobility and Automated Vehicles, and by the National Science Foundation under Award No. 2346267, POSE: Phase II - DriveX: An Open Source Ecosystem for Automated Driving and Intelligent Transportation Research.

## References

1. Arena, F., Pau, G.: An overview of vehicular communications. Future Internet **11**(2) (2019). https://doi.org/10.3390/fi11020027, https://www.mdpi.com/1999-5903/11/2/27
2. Chen, Q., Ma, X., Tang, S., Guo, J., Yang, Q., Fu, S.: 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. pp. 88–100 (2019)
3. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. arXiv: Computer Vision and Pattern Recognition (2015), https://api.semanticscholar.org/CorpusID:2134321
4. Hu, Y., Fang, S., Lei, Z., Zhong, Y., Chen, S.: Where2comm: communication-efficient collaborative perception via spatial confidence maps. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. NIPS '22, Curran Associates Inc., Red Hook, NY, USA (2022)
5. Hu, Y., Peng, J., Liu, S., Ge, J., Liu, S., Chen, S.: Communication-Efficient Collaborative Perception via Information Filling with Codebook . In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15481–15490. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2024). https://doi.org/10.1109/CVPR52733.2024.01466, https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.01466
6. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12697–12705 (2019)
7. Lei, M., Zhou, Z., Li, H., Ma, J., Hu, J.: Risk map as middleware: Towards interpretable cooperative end-to-end autonomous driving for risk-aware planning. arXiv preprint arXiv:2508.07686 (2025)
8. Li, Y., Ren, S., Wu, P., Chen, S., Feng, C., Zhang, W.: Learning distilled collaboration graph for multi-agent perception. Advances in Neural Information Processing Systems **34**, 29541–29552 (2021)
9. Liu, L., Li, S., Gong, Y., Yang, X., Zhang, W., Hu, X.: Pd-quant: Post-training quantization based on prediction difference metric. arXiv preprint arXiv:2212.07048 (2022), https://arxiv.org/abs/2212.07048
10. Liu, Y.C., Tian, J., Ma, C.Y., Glaser, N., Kuo, C.W., Kira, Z.: Who2com: Collaborative perception via learnable handshake communication. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 6876–6883 (2020). https://doi.org/10.1109/ICRA40945.2020.9197364

11. Lu, Y., Hu, Y., Zhong, Y., Wang, D., Chen, S., Wang, Y.: An extensible framework for open heterogeneous collaborative perception. In: The Twelfth International Conference on Learning Representations (2024)
12. Migacz, S.: 8-bit inference with tensorrt. In: GPU Technology Conference (2017), https://www.cse.iitd.ac.in/~rijurekha/course/tensorrt.pdf, presentation
13. Nagel, M., Amjad, R., Van Baalen, M., Louizos, C., Blankevoort, T.: Up or down? adaptive rounding for post-training quantization. arXiv preprint arXiv:2004.10568 (2020), https://arxiv.org/abs/2004.10568
14. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: Proceedings of the European Conference on Computer Vision (2020)
15. Rauch, A., Klanner, F., Dietmayer, K.: Analysis of v2x communication parameters for the development of a fusion architecture for cooperative perception systems. In: 2011 IEEE Intelligent Vehicles Symposium (IV). pp. 685–690 (2011). https://doi.org/10.1109/IVS.2011.5940479
16. Xiang, H., Zheng, Z., Xia, X., Xu, R., Gao, L., Zhou, Z., Han, X., Ji, X., Li, M., Meng, Z., et al.: V2x-real: a largs-scale dataset for vehicle-to-everything cooperative perception. arXiv preprint arXiv:2403.16034 (2024)
17. Xiang, H., Zheng, Z., Xia, X., Zhao, S.Z., Gao, L., Zhou, Z., Cai, T., Zhang, Y., Ma, J.: V2x-realo: An open online framework and dataset for cooperative perception in reality. arXiv preprint arXiv:2503.10034 (2025)
18. Xu, R., Chen, C.J., Tu, Z., Yang, M.H.: V2x-vitv2: Improved vision transformers for vehicle-to-everything cooperative perception. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(1), 650–662 (2025). https://doi.org/10.1109/TPAMI.2024.3479222
19. Xu, R., Xiang, H., Tu, Z., Xia, X., Yang, M.H., Ma, J.: V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In: European conference on computer vision. pp. 107–124. Springer (2022)
20. Xu, R., Xiang, H., Xia, X., Han, X., Li, J., Ma, J.: 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 (2022)
21. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10) (2018). https://doi.org/10.3390/s18103337, https://www.mdpi.com/1424-8220/18/10/3337
22. Yu, H., Luo, Y., Shu, M., Huo, Y., Yang, Z., Shi, Y., Guo, Z., Li, H., Hu, X., Yuan, J., et al.: Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21361–21370 (2022)
23. Zhao, S.Z., Xiang, H., Xu, C., Xia, X., Zhou, B., Ma, J.: Coopre: Cooperative pretraining for v2x cooperative perception. arXiv preprint arXiv:2408.11241 (2024)
24. Zhou, S., Li, L., Zhang, X., Zhang, B., Bai, S., Sun, M., Zhao, Z., Lu, X., Chu, X.: LiDAR-PTQ: Post-training quantization for point cloud 3d object detection. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=0d1gQI114C
25. Zhou, Z., Xiang, H., Zheng, Z., Zhao, S.Z., Lei, M., Zhang, Y., Cai, T., Liu, X., Liu, J., Bajji, M., et al.: V2XPnP: Vehicle-to-everything spatio-temporal fusion for multi-agent perception and prediction. arXiv preprint arXiv:2412.01812 (2024)
26. Zhou, Z., Zhao, S.Z., Cai, T., Huang, Z., Zhou, B., Ma, J.: TurboTrain: Towards efficient and balanced multi-task learning for multi-agent perception and prediction. arXiv preprint arXiv:2508.04682 (2025)