ALIGNING SPARSE MESSAGES FOR POSE-ROBUST COLLABORATIVE PERCEPTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Collaborative perception improves detection by sharing observations across vehicles and roadside units, but its benefits are limited by communication cost and pose noise. Under a limited communication budget, features useful for detection may still be difficult to match across agents. We propose SAComm, a collaborative perception framework that jointly designs sparse patch messages and receiver-side alignment and fusion. Each sender transmits a small number of compressed local feature patches, preserving local structure for both detection and matching. The receiver matches these patches against ego features and estimates a shared translation and yaw correction for each sender while accounting for matching uncertainty and geometric consistency. An adaptive gate controls how much correction is applied to the entire message, while pose reliability guides local fusion. Thus, the same sparse messages support both alignment and collaborative detection. Experiments on OPV2V, V2XSet, and DAIR-V2X-C show that SAComm maintains strong detection performance under pose noise at the lowest communication rates among the evaluated feature-level methods. On OPV2V, it achieves 86.98% BEV AP_{70} under strong pose noise at only 0.393 Mbps per sender.

1 INTRODUCTION

Single-agent perception is inherently limited by occlusion, restricted fields of view, and sensing range (Han et al., 2023; Xu et al., 2023). Collaborative perception allows connected autonomous vehicles (CAVs) and roadside units (RSUs) to share observations, extending the ego CAV's view and improving detection (Chen et al., 2019b; Wang et al., 2020; Yu et al., 2022; Xiang et al., 2024). However, practical collaborative perception faces both communication bandwidth constraints and pose noise. Sharing raw sensor inputs or dense features incurs high communication costs, while pose noise can misalign shared observations and degrade detection. Therefore, low-bandwidth collaborative perception requires more than reducing transmission: the retained information has to also be effectively aligned and used under pose noise.

To reduce communication cost, existing methods use feature compression and spatial selection. V2VNet (Wang et al., 2020) compresses feature representations, while Where2comm (Hu et al., 2022) and InfoCom (Wei et al., 2026) select content relevant to detection. CoSDH (Xu et al., 2025) and BlindMap (Zhu et al., 2026) further consider what the ego CAV already observes and prioritize information it lacks. Other methods combine compact communication with geometric matching and alignment. ERMVP (Zhang et al., 2024) uses representative features and sparse consensus for pose correction, while CPPC (Ding et al., 2025) organizes semantic and structural information into point clusters to support matching and pose correction. These methods link message representation with geometric alignment. In this paper, we further study how to organize and use a small number of low-dimensional features for both alignment and detection.

For sparse messages, features useful for detection may still be difficult to match across agents. Under a limited communication budget, selecting more centers leaves less room for local neighborhoods. These neighborhoods preserve feature variations and spatial layout that can help distinguish candidate matches. Yet sparse matches may remain ambiguous or disagree on the relative pose, and applying an unreliable correction can worsen the alignment of the entire message. These challenges call for jointly designing the message and receiver: the message should retain local information for

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.

detection and matching, while the receiver should use matching uncertainty and geometric consistency to guide pose correction and fusion.

To address these challenges, we propose SAComm, which jointly designs sparse patch messages and receiver-side alignment and fusion. Sparse Patch Communication selects centers based on detection confidence and transmits compressed local feature patches around them, retaining local cues for both detection and matching. Observation-Based Pose Correction performs probabilistic matching between the received patches and ego features. It estimates a single relative pose correction for each sender, accounting for matching uncertainty and geometric consistency across patches. An adaptive gate controls how much correction is applied to the entire message, including patches without direct matches in the ego features. Reliability-Guided Local Fusion reuses the matching statistics and applied correction to estimate pose reliability and weight each sender's contribution. Local attention selects nearby candidate features to reduce the effect of residual misalignment. Thus, the same sparse message supports both alignment and detection.

We evaluate SAComm on OPV2V, V2XSet, and DAIR-V2X-C. Across all three datasets, it achieves the lowest communication rates among the evaluated feature-level methods while maintaining strong detection performance under pose noise. On OPV2V, it achieves 86.98% BEV AP_{70} at only 0.393 Mbps per sender, with noise standard deviations of 0.6 m per horizontal axis and 0.6° in yaw. Ablations at similar communication rates support the joint design of local message structure and receiver-side processing. Pose diagnostics show that the same sparse messages support effective pose correction. Our main contributions are summarized as follows:

- We introduce SAComm, which jointly designs compressed local patch messages and receiver-side alignment and fusion for robust detection with low communication overhead.
- We use matching uncertainty and geometric consistency to estimate a shared pose correction for each sender and adaptively control its magnitude. Pose reliability and local attention then guide fusion of the corrected messages to handle residual misalignment.
- We evaluate SAComm on three datasets, demonstrating robust detection under pose noise with low communication overhead.

2 Related Work

Communication efficiency in collaborative perception. Intermediate feature collaboration offers a trade-off between perception accuracy and communication cost (Chen et al., 2019a), but dense features remain expensive to transmit. Existing methods reduce this cost through compact encoding and selective communication. CodeFilling (Hu et al., 2024) uses codebook indices, while InfoCom (Wei et al., 2026) and RDComm (Liu et al., 2026) combine compact representations with message selection. Selective communication uses detection confidence, spatial and channel importance, or region selection at multiple resolutions (Hu et al., 2022; Yang et al., 2023; Wang et al., 2023). CoSDH (Xu et al., 2025) and BlindMap (Zhu et al., 2026) further consider the ego CAV's needs or blind areas when selecting content. Selection can also operate across agents or time: Who2com (Liu et al., 2020) learns communication handshakes for partner selection, while CooperTrim (Mukhopadhyay et al., 2026) uses temporal uncertainty to guide sharing. SAComm combines channel compression with sparse local patches, retaining context for both detection and pose correction.

Pose robustness in collaborative perception. Pose noise perturbs the coordinate transformations used to align shared features, and existing methods address the resulting misalignment through robust fusion or explicit pose correction. V2X-ViT (Xu et al., 2022a) and MRCNet (Hong et al., 2024) use attention and multi-scale aggregation to reduce sensitivity to spatial offsets. Learned pose correction (Vadivelu et al., 2021) estimates relative errors and enforces consistency across vehicles. CoAlign (Lu et al., 2023) optimizes a graph of vehicles and shared detections with uncertainty estimates, while RoCo (Huang et al., 2024) alternates object matching and pose adjustment. Other methods combine compact communication with robust alignment. ERMVP (Zhang et al., 2024) uses sparse consensus features and geometric verification. mmCooper (Liu et al., 2025) combines selective feature and detection sharing with offset-aware fusion and box calibration, while CPPC (Ding et al., 2025) preserves object structure in point clusters for matching and pose correction. SAComm uses compressed local patches and probabilistic matching, allowing the same sparse message to support pose correction and reliability-guided local fusion.

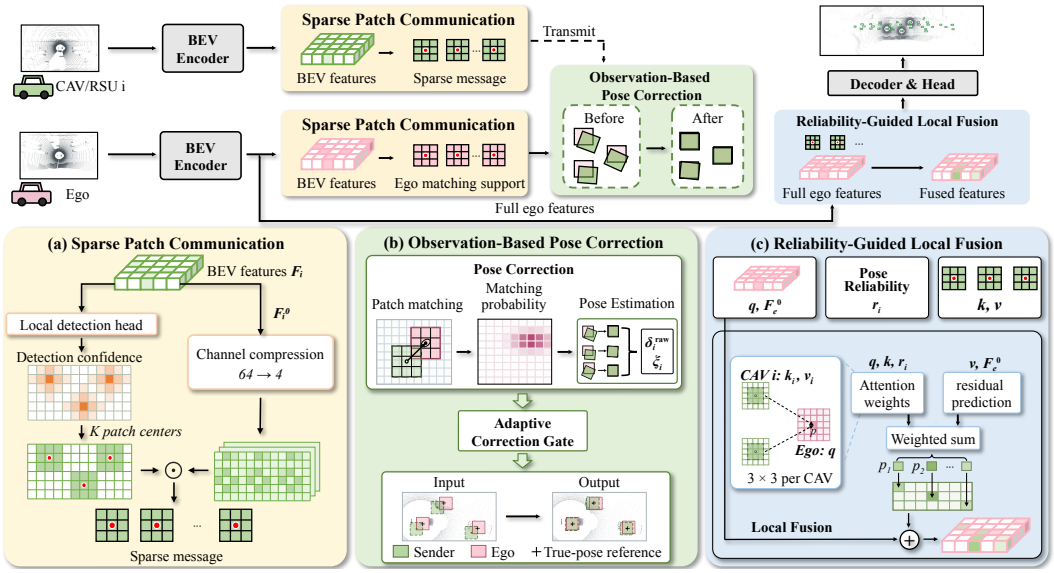


Figure 1: Overview of SAComm. Top: the overall pipeline. Bottom: (a) Sparse Patch Communication, (b) Observation-Based Pose Correction, and (c) Reliability-Guided Local Fusion.

3 METHOD

3.1 PROBLEM FORMULATION

We consider an ego CAV e and a set $\mathcal{N}_e$ of nearby CAVs and RSUs. Each participant extracts features F_i from its LiDAR point cloud X_i. Senders construct and transmit messages m_i from these features. The ego CAV uses the estimated relative rigid transformation $\widetilde{T}_{e\leftarrow i}$ to align the received features in the ego frame. The ego CAV combines the aligned features with its own features F_e to predict 3D bounding boxes $\widehat{Y}_e$. Pose noise independently perturbs the horizontal translation and yaw of each participant. The resulting relative transformation is

$$\boldsymbol{\epsilon}_e, \boldsymbol{\epsilon}_i \sim \mathcal{P}_s, \qquad \widetilde{T}_{e\leftarrow i} = G(\mathbf{p}_e + \boldsymbol{\epsilon}_e)^{-1} G(\mathbf{p}_i + \boldsymbol{\epsilon}_i). \tag{1}$$

Here, $\mathbf{p}_e$ and $\mathbf{p}_i$ denote the true poses of the ego CAV and sender. $G(\mathbf{p})$ transforms coordinates from the local frame to a shared frame, and $\mathcal{P}_s$ specifies the pose noise distribution. At a transmission frequency of f frames per second, the average communication rate per sender is $R = f\,\mathbb{E}[\ell(m_i)]/10^6$ Mbps. The message length $\ell(m_i)$ counts transmitted content and spatial indices in bits. The expectation averages over frames and senders. Our goal is to maintain detection accuracy under pose noise with low communication overhead.

3.2 SPARSE PATCH COMMUNICATION

Sparse Patch Communication (SPC). SPC reduces transmission through channel compression and sparse spatial selection (Figure 1(a)). The same sparse message provides features for detection and pose correction. Sender i uses a PointPillars (Lang et al., 2019) encoder to extract BEV features at three scales, $F_i = \{F_i^0, F_i^1, F_i^2\}$. We construct messages from $F_i^0 \in \mathbb{R}^{C\times H\times W}$, the feature map with the highest spatial resolution, to retain fine spatial detail. A channel compression module $\mathcal{E}_c$ compresses $C = 64$ channels to $d = 4$, producing $Z_i = \mathcal{E}_c(F_i^0) \in \mathbb{R}^{d\times H\times W}$. A local detection head predicts classification logits L_i from F_i. The confidence at u and the number of centers are

$$s_i(u) = \max_a \sigma(L_{i,a}(u)), \qquad K = \mathrm{round}(\rho HW), \tag{2}$$

where a indexes anchors, σ is sigmoid, and ρ is the patch-center selection ratio. We select a set $\mathcal{C}_i$ of K centers based on these scores, prioritizing local peaks. Around each center $c \in \mathcal{C}_i$, we retain its 3×3 neighborhood $\mathcal{N}_{3\times3}(c)$ within the feature map. These neighborhoods preserve the center response and nearby feature variation for detection and matching. The transmitted locations and message are

$$\Omega_i = \bigcup_{c\in\mathcal{C}_i} \mathcal{N}_{3\times3}(c), \qquad m_i = (\mathcal{C}_i, \{Z_i(u)\}_{u\in\Omega_i}). \tag{3}$$

The fixed patch shape allows the receiver to recover the transmitted locations from the center indices. Features at overlapping locations are sent only once. With b_{val} bits per feature value and b_{idx} bits per center index, the message size is $\ell(m_i) = d\, b_{\mathrm{val}} |\Omega_i| + K b_{\mathrm{idx}}$. Each sender constructs its message independently, without a prior request. The same message can serve multiple receivers.

3.3 Observation-Based Pose Correction

Observation-based pose correction uses correspondences between received patches and ego features to correct the noisy relative pose $\widetilde{T}_{e\leftarrow i}$. Matching uncertainty and geometric consistency guide the correction, which is adaptively scaled before aligning the entire message (Figure 1(b)).

Pose Correction (PC). PC combines patch matching with uncertainty-aware pose estimation to obtain a single pose correction for each sender. For patch k from sender i, the current relative transformation $\widetilde{T}_{e\leftarrow i}$ maps its center to $\widetilde{x}_k$ in the ego frame. The ego matching support consists of high-confidence locations and their neighborhoods. We search within this support near each transformed sender center. Let $\mathcal{U}_k$ contain the candidate offsets from $\widetilde{x}_k$ in meters. A shared convolutional network extracts normalized descriptors from the received sparse features and the full compressed ego features. Each descriptor includes local context. Let $d_{i,k}$ denote the sender descriptor at center k and $d_e(x)$ the ego descriptor at location x. Several locations may have similar descriptors, so patch matching retains a distribution over candidate offsets. The matching probability is

$$p_k(u) = \mathrm{softmax}_{u\in\mathcal{U}_k}\left[\frac{d_{i,k}^{\top} d_e(\widetilde{x}_k + u) - \alpha\|u\|_2/R_s}{\tau}\right]. \tag{4}$$

Here, R_s is the matching search range, τ is the temperature, and α controls the distance penalty that favors nearby candidates. The distribution gives the mean offset and covariance

$$\mu_k = \sum_{u\in\mathcal{U}_k} p_k(u)\, u, \qquad \Sigma_k = \sum_{u\in\mathcal{U}_k} p_k(u)(u-\mu_k)(u-\mu_k)^{\top} + \sigma_0^2 I_2. \tag{5}$$

The mean μ_k estimates the patch offset, while Σ_k describes its uncertainty in different directions. A small $\sigma_0 > 0$ ensures numerical stability. Uncertainty-aware pose estimation combines these offsets and covariances across patches. Since all patches from the same sender share one relative pose, we fit a single correction to their offsets. Let $\eta \in \mathbb{R}^3$ denote the normalized translation and yaw correction. Translation shifts all centers equally, whereas yaw produces shifts that depend on their positions. Near the current pose, these shifts are approximated by $J_k\eta$, where J_k is the position Jacobian. The correction is estimated by

$$\widehat{\eta} = \arg\min_{\eta}\left[\sum_k (J_k\eta - \mu_k)^{\top}\Lambda_k(J_k\eta - \mu_k) + \lambda\|\eta\|_2^2\right], \qquad \Lambda_k = w_k\Sigma_k^{-1}. \tag{6}$$

Here, w_k is a fitting weight derived from entropy-based matching confidence, and $\lambda > 0$ regularizes the correction. The inverse covariance reduces the influence of uncertain directions (Barfoot, 2017). A confident match can still disagree with other matches. After the initial fit, we reduce the weights of matches with large residuals and solve once more (Holland & Welsch, 1977). Bounding the updated estimate gives the normalized raw pose correction δ_i^{raw}.

Adaptive Correction Gate (ACG). The matching offsets can arise from feature differences as well as pose noise. Applying the full correction may therefore cause overcorrection when pose noise is small. ACG learns how much of the raw pose correction δ_i^{raw} to apply. A small MLP predicts a correction factor $g_i \in [0,1]$ from the raw correction magnitude and matching statistics ξ_i. These statistics summarize matching confidence, fitting residuals, and pose estimation uncertainty. The applied pose correction is

$$\delta_i = g_i\, \delta_i^{\mathrm{raw}}. \tag{7}$$

During training, the gate is supervised by the factor that brings the scaled raw correction closest to the ground truth correction (Equation 13). The translation and yaw obtained from δ_i update the relative transformation to $\widehat{T}_{e\leftarrow i}$. This corrected transformation aligns the entire received message, including patches without reliable correspondences in the ego features.

3.4 RELIABILITY-GUIDED LOCAL FUSION

After pose correction, received features are fused with the full ego features. Pose reliability controls sender contributions, while local attention selects nearby features at each ego location (Figure 1(c)).

Pose Reliability (PR). PR estimates the residual pose error after correction to weight each sender during fusion. An MLP predicts normalized residual translation and yaw errors from the applied correction δ_i and matching statistics ξ_i. These predictions give a pose reliability score r_i.

$$[\widehat{e}_{t,i}, \widehat{e}_{\theta,i}]^\top = \mathrm{softplus}(\mathrm{MLP}_{\mathrm{rel}}([\,|\delta_i| \oslash \delta_{\max};\, \xi_i\,])), \quad r_i = \exp(-a_t \widehat{e}_{t,i} - a_\theta \widehat{e}_{\theta,i}). \tag{8}$$

Here, $[\,;\,]$ and $\oslash$ denote concatenation and elementwise division. $\delta_{\max}$ gives the normalized correction bounds, and $a_t, a_\theta > 0$ are learned scales. Larger predicted errors give lower reliability.

Local Fusion (LF). Local candidates allow fusion to accommodate small spatial offsets. We align the received sparse features and their validity masks using $\widehat{T}_{e\leftarrow i}$. Linear projections produce the ego query q_e from F_e^0 and sender keys k_i from the aligned compressed features, both with dimension d_k. A convolutional decoder produces sender values v_i. At each ego location p with valid sender coverage, we collect valid candidates from a 3×3 neighborhood of each sender. The set $\mathcal{A}(p)$ contains pairs (i, u), where i is the sender and u is an offset on the receiver's feature grid. Attention weights combine scaled dot products (Vaswani et al., 2017) with pose reliability. A zero-update candidate lets the network decide whether to update the ego feature. Attention is normalized jointly over this candidate and all valid sender candidates. For $(i, u) \in \mathcal{A}(p)$, the scores and weights are

$$\begin{aligned} s_{i,u}(p) &= \frac{q_e(p)^\top k_i(p+u)}{\sqrt{d_k}} + b_u + \log\max(r_i, \epsilon), \\ s_{\mathrm{keep}}(p) &= w_{\mathrm{keep}}^\top q_e(p) + b_{\mathrm{keep}}, \\ [a_{\mathrm{keep}}(p), \{a_{i,u}(p)\}] &= \mathrm{softmax}([s_{\mathrm{keep}}(p), \{s_{i,u}(p)\}]). \end{aligned} \tag{9}$$

The offset bias b_u and parameters $w_{\mathrm{keep}}, b_{\mathrm{keep}}$ are learned, and $\epsilon > 0$ keeps the logarithm finite. Each sender shares one reliability score r_i, while attention weights vary across ego locations and local candidates.

In parallel, residual prediction uses an MLP to produce an update $h_{i,u}(p)$ from the ego feature $F_e^0(p)$ and candidate value $v_i(p+u)$. At each p, a weighted sum of these candidate updates gives the residual, scaled by a learned $\beta \in (0, 1)$. Residuals are placed at their corresponding locations to form a residual map, which is added to the full ego features:

$$\widehat{F}_e^0(p) = F_e^0(p) + \beta \sum_{(i,u)\in\mathcal{A}(p)} a_{i,u}(p)\, h_{i,u}(p). \tag{10}$$

The zero-update candidate contributes no residual, and locations without valid sender coverage remain unchanged. Only F_e^0 is updated. The multiscale decoder and detection head take $\widehat{F}_e^0, F_e^1, F_e^2$ to predict $\widehat{Y}_e$.

3.5 TRAINING STRATEGY AND OBJECTIVES

Stage I: Clean pretraining. We first train sparse communication, local fusion, and detection with accurate relative poses. The model learns to retain detection information in sparse messages and combine it with the full ego features. PC and ACG are disabled, and sender reliability is fixed at $r_i = 1$. The objective is

$$\mathcal{L}_{\mathrm{clean}} = \mathcal{L}_{\mathrm{det}} + \mathcal{L}_{\mathrm{single}}. \tag{11}$$

Here, $\mathcal{L}_{\mathrm{det}}$ supervises fused detections, while $\mathcal{L}_{\mathrm{single}}$ trains the local detection head that provides confidence scores for patch selection.

Stage II: Robust fine-tuning. We initialize from Stage I, enable PC, ACG, and PR, and fine-tune the full model on clean and noisy samples. This stage learns to correct pose errors and use sender features according to pose reliability. The objective is

$$\mathcal{L}_{\mathrm{robust}} = \mathcal{L}_{\mathrm{det}} + \mathcal{L}_{\mathrm{single}} + \lambda_{\mathrm{pose}}\mathcal{L}_{\mathrm{pose}} + \mathcal{L}_{\mathrm{match}} + \lambda_{\mathrm{gate}}\mathcal{L}_{\mathrm{gate}} + \lambda_{\mathrm{rel}}\mathcal{L}_{\mathrm{rel}}. \tag{12}$$

The pose loss $\mathcal{L}_{\text{pose}}$ supervises the applied correction δ_i using the normalized target δ_i^{gt} derived from ground truth poses. The matching loss $\mathcal{L}_{\text{match}}$ trains patch matching through descriptor consistency and geometric agreement. The gate loss $\mathcal{L}_{\text{gate}}$ supervises the ACG output g_i with the factor that brings the scaled raw correction closest to the target:

$$g_i^\star = \arg\min_{0 \le g \le 1} \left\| g\, \delta_i^{\text{raw}} - \delta_i^{\text{gt}} \right\|_2^2 . \tag{13}$$

The reliability loss $\mathcal{L}_{\text{rel}}$ trains PR to predict the residual translation and yaw errors after applying δ_i. Detection gradients also update PR and LF, but do not pass through the corrected pose or the inputs to PR. Additional method details and loss definitions are provided in Appendix A.

4 Experiments

4.1 Experimental Setup

We evaluate LiDAR vehicle detection on the OPV2V (Xu et al., 2022b), V2XSet (Xu et al., 2022a), and DAIR-V2X-C (Yu et al., 2022) datasets. On OPV2V, we use up to five CAVs, including the ego CAV. V2XSet uses the same limit, counting both CAVs and RSUs. DAIR-V2X-C uses one CAV and one RSU. We compare SAComm with NoFusion, Early Fusion, Late Fusion, AttFuse (Xu et al., 2022b), Where2comm (Hu et al., 2022), ERMVP (Zhang et al., 2024), CoSDH (Xu et al., 2025), InfoCom (Wei et al., 2026), and BlindMap (Zhu et al., 2026). All methods are evaluated on the same data splits and noise settings. We follow the Gaussian pose noise setting of CoAlign (Lu et al., 2023), with standard deviations of s meters for each horizontal position component and s degrees for yaw. For SAComm, Stage I uses accurate poses, while Stage II samples $s = 0$ or 0.2 with equal probability per scene. For each method, a fixed checkpoint is evaluated at $s \in \{0, 0.2, 0.4, 0.6\}$. Ground truth boxes remain in the clean ego coordinate frame. We report BEV AP_{50} and AP_{70} at IoU thresholds of 0.5 and 0.7 using global confidence ranking. The average communication rate per sender is computed at 10 Hz, as defined in Section 3.1. Baseline configurations and training settings are detailed in Appendix B.

4.2 Quantitative Results

Main results. Table 1 reports detection accuracy under different pose noise levels together with communication rates. SAComm achieves the lowest communication rate among the evaluated feature-level methods on all three datasets. On OPV2V, it uses 92.1% less communication than InfoCom while improving AP_{70} by 16.26 percentage points at $s = 0.6$. Although Early Fusion performs best at zero pose noise on OPV2V and V2XSet, SAComm achieves the highest AP_{50} and AP_{70} at every tested nonzero noise level on both datasets, including noise levels higher than those used for robust fine-tuning. At $s = 0.6$, it exceeds ERMVP, the strongest competing method in AP_{70} on these datasets, by 9.23 and 7.99 percentage points on OPV2V and V2XSet, respectively. In contrast, Early Fusion and Late Fusion fall below NoFusion at $s = 0.6$ on both datasets, whereas SAComm continues to benefit from collaboration. On DAIR-V2X-C, SAComm achieves the highest AP_{50} across all tested noise levels and the highest AP_{70} at zero and low pose noise. At $s = 0.4$ and 0.6, its AP_{70} remains close to InfoCom's while using 84.7% less communication. Overall, these results show that sparse communication can preserve the benefits of collaboration under pose noise.

Component ablations. Table 2 examines whether sparse messages can support both pose correction and reliability-guided fusion. We keep the SPC settings fixed, compare mean fusion with LF, and progressively add PC, PR, and ACG. Mean fusion averages the ego and valid aligned sender features at each location. At $s = 0.6$, LF improves AP_{70} over mean fusion by 6.18 percentage points, while adding PC brings a further 6.04-point gain. This suggests that LF alone does not fully address misalignment under strong pose noise. Adding PR further improves accuracy at all tested noise levels, showing the benefit of weighting corrected messages by pose reliability. PC reduces accuracy at zero pose noise. Compared with LF + PC + PR, adding ACG improves AP_{70} by 1.43 percentage points at $s = 0$, but reduces it by 1.14 points at $s = 0.6$. Despite this trade-off, the full model outperforms LF alone at every tested noise level with similar communication rates. Together, these results support using the same sparse messages for pose correction and collaborative detection, with pose reliability guiding fusion without additional alignment messages.

Table 1: BEV detection under pose noise. Rates are measured at $s = 0$. Best AP values are bold.

Dataset	Method	Rate (Mbps) ↓	AP_{50} (%) ↑				AP_{70} (%) ↑			
			$s = 0$	$s = 0.2$	$s = 0.4$	$s = 0.6$	$s = 0$	$s = 0.2$	$s = 0.4$	$s = 0.6$
OPV2V	NoFusion	0	81.53	81.53	81.53	81.53	72.93	72.93	72.93	72.93
	Early Fusion	71.598	**96.70**	92.55	70.01	51.96	**93.31**	66.30	35.95	25.12
	Late Fusion	0.216	96.13	85.08	48.27	35.14	92.01	43.91	21.88	19.19
	AttFuse	86.508	96.19	94.78	89.80	86.21	91.84	81.95	70.74	67.98
	Where2comm	38.976	94.67	93.59	88.92	85.67	89.99	79.82	67.13	63.29
	ERMVP	115.081	95.88	94.59	91.59	88.90	87.27	84.38	80.18	77.75
	CoSDH	19.310	93.59	92.49	86.45	79.90	87.75	80.10	67.17	60.62
	InfoCom	4.948	95.72	94.56	90.19	86.75	89.59	82.53	74.36	70.72
	BlindMap	14.366	95.47	93.61	86.05	80.23	90.73	78.73	64.80	60.13
	SAComm	0.393	96.47	**96.29**	**95.56**	**93.92**	92.41	**91.25**	**89.76**	**86.98**
V2XSet	NoFusion	0	74.96	74.96	74.96	74.96	62.47	62.47	62.47	62.47
	Early Fusion	47.437	**95.04**	89.49	64.31	45.67	**90.26**	59.33	29.73	20.06
	Late Fusion	0.194	88.63	77.54	50.47	39.00	79.32	41.99	26.02	23.07
	AttFuse	86.508	90.40	88.09	80.98	75.69	82.48	69.69	57.56	53.79
	Where2comm	1.907	89.88	88.07	83.32	80.13	78.89	71.24	63.41	60.38
	ERMVP	115.081	89.82	88.36	85.00	81.97	77.46	73.91	69.62	67.27
	CoSDH	53.462	91.61	88.47	76.11	66.04	83.11	64.81	45.15	38.28
	InfoCom	4.948	91.85	90.34	85.99	82.33	81.50	75.31	68.25	65.09
	BlindMap	14.497	91.30	89.60	84.20	79.80	80.93	73.02	63.93	60.04
	SAComm	0.391	91.30	**90.83**	**89.66**	**87.99**	82.01	**79.98**	**77.70**	**75.26**
DAIR-V2X-C	NoFusion	0	65.33	65.33	65.33	65.33	55.47	55.47	55.47	55.47
	Early Fusion	77.521	74.79	72.37	67.63	64.42	59.55	55.22	51.89	50.51
	Late Fusion	0.188	63.44	58.60	49.26	44.52	46.26	37.93	33.55	32.44
	AttFuse	64.512	78.11	75.75	71.04	68.36	64.34	59.69	56.38	55.15
	Where2comm	18.324	74.63	73.07	70.75	69.31	60.86	58.89	57.61	57.04
	ERMVP	85.801	74.73	73.15	70.95	69.40	58.88	58.00	56.93	56.11
	CoSDH	34.830	73.78	71.74	68.26	66.07	58.14	55.43	53.35	52.46
	InfoCom	3.711	77.45	75.35	71.47	69.28	64.18	60.76	**58.72**	57.74
	BlindMap	10.404	76.25	74.32	70.73	68.74	60.96	57.40	54.99	54.28
	SAComm	0.566	**78.74**	**76.59**	**72.34**	**69.95**	**65.17**	**61.15**	58.64	**57.79**

Table 2: Receiver ablations on OPV2V with fixed SPC settings: $d = 4$, $\rho = 0.1\%$, and 3×3 patches. Mean: averaging over valid regions. Last row: full SAComm.

Fusion	PC	PR	ACG	Rate (Mbps)	AP_{70} (%)			
					$s = 0$	$s = 0.2$	$s = 0.4$	$s = 0.6$
Mean	×	×	×	0.393	92.31	86.42	78.78	75.12
LF	×	×	×	0.394	91.96	88.21	83.82	81.30
LF	✓	×	×	0.395	90.29	90.31	89.45	87.34
LF	✓	✓	×	0.394	90.98	90.72	**90.09**	**88.12**
LF	✓	✓	✓	0.393	**92.41**	**91.25**	89.76	86.98

Table 3: OPV2V ablations after robust fine-tuning. Sizes: side lengths in cells. K: centers per sender. Rate groups at $s = 0$ reference the default (both sizes 3, $K = 34$).

Rate group	SPC patch	LF neigh.	K	Rate (Mbps)	AP_{70} (%)			
					$s = 0$	$s = 0.2$	$s = 0.4$	$s = 0.6$
Similar	1	1	246	0.394	87.36	86.53	85.48	84.37
	3	1	34	0.395	92.57	91.17	88.38	83.50
	3	3	34	0.393	92.41	91.25	**89.76**	**86.98**
	5	5	12	0.377	89.29	88.78	87.66	84.99
Lower	1	1	34	0.054	85.59	85.23	84.73	83.76
	3	3	7	0.083	87.72	87.25	86.77	85.43
	3	3	17	0.199	91.66	90.97	89.63	86.68
Higher	3	3	68	0.776	91.78	89.68	87.05	84.17
	5	5	34	1.016	**93.32**	**91.82**	89.40	85.05

Table 4: Pose diagnostics of full SAComm on OPV2V, averaged over ego–sender pairs. Input: relative pose error before correction. PC: residual error after applying δ_i^{raw}. PC+ACG: residual error after applying δ_i (Eq. 7). Both use the same forward pass. Mean r_i: pose reliability (Eq. 8).

s	Translation error (m)			Yaw error (°)			Mean r_i
	Input	PC	PC+ACG	Input	PC	PC+ACG	
0	0.0000	0.1787	0.0247	0.0000	0.2111	0.0239	0.784
0.2	0.3703	0.1851	0.1786	0.2227	0.2200	0.1560	0.690
0.4	0.7426	0.2303	0.3023	0.4514	0.2621	0.2834	0.577
0.6	1.0933	0.3687	0.5451	0.6969	0.4009	0.4936	0.506

Message and fusion settings. Table 3 examines how to allocate and use a limited communication budget for robust detection under pose noise. We vary the transmitted patch size and number of centers in SPC, together with the local attention neighborhood in LF. At comparable communication rates, the default setting with 3×3 patches and a 3×3 fusion neighborhood outperforms both isolated features and configurations with fewer 5×5 patches, suggesting a balance between local context and spatial coverage. The local attention neighborhood in LF also matters. With the transmitted patch size and number of centers fixed, reducing the local attention neighborhood from 3×3 to 1×1 lowers AP_{70} by about 3.5 percentage points at $s = 0.6$, while having little effect at zero pose noise. This suggests that using nearby features during LF is particularly useful under strong pose noise, even when local patches are already transmitted. With both patch size and fusion neighborhood fixed at 3×3, reducing the number of centers from 34 to 17 saves approximately 49% of communication with only a 0.30-point AP_{70} loss at $s = 0.6$. This provides a lower-rate option, although the accuracy loss is larger at zero pose noise. Increasing the number of centers to 68 or using 5×5 patches and LF neighborhoods with 34 centers raises communication without improving accuracy under strong pose noise. The 5×5 configuration improves accuracy at zero and low pose noise. Overall, these results support preserving local context in sparse messages and using it during fusion, rather than simply transmitting more features. The default configuration provides a favorable balance between robustness and communication cost.

Pose correction and reliability. Table 4 reports relative pose errors before and after correction, together with the estimated pose reliability. PC reduces both translation and yaw errors at all tested nonzero noise levels. At $s = 0.6$, the mean translation error decreases from 1.0933 m to 0.3687 m, while the mean yaw error decreases from 0.6969° to 0.4009°. These reductions show that the sparse messages retain geometric information that the receiver can use to correct both translation and yaw. The improvements also occur at $s = 0.4$ and 0.6, above the noise levels used during robust fine-tuning. Thus, the benefit of PC is not limited to the training noise settings: correspondences from sparse messages remain useful for correcting larger pose errors. Comparing the raw and gated corrections within the same forward pass reveals the role of ACG. At zero pose noise, PC introduces nonzero errors despite accurate input poses. ACG reduces the resulting translation error from 0.1787 m to 0.0247 m and the yaw error from 0.2111° to 0.0239°. At $s = 0.6$, however, the gated correction leaves larger residual errors than the raw correction, with 0.5451 m translation error and 0.4936° yaw error. ACG therefore suppresses unnecessary corrections at low noise but can weaken useful corrections at high noise. Mean pose reliability decreases from 0.784 to 0.506 as pose noise increases, alongside larger residual errors after correction. The component ablations in Table 2 provide complementary detection evidence: ACG improves accuracy at low noise but reduces it at higher noise levels, whereas PR improves accuracy at all tested noise levels. Together, these results show that sparse matching evidence supports pose correction, while the estimated pose reliability guides how corrected messages contribute to fusion.

Number of participating CAVs. Under pose noise, additional CAVs provide more observations, but their misaligned messages may also reduce detection accuracy. We therefore examine whether adding more collaborators still improves detection. We evaluate two settings on OPV2V, with all counts including the ego CAV. On the full test set, we vary the maximum CAV count from two to five, although some frames contain fewer available CAVs. On the subset of all 101 OPV2V test frames with five available CAVs, we vary the actual participating count from two to five while keeping the frames and ground truth fixed. Each method uses fixed model weights, and CAVs shared across count settings receive the same sampled pose noise. At zero pose noise, moving from the two-CAV setting to the five-CAV setting improves AP_{70} for all evaluated methods in both evaluations (Figure 2(a, b)). At $s = 0.6$, the same change reduces AP_{70} for all eight comparison methods,

whereas SAComm gains 0.57 percentage points on the full test set and 3.72 points on the five-CAV subset (Figure 2(c, d)). These results show that SAComm can benefit from additional collaborators even under strong pose noise. Full results and evaluation details are provided in Appendix C.

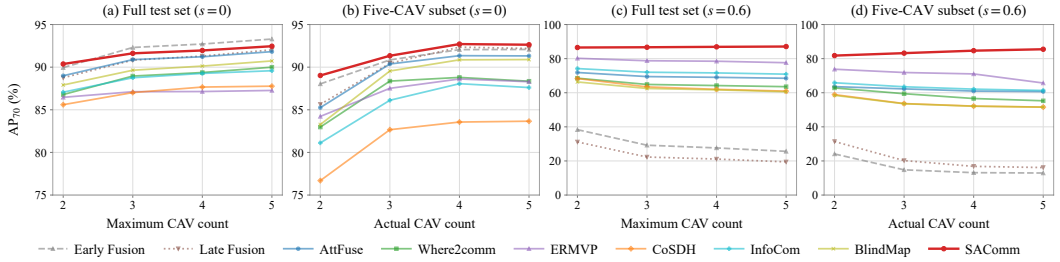


Figure 2: AP_{70} versus CAV count on OPV2V at $s = 0$ and $s = 0.6$. (a, c): maximum count on the full test set (2,170 frames). (b, d): actual count on the five-CAV subset (101 frames).

4.3 Qualitative Results

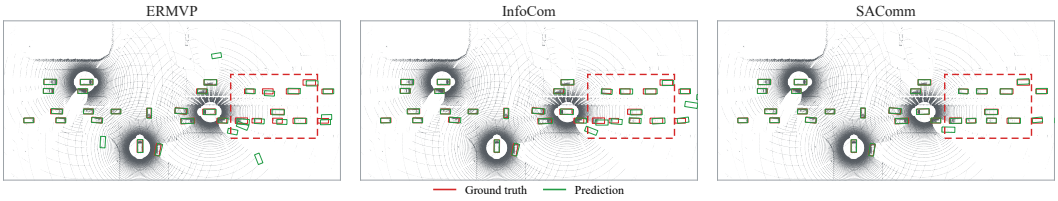


Figure 3: Detection comparison on OPV2V at $s = 0.6$ with identical pose noise. Red: ground-truth boxes. Green: predicted boxes (score ≥ 0.2). Point clouds use true poses for display only.

We provide qualitative comparisons of detection under pose noise and patch alignment before and after correction. Figure 3 compares ERMVP, InfoCom, and SAComm on the same OPV2V scene at $s = 0.6$. In the marked region, SAComm's predictions show better agreement with the ground-truth boxes, whereas ERMVP and InfoCom show shifted or extra boxes. Figure 4 enlarges the red dashed region in Figure 3 and shows one sender's patches before correction and after applying the gated pose correction. After correction, the patch centers move closer to their reference positions under the true relative pose: the translation error decreases from 1.569 m to 0.281 m, and the yaw error decreases from $1.013°$ to $0.286°$. The same correction also brings the other transmitted patches, shown in orange, closer to their reference positions. This illustrates the alignment improvement reported in the Input and PC+ACG columns of Table 4. Matching evidence from the same sparse message used for detection helps correct the relative pose and align the entire message. Additional examples are provided in Appendix D.

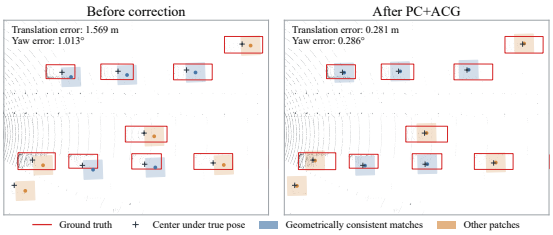


Figure 4: One sender's patch alignment. Blue: patches with geometrically consistent matches. Orange: other transmitted patches. Black crosses: centers under the true pose.

5 Conclusion

We presented SAComm, which preserves local structure in sparse patch messages for alignment and detection. Matching uncertainty and geometric consistency guide pose correction, while an adaptive gate controls its magnitude and pose reliability guides local fusion. Experiments on three datasets demonstrate robust detection with low communication overhead. OPV2V analyses further show pose error reductions and detection gains from additional CAVs under pose noise. These results suggest that communication-efficient collaborative perception depends not only on message size, but also on how shared information is organized, aligned, and used.

AI USE STATEMENT

We used generative AI tools to assist with literature discovery, translation, manuscript drafting and revision, experimental planning and result interpretation, and the preparation of analysis scripts and figures. The authors reviewed the AI-assisted content and take responsibility for the final manuscript, code, and reported results.

REPRODUCIBILITY STATEMENT

Appendix A provides the network architectures, algorithmic details, and training objectives. Appendix B specifies the dataset splits, preprocessing, training schedules, checkpoint selection, evaluation settings, and communication accounting. Additional quantitative results and diagnostic analyses are provided in Appendix C.

REFERENCES

Timothy D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017. doi: 10.1017/9781316671528.

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, 2019a.

Qi Chen, Sihai Tang, Qing Yang, and Song Fu. Cooper: Cooperative perception for connected autonomous vehicles based on 3D point clouds. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 514–524, 2019b.

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

Yushan Han, Hui Zhang, Huifang Li, Yi Jin, Congyan Lang, and Yidong Li. Collaborative perception in autonomous driving: Methods, datasets, and challenges. *IEEE Intelligent Transportation Systems Magazine*, 15(6):131–151, 2023.

Paul W. Holland and Roy E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, 6(9):813–827, 1977. doi: 10.1080/03610927708827533.

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 *Advances in Neural Information Processing Systems*, volume 35, 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15481–15490, June 2024.

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 32nd ACM International Conference on Multimedia*, pp. 7833–7842, 2024.

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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12697–12705, June 2019.

Bingyi Liu, Jian Teng, Hongfei Xue, Enshu Wang, Chuanhui Zhu, Pu Wang, and Libing Wu. mmCooper: A multi-agent multi-stage communication-efficient and collaboration-robust cooperative perception framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28396–28406, 2025.

Genjia Liu, Anning Hu, Yue Hu, Wenjun Zhang, and Siheng Chen. Rate-distortion optimized pragmatic communication for collaborative perception. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 119998–120019, 2026.

Yen-Cheng Liu, Junjiao Tian, Chih-Yao Ma, Nathan Glaser, Chia-Wen Kuo, and Zsolt Kira. Who2com: Collaborative perception via learnable handshake communication. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6876–6883, 2020.

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.

Shilpa Mukhopadhyay, Amit Roy-Chowdhury, and Hang Qiu. CooperTrim: Adaptive data selection for uncertainty-aware cooperative perception. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Nicholas Vadivelu, Mengye Ren, James Tu, Jingkang Wang, and Raquel Urtasun. Learning to communicate and correct pose errors. In *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 1195–1210. PMLR, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Tianhang Wang, Guang Chen, Kai Chen, Zhengfa Liu, Bo Zhang, Alois Knoll, and Changjun Jiang. UMC: A unified bandwidth-efficient and multi-resolution based collaborative perception framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8187–8196, 2023.

Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. V2VNet: Vehicle-to-vehicle communication for joint perception and prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 605–621. Springer, 2020.

Quanmin Wei, Penglin Dai, Wei Li, Bingyi Liu, and Xiao Wu. InfoCom: Kilobyte-scale communication-efficient collaborative perception with information bottleneck. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(35):29731–29739, 2026.

Hao Xiang, Zhaoliang Zheng, Xin Xia, Runsheng Xu, Letian Gao, Zewei Zhou, Xu Han, Xinkai Ji, Mingxi Li, Zonglin Meng, Li Jin, Mingyue Lei, Zhaoyang Ma, Zihang He, Haoxuan Ma, Yunshuang Yuan, Yingqian Zhao, and Jiaqi Ma. V2X-Real: A largs-scale dataset for vehicle-to-everything cooperative perception. In *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, June 2025.

Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2X-ViT: Vehicle-to-everything cooperative perception with vision transformer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 107–124, 2022a.

Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. OPV2V: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2583–2589, 2022b.

Runsheng Xu, Xin Xia, Jinlong Li, Hanzhao Li, Shuo Zhang, Zhengzhong Tu, Zonglin Meng, Hao Xiang, Xiaoyu Dong, Rui Song, Hongkai Yu, Bolei Zhou, and Jiaqi Ma. V2V4Real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13712–13722, June 2023.

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

Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, and Zaiqing Nie. 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 (CVPR)*, pp. 21361–21370, June 2022.

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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12575–12584, June 2024.

Zhenhan Zhu, Yihang Jiang, and Yanchao Zhao. BlindMap: Explicit blind-area prediction and request-free communication for efficient cooperative perception. In *IEEE Conference on Computer Communications (INFOCOM)*, 2026.

A METHOD DETAILS

This appendix expands the method in Sections 3.2–3.5, with an inference algorithm, implementation details, and loss definitions. Dataset settings and training schedules are given in Appendix B.

A.1 OVERALL ALGORITHM

Algorithm 1 summarizes the pipeline in Section 3. Each sender transmits one sparse message; the ego features remain dense.

Algorithm 1: SAComm inference

Input: Local point clouds $\{X_j\}_{j\in\{e\}\cup\mathcal{N}_e}$ and estimated relative poses $\{\widetilde{T}_{e\leftarrow i}\}_{i\in\mathcal{N}_e}$.

Output: Detected boxes $\widehat{Y}_e$ in the ego frame.

1. Extract multiscale features F_j and local detection scores for each participant.
2. For each sender i, compress F_i^0, select patch centers $\mathcal{C}_i$, and transmit m_i from Equation 3.
3. At the receiver, recover each sparse feature map and its mask. Extract descriptors from the received maps and the dense compressed ego map.
4. Select the top-scoring ego locations and expand their neighborhoods to form the matching support.
5. For each sender, map its patch centers with $\widetilde{T}_{e\leftarrow i}$ and retain valid local matching candidates within the ego support. Compute p_k, μ_k, and Σ_k using Equations 4 and 5.
6. Fit one pose correction, reweight inconsistent matches, and fit again. Bound the estimate to obtain δ_i^{raw}.
7. Use ACG to obtain $\delta_i = g_i\delta_i^{\mathrm{raw}}$ and update the relative pose. Use PR to obtain r_i from the applied correction and matching statistics.
8. Align the entire sparse message and its mask with the updated pose.
9. At each covered ego location, score valid local sender candidates and the zero-update candidate. Add their weighted residual updates to F_e^0 using Equations 9 and 10.
10. Decode $\{\widehat{F}_e^0, F_e^1, F_e^2\}$ and predict $\widehat{Y}_e$.

During clean pretraining, PC and ACG are disabled, the supplied poses are used directly, and $r_i = 1$. Locations without sender coverage retain the ego features; with no senders, detection uses only ego features.

A.2 SPARSE PATCH COMMUNICATION

This subsection details the selection and message construction in Equations 2 and 3 of Section 3.2.

Feature representation. The BEV backbone takes a 64-channel feature map from PointPillars. It produces three scales with 64, 128, and 256 channels at strides 2, 4, and 8 relative to this input. The highest-resolution feature F_i^0 has spatial size $H \times W$. A shared multiscale decoder maps each scale to 128 channels at this resolution, concatenates them, and reduces 384 channels to 256. Separate local and fused 1×1 convolutional heads predict classification scores, seven box parameters, and two direction bins per anchor. The channel encoder compresses F_i^0 to $Z_i \in \mathbb{R}^{4\times H\times W}$. Table 5 lists its layers and the other small networks.

Center selection. Local peaks have confidence no smaller than any location in their 3×3 neighborhood. We select the K highest-scoring peaks, filling any shortage from the highest-scoring unselected locations. The count in Equation 2 is clipped to $[1, HW]$, so selection always returns the requested number of centers.

Patch construction and recovery. A stride-one 3×3 max pool expands the binary center mask into M_i, the indicator of Ω_i in Equation 3. Patches are clipped at grid boundaries, and overlapping locations are included only once. The center indices determine the payload mask M_i. We represent the received features as $\overline{Z}_i = M_i \odot Z_i$, with untransmitted locations set to zero. Payload accounting is given in Appendix B.2.

Ego matching support. The ego CAV independently selects its top 5% confidence locations without peak suppression and expands them by a 3×3 max pool. This untransmitted mask restricts matching locations, not feature values: descriptors use the full compressed ego map Z_e, and fusion retains the full 64-channel map F_e^0.

A.3 Observation-Based Pose Correction

This subsection expands PC and ACG in Section 3.3, specifying matching, the solver for Equation 6, and the pose update.

Descriptors and matching. The shared network in Table 5 extracts descriptors from received sparse maps and the full ego map. Around each transmitted center's predicted ego position, a 17×17 grid searches $[-1.6, 1.6]$ m along each axis at 0.2 m intervals. Ego descriptors and support are sampled bilinearly; descriptors are normalized again. Candidates must lie inside the ego grid and have interpolated support at least 0.1. Equations 4 and 5 give the masked matching distribution p_k, mean offset μ_k, and covariance Σ_k. Initial solver weights are based on the entropy confidence

$$c_k = 1 - \frac{H(p_k)}{\log M}, \tag{14}$$

where H is entropy and M counts the full search grid. Centers without valid candidates have zero confidence. The initial fitting weight in Equation 6 is $w_k = \max(c_k, 10^{-3})$.

Joint pose estimation. Translation and yaw are normalized by $\kappa_t = 0.2$ m and $\kappa_\theta = 0.2^\circ$. We compute J_k by finite differences of transformed patch-center positions, using unit changes in these normalized parameters. The objective in Equation 6 gives

$$A\widehat{\eta} = b, \qquad A = \sum_k J_k^\top \Lambda_k J_k + \lambda I_3, \qquad b = \sum_k J_k^\top \Lambda_k \mu_k. \tag{15}$$

After the first fit, each weight is multiplied by $\min(1, 2/\nu_k)$, where ν_k is the covariance-normalized residual magnitude. Solving again gives the final estimate $\widehat{\eta}$ and inverse information matrix $P_i = A^{-1}$. The bounded correction is $\delta_i^{\mathrm{raw}} = \delta_{\max} \odot \tanh(\widehat{\eta} \oslash \delta_{\max})$, with $\delta_{\max} = (8, 8, 6)^\top$.

Matching statistics. The nine entries of ξ_i comprise three uncertainty values $\log(1 + \operatorname{diag} P_i)/5$, the root mean squared residual norm divided by R_s, mean matching confidence, the inlier ratio at residual threshold 2, the center count encoded as $\log(1 + K_i)/\log 65$, and horizontal and vertical spans divided by $W - 1$ and $H - 1$, respectively. Here, $K_i = |\mathcal{C}_i|$ is the total number of transmitted patch centers from sender i, including centers without valid matching candidates. The spans are measured from sender-center pixel coordinates. All statistics use the final unbounded fit before ACG; the inlier test uses covariance-normalized residuals.

Gate and pose update. ACG combines the raw correction magnitude and matching statistics:

$$h_i^{\mathrm{raw}} = [\, |\delta_i^{\mathrm{raw}}| \oslash \delta_{\max} \,;\, \xi_i \,], \qquad g_i = \sigma(\mathrm{MLP}_g(h_i^{\mathrm{raw}})). \tag{16}$$

The MLP is specified in Table 5. Image warping uses the inverse sampling transform $B_i = \widetilde{T}_{e \leftarrow i}^{-1}$. Let $C(\delta_i)$ convert the applied correction $\delta_i = g_i \delta_i^{\mathrm{raw}}$ into a planar rigid transform using κ_t, κ_θ. The updated transforms are

$$\widehat{B}_i = C(\delta_i) B_i, \qquad \widehat{T}_{e \leftarrow i} = \widehat{B}_i^{-1} = \widetilde{T}_{e \leftarrow i} C(\delta_i)^{-1}. \tag{17}$$

This pose aligns the entire received message.

A.4 Reliability-Guided Local Fusion

This subsection completes the network and mask definitions behind Equations 8–10 in Section 3.4.

Pose reliability. PR uses the applied correction δ_i and matching statistics ξ_i. The positive scales in Equation 8 are $a_j = \mathrm{softplus}(\gamma_j) + 10^{-3}$ for $j \in \{t, \theta\}$. PR gives one reliability per sender, while LF assigns attention weights to individual spatial candidates.

Alignment and valid coverage. Bilinear warping $\mathcal{W}_{\mathrm{bil}}$ and nearest-neighbor warping $\mathcal{W}_{\mathrm{near}}$ use the sampling transform $\widehat{B}_i$ from Equation 17, with zero padding:

$$\begin{aligned} V_i &= \mathcal{W}_{\mathrm{near}}(M_i; \widehat{B}_i) \odot \mathcal{W}_{\mathrm{near}}(\mathbf{1}; \widehat{B}_i), \\ \widetilde{Z}_i &= \mathcal{W}_{\mathrm{bil}}(\overline{Z}_i; \widehat{B}_i) \odot V_i. \end{aligned} \tag{18}$$

The second factor in V_i marks the sender grid's valid extent. The mask is reapplied after key projection and value decoding. Only covered positions $\mathcal{P} = \{p : \sum_i V_i(p) > \epsilon\}$ receive updates, using candidates

$$\mathcal{A}(p) = \left\{(i,u) : u \in \{-1,0,1\}^2,\ V_i(p+u) > \epsilon,\ r_i > \epsilon\right\}. \tag{19}$$

Out-of-grid samples are invalid, and the local neighborhood does not expand $\mathcal{P}$. Attention includes the zero-update candidate in Equation 9, with invalid sender weights set to zero. Without a valid sender candidate, the residual update is zero.

Residual update. With $f = F_e^0(p)$ and $v = v_i(p+u)$, the residual in Equation 10 is

$$h_{i,u}(p) = \tanh(W_2\ \mathrm{SiLU}(W_1[\,f; v; v-f; |v-f|\,] + b_1) + b_2)\,. \tag{20}$$

The dense ego feature remains the residual base. Table 5 summarizes the network layers.

Table 5: Small networks used by SPC, PC, ACG, PR, and LF. BN denotes batch normalization. Convolutions preserve spatial resolution.

Component	Layers and channel dimensions
Channel encoder	Conv 3×3, $64 \to 4$; BN; ReLU
PC descriptor	Conv 3×3, $4 \to 32$; ReLU; Conv 1×1, $32 \to 32$; ℓ_2 normalization
ACG	Linear $12 \to 32$; SiLU; Linear $32 \to 1$; sigmoid
PR	Linear $12 \to 32$; ReLU; Linear $32 \to 2$; softplus
LF query / key	Conv 1×1, $64 \to 32$ / $4 \to 32$, without bias
LF value decoder	Conv 3×3, $4 \to 64$; BN; ReLU; Conv 3×3, $64 \to 64$; BN; ReLU
Zero-update score	Linear $32 \to 1$
Residual update	Linear $256 \to 64$; SiLU; Linear $64 \to 64$; tanh

Initialization. ACG's output-layer weights follow $\mathcal{N}(0, 0.01^2)$ with zero bias, giving an initial factor near 0.5. PR uses zero output-layer weights and biases of -3, with $\gamma_t = \gamma_\theta = 0$. LF's learned update scale $\beta = \sigma(b_\beta)$ starts at 0.1, and its learned offset biases start at $b_u = -0.25\|u\|_2^2$.

A.5 Training Objectives and Gradient Flow

This subsection specifies the objectives in Equations 11 and 12. Training schedules are given in Appendix B.3. We write $\mathcal{H}_\zeta$ for Smooth L1 loss with transition point ζ. The index i denotes an ego–sender pair; empty averages contribute zero.

Detection supervision. Both detection losses use $\mathcal{L}_{\mathrm{cls}} + 2\mathcal{L}_{\mathrm{box}} + 0.2\mathcal{L}_{\mathrm{dir}}$: sigmoid focal loss ($\alpha = 0.25$, $\gamma = 2$), box regression with $\mathcal{H}_{1/9}$, and two-bin direction cross-entropy. Classification weights positive and negative anchors by 2 and 1; losses are normalized by the positive-anchor count. Fused supervision averages over scenes, while local supervision averages over all participants.

Pose supervision. Let $B_i^{\mathrm{gt}} = (T_{e\leftarrow i}^{\mathrm{gt}})^{-1}$ be the clean sampling transform. The correction target is

$$C_i^{\mathrm{gt}} = B_i^{\mathrm{gt}} B_i^{-1}, \qquad \delta_i^{\mathrm{gt}} = \left(\frac{t_{x,i}^{\mathrm{gt}}}{\kappa_t}, \frac{t_{y,i}^{\mathrm{gt}}}{\kappa_t}, \frac{\theta_i^{\mathrm{gt}}}{\kappa_\theta}\right)^\top, \tag{21}$$

where translation and yaw are extracted from C_i^{gt}. The applied correction is supervised by

$$\mathcal{L}_{\mathrm{pose}} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\frac{1}{3}\sum_{j=1}^{3}\mathcal{H}_{0.25}\left(\delta_{i,j} - \delta_{i,j}^{\mathrm{gt}}\right), \tag{22}$$

with equal weights for clean and noisy pairs in $\mathcal{B}$. Accurate poses provide training targets, not inputs to matching, ACG, or PR.

Matching supervision. Let $s_{i,k,u} = d_{i,k}^{\top} d_e(\widetilde{x}_k + u)$, $\overline{w}_{i,k} = \mathrm{stopgrad}(\max(c_{i,k}, 10^{-3}))$, and $r_{i,k}^{\mathrm{fit}} = \mu_{i,k} - J_{i,k}\widehat{\eta}_i$ be the final unbounded fitting residual. Descriptor consistency, matching confidence, and geometric agreement are trained with

$$\begin{aligned} \mathcal{L}_{\mathrm{cons}} &= \mathrm{mean}_i \frac{\sum_k \overline{w}_{i,k} \left(1 - \sum_u p_{i,k}(u) s_{i,k,u}\right)}{\sum_k \overline{w}_{i,k}}, \\ \mathcal{L}_{\mathrm{ent}} &= \mathrm{mean}_i \, \mathrm{mean}_k (1 - c_{i,k}), \\ \mathcal{L}_{\mathrm{fit}} &= \mathrm{mean}_i \, \mathrm{mean}_k \, \|r_{i,k}^{\mathrm{fit}}\|_2^2. \end{aligned} \tag{23}$$

A descriptor-spread penalty discourages constant outputs:

$$\mathcal{L}_{\mathrm{var}} = \frac{1}{32} \sum_{\ell=1}^{32} \max(0, 0.10 - \mathrm{Std}_\ell), \tag{24}$$

where Std_ℓ is the population standard deviation of component ℓ across sender-center descriptors in the minibatch. Clean pairs $\mathcal{B}_0 = \{i : \|\delta_i^{\mathrm{gt}}\|_\infty \le 0.002\}$ also receive zero-offset supervision:

$$\mathcal{L}_0 = \mathrm{mean}_{i \in \mathcal{B}_0} \, \mathrm{mean}_{k \in \mathcal{K}_i^0} \left[-\log \max(p_{i,k}(0), \epsilon) \right]. \tag{25}$$

Here, $\mathcal{K}_i^0$ contains centers with a valid zero-offset candidate or no valid candidate; the latter contribute zero. The combined loss is

$$\mathcal{L}_{\mathrm{match}} = 0.1\mathcal{L}_{\mathrm{cons}} + 0.01\mathcal{L}_{\mathrm{ent}} + 0.05\mathcal{L}_{\mathrm{fit}} + 0.05\mathcal{L}_{\mathrm{var}} + 0.1\mathcal{L}_0. \tag{26}$$

Gate supervision. The target in Equation 13 scales the raw correction toward the ground truth without changing its direction. Its closed form and loss are

$$g_i^\star = \mathrm{clip}_{[0,1]} \left(\frac{(\delta_i^{\mathrm{raw}})^{\top} \delta_i^{\mathrm{gt}}}{\|\delta_i^{\mathrm{raw}}\|_2^2} \right), \qquad \mathcal{L}_{\mathrm{gate}} = \mathrm{mean}_i \, \mathcal{H}_{0.1}(g_i - g_i^\star). \tag{27}$$

The target is set to zero for negligible raw corrections.

Reliability supervision. PR predicts the remaining error in normalized correction coordinates after ACG:

$$e_{t,i}^\star = \|\delta_{i,1:2}^{\mathrm{gt}} - \delta_{i,1:2}\|_2, \qquad e_{\theta,i}^\star = |\delta_{i,3}^{\mathrm{gt}} - \delta_{i,3}|. \tag{28}$$

The loss averages the translation and yaw prediction errors:

$$\mathcal{L}_{\mathrm{rel}} = \mathrm{mean}_i \, \frac{1}{2} \left[\mathcal{H}_{0.25}(\widehat{e}_{t,i} - e_{t,i}^\star) + \mathcal{H}_{0.25}(\widehat{e}_{\theta,i} - e_{\theta,i}^\star) \right]. \tag{29}$$

Across all three datasets, Equation 12 uses $\lambda_{\mathrm{pose}} = 1$, $\lambda_{\mathrm{gate}} = 0.5$, $\lambda_{\mathrm{rel}} = 0.05$, and unit weights for both detection losses.

Gradient flow. Stage II fine-tunes all parameters along the paths in Table 6. The fusion sampling transform, ACG/PR inputs, and supervision targets are detached. Discrete center selection is not differentiated. PR's reliability scales are learned through $\mathcal{L}_{\mathrm{det}}$.

Table 6: Modules updated by each loss during robust fine-tuning.

Loss	Updated modules
$\mathcal{L}_{\mathrm{det}}$	Feature extractor, channel encoder, multiscale decoder, fused head, LF, and PR
$\mathcal{L}_{\mathrm{single}}$	Feature extractor, multiscale decoder, and local head
$\mathcal{L}_{\mathrm{pose}}$	PC, ACG, feature extractor, and channel encoder
$\mathcal{L}_{\mathrm{match}}$	Descriptor network, feature extractor, and channel encoder
$\mathcal{L}_{\mathrm{gate}}$	ACG
$\mathcal{L}_{\mathrm{rel}}$	PR error predictor

B Experimental Settings

B.1 Datasets and Evaluation Protocol

We use the provided training, validation, and test splits of OPV2V and V2XSet. For DAIR-V2X-C, we follow the training, evaluation protocol and checkpoint selection used by CoSDH (Xu et al., 2025), BlindMap (Zhu et al., 2026), and InfoCom (Wei et al., 2026), with CoAlign's complemented annotations (Lu et al., 2023). SAComm and all compared methods use the same data splits, annotations, and point-cloud and ground-truth ranges (Table 7).

Table 7: Dataset settings. Counts include the ego CAV. Ranges apply to both point clouds and ground truth.

Setting	OPV2V	V2XSet	DAIR-V2X-C
Evaluation split	Test	Test	Validation
Evaluation frames	2,170	2,834	1,789
Maximum participants	5	5	2
Communication range (m)	70	70	100
Range x (m)	$[-140.8, 140.8]$	$[-140.8, 140.8]$	$[-100.8, 100.8]$
Range y (m)	$[-38.4, 38.4]$	$[-38.4, 38.4]$	$[-40, 40]$
Range z (m)	$[-3, 1]$	$[-3, 1]$	$[-3.5, 1.5]$
Voxel size (m)	$0.4 \times 0.4 \times 4$	$0.4 \times 0.4 \times 4$	$0.4 \times 0.4 \times 5$
SAComm center ratio ρ	0.1%	0.1%	0.2%

Pose noise and AP follow Section 4.1. Noise also perturbs the ego pose; height, roll, and pitch remain unchanged. Score and NMS thresholds are 0.2 and 0.15. On OPV2V and V2XSet, participants retain their dataset ordering after range filtering. The CAV-count protocols are specified in Appendices C.1 and C.2.

B.2 Compared Methods and Communication Accounting

All methods use LiDAR inputs and PointPillars-based point-cloud encoding (Lang et al., 2019). Their BEV backbones, communication modules, and fusion modules are method-specific. Table 8 summarizes the communication settings and counted payloads.

Table 8: Communication settings and counted payloads. FP32 values, spatial indices, and cluster assignments each use 32 bits.

Method	Communication setting	Counted payload
NoFusion	No transmission	None
Early Fusion	Point-cloud sharing	FP32 coordinates and intensity
Late Fusion	Detection sharing	Seven box parameters and score (FP32)
AttFuse (Xu et al., 2022b)	Compression $32\times$; 8 channels	Dense FP32 features
Where2comm (Hu et al., 2022)	Compression $32\times$; threshold 0.01	FP32 features and spatial indices
ERMVP (Zhang et al., 2024)	Token and cluster ratios: both 0.2	FP32 representatives, positions, and assignments
CoSDH (Xu et al., 2025)	Compression $16\times$; 4/8/16 channels	Compressed FP32 features and per-scale indices
InfoCom (Wei et al., 2026)	256-D vector; top 10% mask	FP32 vector, 4-bit mask values, and indices
BlindMap (Zhu et al., 2026)	Selection ratio 1%	Multiscale FP32 features and finest-scale indices
SAComm	4 channels; 3×3 patches	Deduplicated FP32 patch features and center indices

Communication accounting. We report estimated application-level payload rates rather than measured wireless throughput. Using the same checkpoints as the detection results, we count non-ego messages over the full evaluation split at $s = 0$. The average communication rate per sender is

$$R = \frac{f}{10^6 N_{\mathrm{msg}}} \sum_{j=1}^{N_{\mathrm{msg}}} \ell(m_j) \quad \text{Mbps}, \tag{30}$$

where N_{msg} is the total number of non-ego messages, $\ell(m_j)$ is the counted payload in bits, and $f = 10\,\text{Hz}$. We count feature values at each method's configured precision and include the indices required to recover their locations. Common metadata, including poses and timestamps, and network overheads, including packet headers and link protocols, are excluded.

A spatial index identifies a selected BEV grid location using one 32-bit integer. CoSDH counts a separate index for each selected location at each scale. BlindMap's finest-scale indices identify locations on the highest-resolution grid; coarser locations are derived from them. ERMVP counts one 32-bit position index and one 32-bit cluster assignment per sampled token. InfoCom quantizes its selected mask weights to 16 levels, requiring four bits per value, and also includes their spatial indices. Late Fusion uses 256 bits per detection: seven FP32 box parameters and one FP32 score. No class ID is needed for the single-class vehicle detection task.

SAComm transmits each covered feature location once, even when patches overlap. Only patch centers require spatial indices because the fixed patch shape determines the remaining locations. With FP32 features and 32-bit center indices, the message size in Section 3.2 becomes

$$\ell(m_i) = 32d|\Omega_i| + 32K \quad \text{bits}.$$

For example, on OPV2V, $d = 4$, $K = 34$, and the mean number of unique transmitted locations is approximately 298.857 per message. At 10 Hz, these values give $R \approx 0.393\,\text{Mbps}$ per sender.

SAComm settings. SAComm uses a 3×3 LF neighborhood and 32-dimensional matching descriptors. Matching searches $\pm 1.6\,\text{m}$ along each horizontal axis at 0.2 m intervals. We set $\tau = 0.08$, $\alpha = 0.03$, $\sigma_0 = 0.05\,\text{m}$, and $\lambda = 0.05$ in Equations 4–6. One residual-reweighting step uses a Mahalanobis residual threshold of 2.0. Translation and yaw are normalized by 0.2 m and 0.2°; before ACG, each translation component is bounded by 1.6 m and yaw by 1.2°. The estimation procedure and ACG architecture are described in Appendix A.3.

B.3 Training Settings

Most methods use 30 epochs of clean pretraining. ERMVP (Zhang et al., 2024) uses 60 epochs on OPV2V to allow sufficient time for convergence. InfoCom (Wei et al., 2026) adds a second 30-epoch clean stage to learn its low-bandwidth communication representation. All methods then undergo 20 epochs of full-model robust fine-tuning, except NoFusion and Late Fusion, which share a clean-pretrained detector. Robust fine-tuning uses the noise mixture in Section 4.1. All methods use random seed 3407 for training and evaluation. Checkpoints are selected by validation loss.

SAComm uses Adam with weight decay 10^{-4} and a batch size of 2 in both training stages on all three datasets. The initial learning rate is 2×10^{-3} for clean pretraining. Robust fine-tuning uses 10^{-4} for pretrained parameters and 10^{-3} for newly introduced parameters. Learning rates are multiplied by 0.1 at epochs 10 and 20 during clean pretraining and at epochs 12 and 17 during robust fine-tuning. Training augmentation includes x-axis flipping, rotation in $[-\pi/4, \pi/4]$, and scaling in $[0.95, 1.05]$; evaluation uses none.

C Additional Quantitative Results

C.1 CAV Count Analysis on the Full Test Set

Table 9 provides the complete results for the full-test CAV-count experiment in Figure 2(a, c), reporting BEV AP_{50} and AP_{70} on all 2,170 OPV2V test frames at $s \in \{0, 0.2, 0.4, 0.6\}$. Counts are maxima and include the ego CAV; fewer may participate after 70 m range filtering based on noisy poses. Each method uses one checkpoint across counts and noise levels, including $s = 0$. Noise is assigned by frame and CAV identity with seed 3407, keeping shared CAVs' perturbations fixed. The subset experiment below fixes frames and ground truth to isolate changes in the actual count.

The rows with a maximum of five CAVs in Table 9 differ slightly from the OPV2V results in Table 1. The Gaussian noise levels are unchanged, but this sweep uses frame-and-CAV-specific noise sampling and a different random stream for point-cloud shuffling. For example, at $s = 0.6$, AP_{70} is 87.09% for SAComm and 77.60% for ERMVP here, compared with 86.98% and 77.75%, respectively, in Table 1.

Table 9: Full OPV2V test results (2,170 frames). CAVs: maximum participating count.

Method	CAVs	AP_{50} (%)				AP_{70} (%)			
		$s = 0$	$s = 0.2$	$s = 0.4$	$s = 0.6$	$s = 0$	$s = 0.2$	$s = 0.4$	$s = 0.6$
Early Fusion	2	**95.30**	92.34	77.44	64.94	89.95	72.07	49.27	38.34
	3	**96.25**	92.23	72.55	56.43	**92.31**	67.94	39.94	29.22
	4	**96.42**	92.09	71.44	54.79	**92.71**	67.11	38.52	27.63
	5	**96.69**	92.11	69.96	52.42	**93.30**	65.99	36.46	25.63
Late Fusion	2	94.49	86.29	58.66	47.04	88.76	53.20	34.15	31.14
	3	95.32	84.52	51.45	38.15	90.82	46.53	25.46	22.30
	4	95.52	84.21	50.43	36.97	91.30	45.59	24.38	21.10
	5	96.13	84.19	49.05	35.18	92.03	44.24	22.59	19.44
AttFuse (Xu et al., 2022b)	2	95.10	94.20	90.68	87.94	88.99	82.30	74.42	71.86
	3	95.81	94.45	89.94	86.77	90.89	81.83	72.14	69.47
	4	95.83	94.45	89.68	86.39	91.22	81.58	71.60	69.03
	5	96.19	94.67	89.59	86.13	91.82	81.57	71.19	68.48
Where2comm (Hu et al., 2022)	2	93.46	92.81	89.87	87.52	86.79	80.17	71.26	68.53
	3	94.21	93.16	89.03	86.04	88.96	79.77	68.35	64.86
	4	94.32	93.15	88.78	85.76	89.37	79.56	67.69	64.21
	5	94.67	93.38	88.73	85.54	89.99	79.57	67.09	63.55
ERMVP (Zhang et al., 2024)	2	94.86	94.17	92.54	90.88	86.46	84.69	82.14	80.19
	3	95.32	94.21	91.97	89.73	87.12	84.62	81.15	78.75
	4	95.52	94.29	91.79	89.54	87.14	84.49	80.77	78.47
	5	95.88	94.49	91.76	89.21	87.27	84.24	80.20	77.60
CoSDH (Xu et al., 2025)	2	92.76	92.02	88.74	84.60	85.60	81.13	73.30	68.33
	3	93.25	92.08	87.25	81.80	87.02	80.51	69.71	63.43
	4	93.54	92.16	86.47	80.74	87.67	80.04	68.35	62.00
	5	93.60	92.12	86.15	80.07	87.77	79.76	67.50	60.94
InfoCom (Wei et al., 2026)	2	94.66	93.75	90.97	88.60	87.07	82.37	76.72	74.17
	3	95.41	94.06	90.22	87.42	88.76	82.07	74.81	72.02
	4	95.51	94.09	89.94	86.91	89.27	81.96	74.40	71.62
	5	95.72	94.12	89.74	86.58	89.58	81.77	73.94	70.89
BlindMap (Zhu et al., 2026)	2	93.92	92.97	88.05	83.86	87.91	80.33	70.27	66.32
	3	94.73	93.11	86.67	81.38	89.64	79.28	66.85	62.50
	4	94.96	93.20	86.23	80.69	90.13	78.91	66.25	61.70
	5	95.46	93.47	85.99	80.14	90.73	78.48	65.17	60.58
SAComm	2	**95.30**	**95.12**	**94.47**	**93.51**	**90.37**	**89.44**	**88.19**	**86.52**
	3	95.84	**95.62**	**94.92**	**93.57**	91.62	**90.41**	**89.01**	**86.67**
	4	96.09	**95.85**	**95.10**	**93.69**	91.96	**90.78**	**89.24**	**86.91**
	5	96.48	**96.22**	**95.43**	**93.92**	92.45	**91.19**	**89.60**	**87.09**

C.2 CONTROLLED FIVE-CAV SUBSET

Table 10 provides the complete results for the controlled five-CAV experiment in Figure 2(b, d), reporting BEV AP_{50} and AP_{70} at $s \in \{0, 0.2, 0.4, 0.6\}$. We evaluate all 101 frames in the OPV2V test set where five CAVs, including the ego CAV, are available within the 70 m communication range under clean poses. For each frame, we fix the ego CAV and add collaborators in order of increasing clean distance, giving nested sets of two, three, four, and five participating CAVs. Membership is determined before adding pose noise. For each method, the clean annotation union from all five CAVs is reused as ground truth across every count and noise setting. Checkpoints and deterministic noise assignments follow the full-test experiment. This single-scene subset provides a controlled comparison rather than evidence of generalization across scenes.

Table 10: Controlled five-CAV subset results (101 frames). CAVs: actual participating count.

Method	CAVs	AP_{50} (%)				AP_{70} (%)			
		$s=0$	$s=0.2$	$s=0.4$	$s=0.6$	$s=0$	$s=0.2$	$s=0.4$	$s=0.6$
Early Fusion	2	96.02	88.72	65.47	48.71	88.04	59.51	33.56	24.10
	3	96.77	88.70	58.88	38.78	90.85	54.19	23.73	14.79
	4	**97.20**	89.57	58.86	37.24	92.06	55.55	23.73	13.12
	5	97.15	88.69	59.27	36.49	92.08	55.27	23.47	12.92
Late Fusion	2	95.89	82.89	55.58	43.99	85.63	47.97	33.26	31.37
	3	96.71	80.18	47.84	34.26	90.45	41.18	23.08	20.21
	4	96.83	81.13	47.15	31.34	92.35	41.51	20.42	16.82
	5	96.81	80.47	47.88	31.16	92.19	41.88	19.96	16.13
AttFuse (Xu et al., 2022b)	2	**96.45**	93.36	86.58	82.37	85.25	75.22	66.64	63.57
	3	96.78	94.16	87.15	82.10	90.37	76.81	65.78	62.25
	4	96.69	94.16	86.76	81.68	91.35	77.51	65.30	60.97
	5	96.07	93.50	86.52	81.53	91.32	76.90	65.16	60.69
Where2comm (Hu et al., 2022)	2	90.67	88.96	84.06	81.04	82.97	74.08	65.20	62.81
	3	93.40	91.22	85.51	81.06	88.36	74.50	62.41	59.42
	4	93.30	91.31	85.14	80.57	88.80	75.38	60.51	56.61
	5	93.02	90.59	84.49	80.27	88.35	74.22	60.06	55.28
ERMVP (Zhang et al., 2024)	2	95.92	94.18	90.06	86.48	84.22	80.70	76.60	73.78
	3	**97.02**	94.98	89.90	85.93	87.52	81.49	75.36	71.86
	4	97.16	94.61	90.22	85.62	88.61	81.65	75.35	71.03
	5	**97.23**	94.31	89.33	83.94	88.29	79.00	70.87	65.70
CoSDH (Xu et al., 2025)	2	89.20	87.86	82.24	77.29	76.69	69.80	63.36	58.89
	3	90.06	88.21	80.11	73.14	82.66	71.31	59.68	53.64
	4	90.08	88.03	79.93	71.71	83.56	72.05	58.25	52.16
	5	90.06	87.57	78.73	71.34	83.66	71.71	58.34	51.57
InfoCom (Wei et al., 2026)	2	94.16	91.71	86.27	82.54	81.12	73.32	68.37	65.85
	3	95.05	92.33	86.58	82.35	86.12	75.37	66.90	63.50
	4	95.00	92.58	86.26	81.95	88.06	76.94	66.90	62.11
	5	94.60	91.82	85.67	81.47	87.61	76.17	66.35	61.32
BlindMap (Zhu et al., 2026)	2	92.59	89.40	82.47	77.75	83.27	70.92	61.44	58.47
	3	95.55	91.58	80.62	73.61	89.55	70.74	57.80	53.54
	4	95.69	91.77	80.98	72.93	90.86	71.86	57.28	52.09
	5	95.69	91.63	81.02	72.49	90.89	71.24	57.53	51.57
SAComm	2	95.79	**95.46**	**94.51**	**92.55**	**89.02**	**87.44**	**85.57**	**81.78**
	3	96.18	**95.92**	**95.04**	**93.17**	**91.34**	**89.79**	**88.29**	**83.22**
	4	96.21	**95.98**	**95.06**	**93.25**	**92.71**	**91.30**	**89.53**	**84.67**
	5	96.21	**95.91**	**95.06**	**93.39**	**92.63**	**91.20**	**89.76**	**85.50**

C.3 Matching Quality and Pose Correction

Table 11 examines which properties of sparse matches are associated with more accurate pose correction. Using the same fixed SAComm checkpoint as Table 4, we evaluate all 2,170 OPV2V test frames at each noise level and analyze the resulting 13,792 ego–sender pairs across the four noise levels. These pairs come from 2,051 frames with collaborating senders in 16 scenes at each noise level.

We compare three measures. Match count is the number of patch centers with valid matching candidates. Spatial spread measures how widely these centers are distributed, using the trace of their weighted position covariance. Information measures the strength of their combined pose constraints along the weakest direction, using the smallest eigenvalue of $H_i = \sum_{k \in \mathcal{V}_i} J_k^\top \Lambda_k J_k$, where $\mathcal{V}_i$ contains centers with valid candidates. It accounts for both spatial layout and match reliability through the final fitting weights from Equation 6, excluding the prior. We combine normalized translation and yaw errors as

$$\mathcal{E}_i = \sqrt{\left(\frac{e_{t,i}}{0.2\,\mathrm{m}}\right)^2 + \left(\frac{e_{\theta,i}}{0.2^\circ}\right)^2}, \tag{31}$$

where $e_{t,i}$ is the translation error magnitude and $e_{\theta,i}$ is the absolute yaw error of the relative pose. We compute this error before correction, after PC, and after PC+ACG using the corresponding transforms from the same forward pass.

The table reports adjusted rank correlations with residual pose error after ACG. Within each noise level, we rank the variables, account for sender distance, initial translation and yaw errors, and scene differences, and correlate the adjusted ranks. At $s = 0$, we account only for distance and scene because the initial pose errors are zero. Negative values mean that higher values of a measure are associated with smaller residual errors; larger negative magnitudes indicate stronger associations, not percentage reductions in error. Brackets give 95% confidence intervals from 2,000 whole-scene bootstrap samples, with ranks and adjustment recomputed in each sample.

Table 11: Adjusted rank correlations with residual pose error after ACG. Brackets: 95% whole-scene bootstrap intervals.

Measure	$s = 0.2$	$s = 0.4$	$s = 0.6$
Match count	-0.076 $[-0.155, -0.001]$	-0.131 $[-0.212, -0.050]$	-0.149 $[-0.225, -0.083]$
Spatial spread	-0.106 $[-0.199, -0.024]$	-0.148 $[-0.236, -0.061]$	-0.133 $[-0.222, -0.055]$
Information	-0.300 $[-0.364, -0.237]$	-0.348 $[-0.413, -0.281]$	-0.415 $[-0.470, -0.352]$

Information has a stronger negative association with residual error than match count or spatial spread at every nonzero noise level. At $s = 0.6$, the correlations are -0.415, -0.149, and -0.133, respectively. The negative association with information also remains before ACG, so it is not solely due to the gate. No clear association is observed at zero noise. These results suggest that match count alone does not adequately describe the usefulness of sparse matching evidence. They support our motivation to preserve matching cues in compact messages and account for matching uncertainty and geometric consistency during pose estimation.

C.4 Adaptive Correction Gate

We examine whether ACG assigns larger correction factors to more useful pose corrections, using the same fixed checkpoint and ego–sender pairs as Appendix C.3. Using the normalized error in Equation 31, correction gain is $\mathcal{E}_i^{\text{input}} - \mathcal{E}_i^{\text{PC}}$, with positive values indicating useful corrections. This gain measures the effect of PC before ACG scales the correction. All measurements come from the same forward pass.

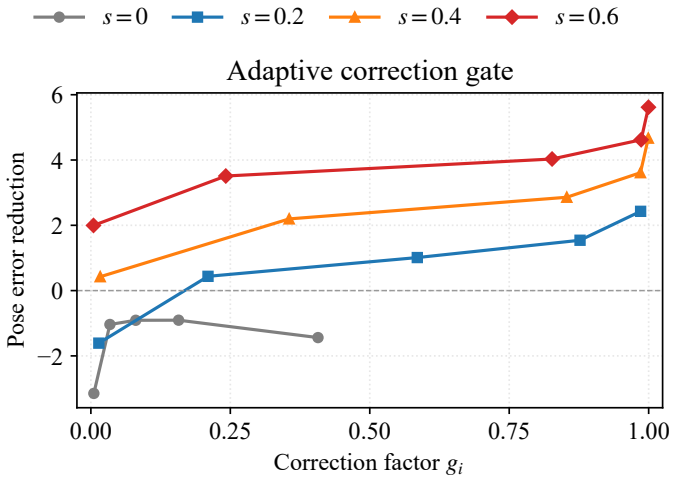


Figure 5: ACG on OPV2V. Points show mean correction factors and mean normalized error reductions after PC in five equal-frequency groups per noise level. Positive gains indicate useful corrections.

Table 12: Spearman correlations between the correction factor and normalized error reduction after PC, computed separately at each noise level.

s	0	0.2	0.4	0.6
Correlation	0.142	0.691	0.564	0.400

Correction factor and correction gain. Figure 5 shows that larger correction factors generally accompany greater error reductions at nonzero noise. Table 12 reports the corresponding Spearman correlations, computed separately at each noise level. At $s = 0.2$, the lowest-factor group has a negative mean gain, while the other groups benefit from correction. This is consistent with ACG assigning less weight to corrections that are less useful. At $s = 0.6$, even the lowest-factor group has a positive mean gain. ACG can therefore also reduce useful corrections under stronger noise, consistent with Tables 2 and 4.

C.5 Effect of Training Noise Coverage

We examine whether broader training-noise coverage improves detection under strong pose noise and allows ACG to retain more of the estimated correction. The standard schedule samples $s = 0$ or 0.2 with equal probability. In the expanded schedule, we retain a probability of 0.5 for $s = 0$ and distribute the remaining probability equally over $s \in \{0.2, 0.4, 0.6\}$. Both runs start from the same clean-pretrained checkpoint and use 20 epochs of full-model robust fine-tuning, with identical message and fusion settings, losses, and optimizer schedules. Checkpoints are selected by validation loss: epoch 19 for the standard run and epoch 18 for the expanded run. Both use seed 3407. Evaluation uses all 2,170 test frames with identical sampled pose perturbations at each noise level.

Table 13: Training-noise coverage on OPV2V. Input: mean relative pose error before correction. PC and PC+ACG: residual errors from the same forward pass of each model. AP_{70} uses PC+ACG. Both schedules keep 50% zero-noise scenes.

s	Training	AP_{70} (%)	Translation (m)			Yaw (°)			Mean g_i
			Input	PC	PC+ACG	Input	PC	PC+ACG	
0	Standard	92.41	0.0000	0.1787	0.0247	0.0000	0.2111	0.0239	0.137
	Expanded	92.16		0.1876	0.0472		0.2106	0.0403	0.112
0.2	Standard	91.25	0.3703	0.1851	0.1786	0.2227	0.2200	0.1560	0.534
	Expanded	91.09		0.1820	0.1809		0.2148	0.1646	0.578
0.4	Standard	89.76	0.7426	0.2303	0.3023	0.4514	0.2621	0.2834	0.642
	Expanded	90.51		0.2237	0.2413		0.2534	0.2441	0.782
0.6	Standard	86.98	1.0933	0.3687	0.5451	0.6969	0.4009	0.4936	0.612
	Expanded	88.40		0.3678	0.4065		0.3940	0.4123	0.808
0.8	Standard	84.36	1.4437	0.6385	0.9082	0.9338	0.6259	0.7483	0.528
	Expanded	86.33		0.6190	0.6927		0.6010	0.6328	0.775

Table 13 shows that expanded training coverage improves detection and reduces residual pose errors at $s = 0.4$, 0.6, and 0.8. At $s = 0.6$, AP_{70} increases by 1.42 percentage points. Mean translation error after PC is nearly unchanged (0.3687 m versus 0.3678 m), while the error after PC+ACG decreases from 0.5451 m to 0.4065 m. The mean correction factor also increases from 0.612 to 0.808. Together, these results suggest that expanded training coverage reduces the suppression of useful corrections. These gains come with small AP_{70} reductions of 0.25 and 0.16 percentage points at $s = 0$ and 0.2, respectively. Thus, the balance between low-noise accuracy and high-noise correction depends in part on training-noise coverage, rather than on the receiver architecture alone. Because all parameters are fine-tuned, this comparison does not isolate ACG from changes in matching or fusion. The $s = 0.8$ setting lies outside both training schedules and is included only as an additional diagnostic. The expanded schedule is not used in the main comparison with other methods.

D Additional Qualitative Results

D.1 Detection Comparisons across Datasets

We extend Section 4.3 with two examples each from OPV2V, V2XSet, and DAIR-V2X-C. Figures 6–11 compare Early Fusion, Where2comm (Hu et al., 2022), ERMVP (Zhang et al., 2024), InfoCom (Wei et al., 2026), BlindMap (Zhu et al., 2026), and SAComm. Each figure shows the full detection views above and local enlargements below, with the same 2×3 method layout in both groups. Red dashed rectangles mark the enlarged region, which is fixed across methods. All examples use $s = 0.6$, with identical participating agents, ground truth, and pose perturbations across methods. Predictions use a score threshold of 0.2. Red boxes denote ground truth, and green boxes denote predictions. Point clouds use true poses for visualization only, while inference uses noisy poses. The marked OPV2V regions are also used for the patch alignment analysis below.

Detection under pose noise. Across the examples, several comparison methods exhibit shifted boxes, missed detections, or additional predictions under pose noise. SAComm more closely follows the ground-truth box positions and orientations in the highlighted regions, particularly in the V2XSet examples. The OPV2V examples are further analyzed below to visualize how pose correction changes the alignment of the transmitted patches.

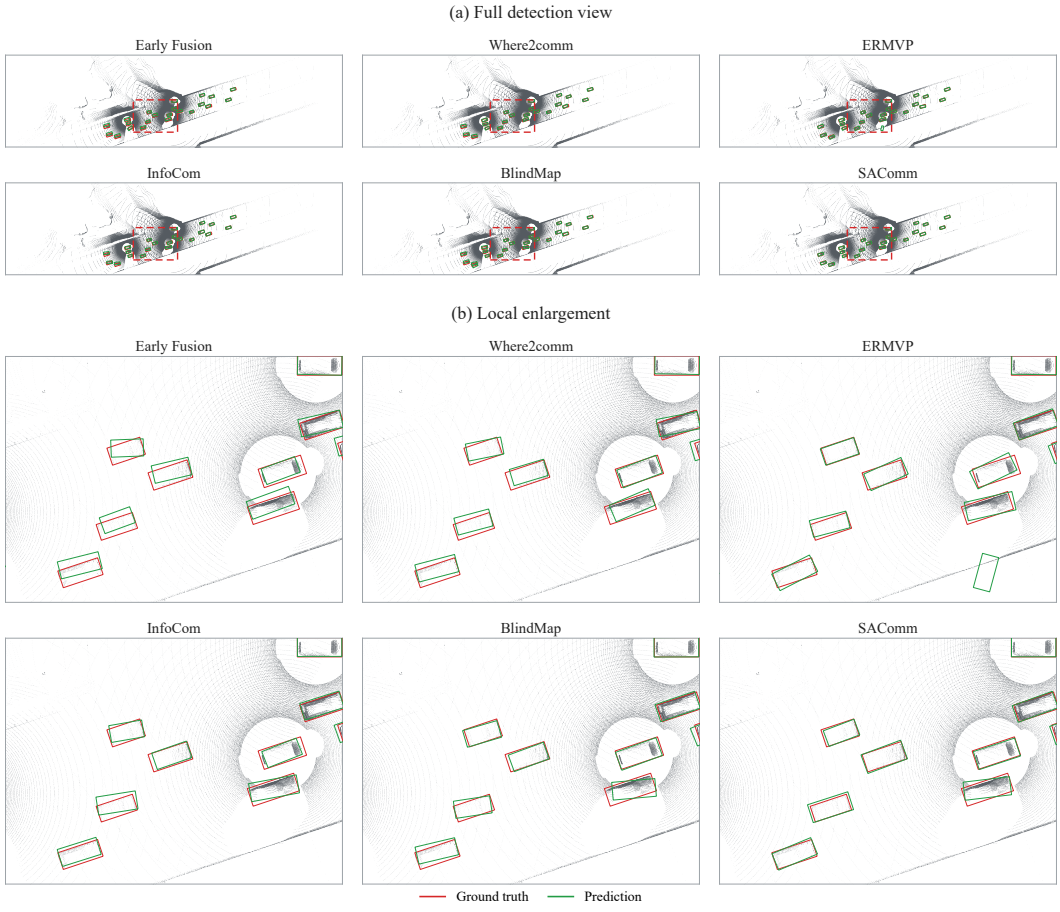


Figure 6: Detection on OPV2V at $s = 0.6$. (a) Full views. (b) Enlargements of the dashed regions. Patch alignment is shown in Figure 12.



Figure 7: Detection on OPV2V at $s = 0.6$. (a) Full views. (b) Enlargements of the dashed regions. Patch alignment is shown in Figure 13.

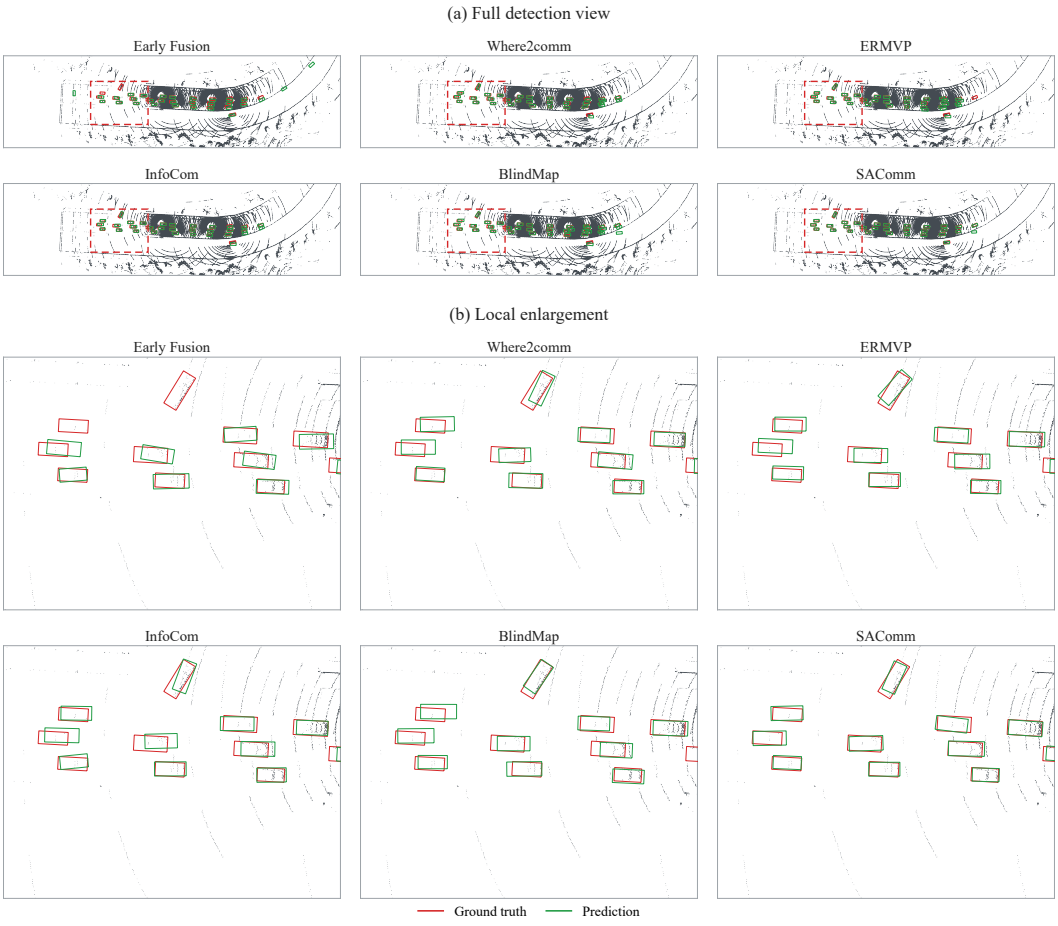


Figure 8: Detection on V2XSet at $s = 0.6$. (a) Full views. (b) Enlargements of the dashed regions.

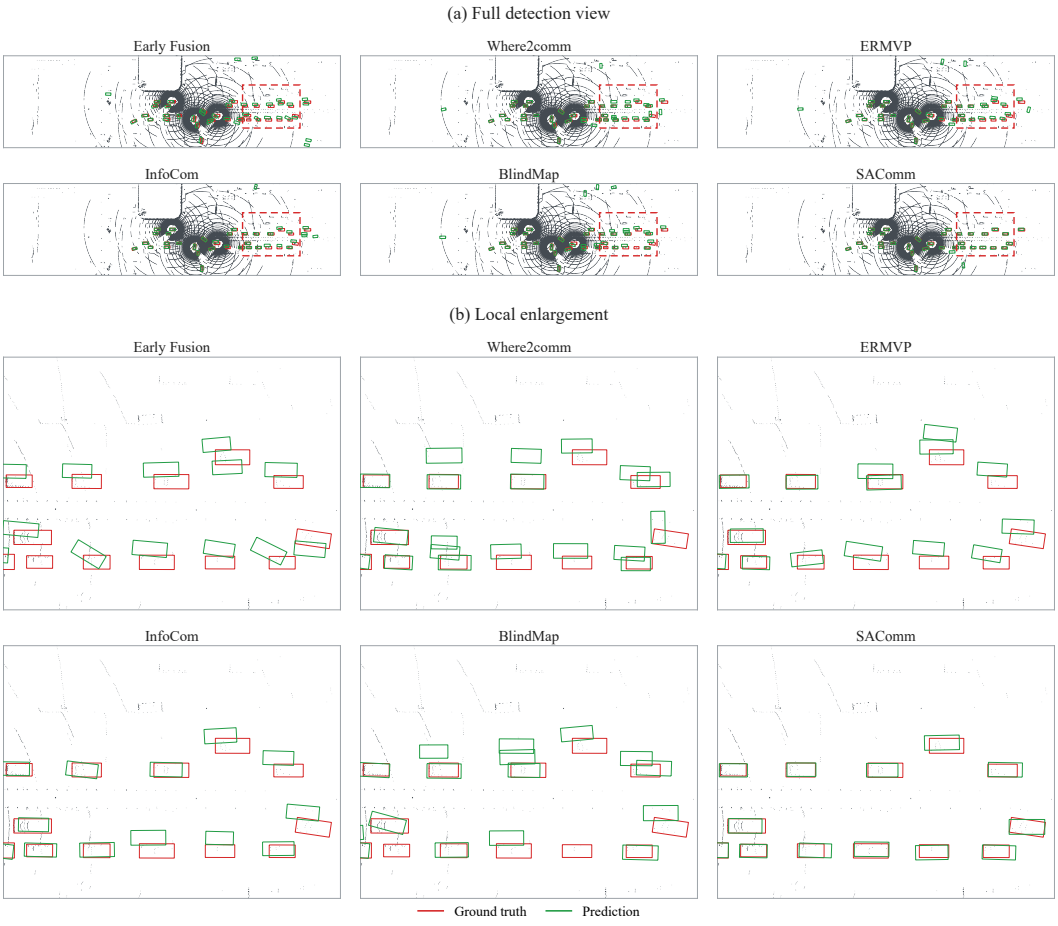


Figure 9: Detection on V2XSet at $s = 0.6$. (a) Full views. (b) Enlargements of the dashed regions.

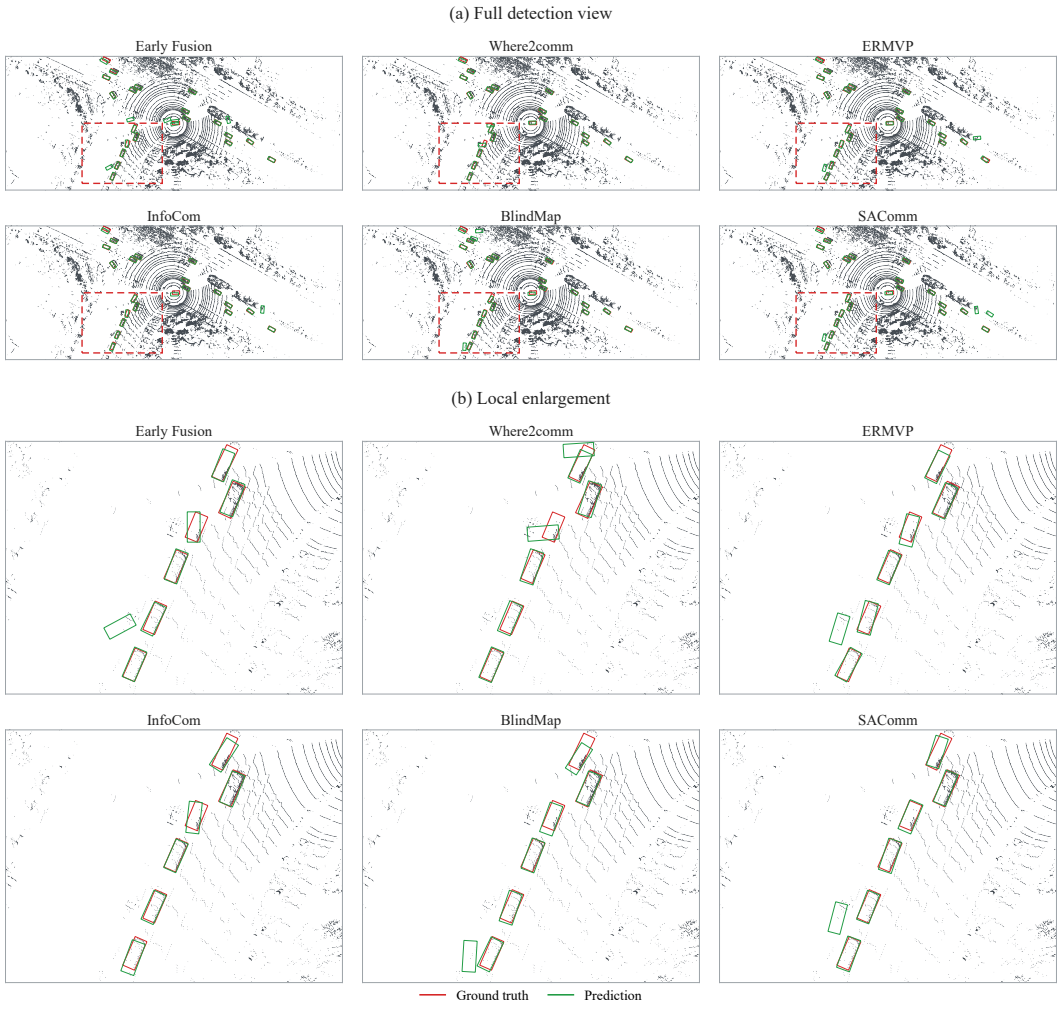


Figure 10: Detection on DAIR-V2X-C at $s = 0.6$. (a) Full views. (b) Enlargements of the dashed regions.

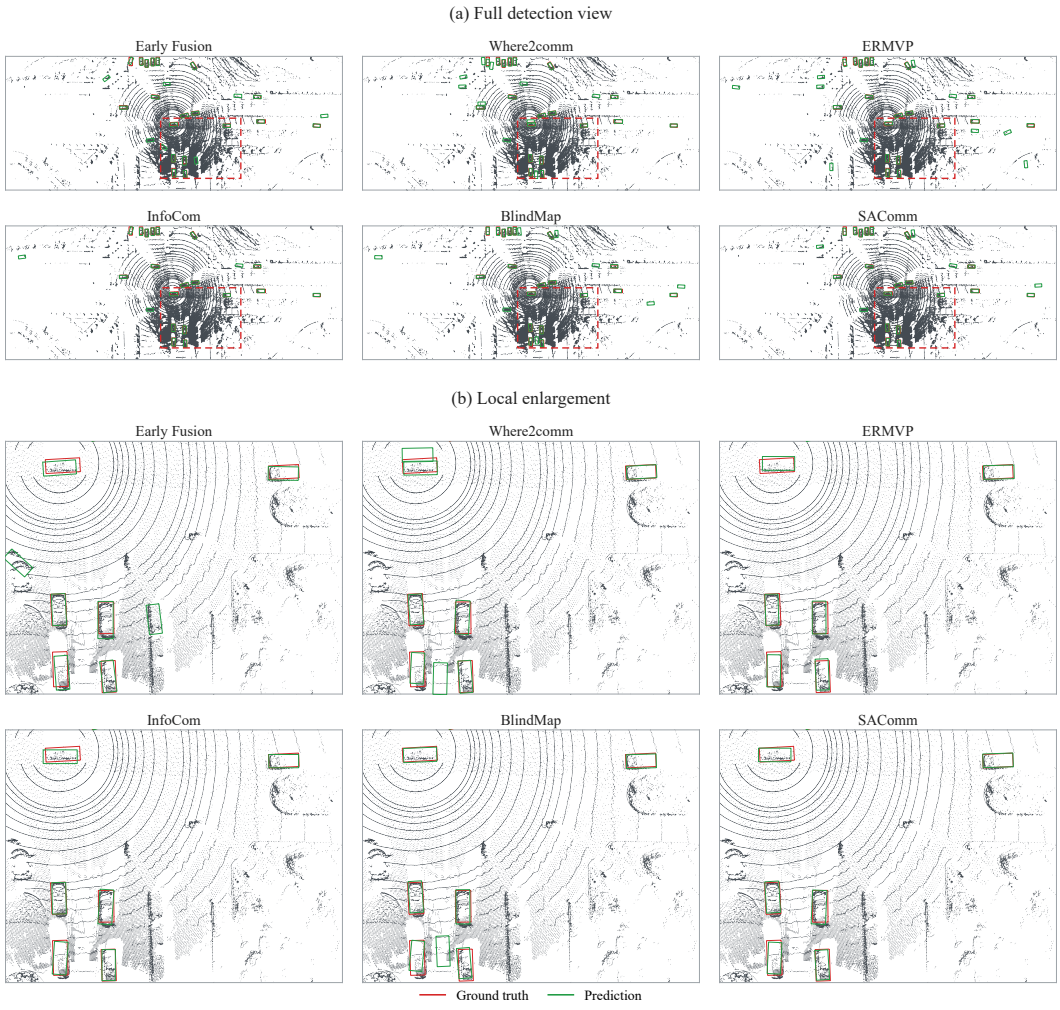


Figure 11: Detection on DAIR-V2X-C at $s = 0.6$. (a) Full views. (b) Enlargements of the dashed regions.

D.2 Additional Patch Alignment Examples

Figures 12 and 13 show the full detection scene, local detections, and patch alignment for two OPV2V examples at $s = 0.6$. Detection uses all participating CAVs. The patch views focus on one sender and compare its message under the noisy relative pose $\widetilde{T}_{e\leftarrow i}$ with the corrected pose $\widehat{T}_{e\leftarrow i}$ after applying the gated correction in Equation 7.

Reading the patch views. Blue patches have valid matching candidates and are classified as geometrically consistent for visualization. Orange patches are the remaining transmitted patches. This grouping is used only for visualization; pose estimation itself uses continuous fitting weights rather than this binary classification. Black crosses mark the sender's patch centers under the true relative pose. Translation and yaw errors refer to the relative pose of the entire message, not the average patch displacement within the crop.

Alignment of the entire message. Relative translation error decreases from 1.165 m to 0.125 m in Figure 12 and from 1.401 m to 0.219 m in Figure 13, with corresponding reductions in yaw error. Both correction factors are close to one, illustrating the applied correction under strong pose noise. The orange patches also move closer to their true-pose references. This illustrates how sparse matching evidence can correct the shared relative pose and thereby align the entire message, including the other transmitted patches shown in orange.

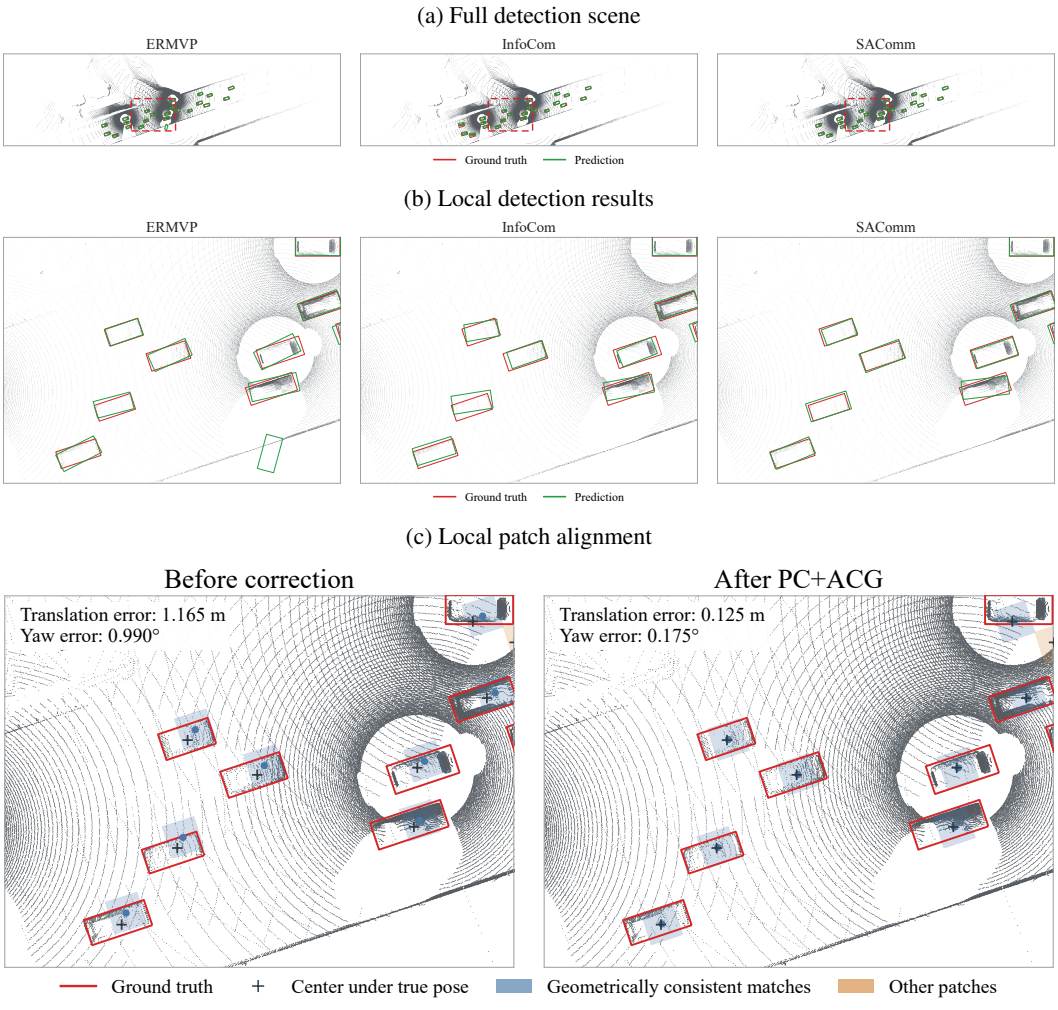


Figure 12: Detection and patch alignment on OPV2V at $s = 0.6$. Panels (b) and (c) enlarge the dashed region in (a). Patch alignment shows one sender; detection uses all participating CAVs.

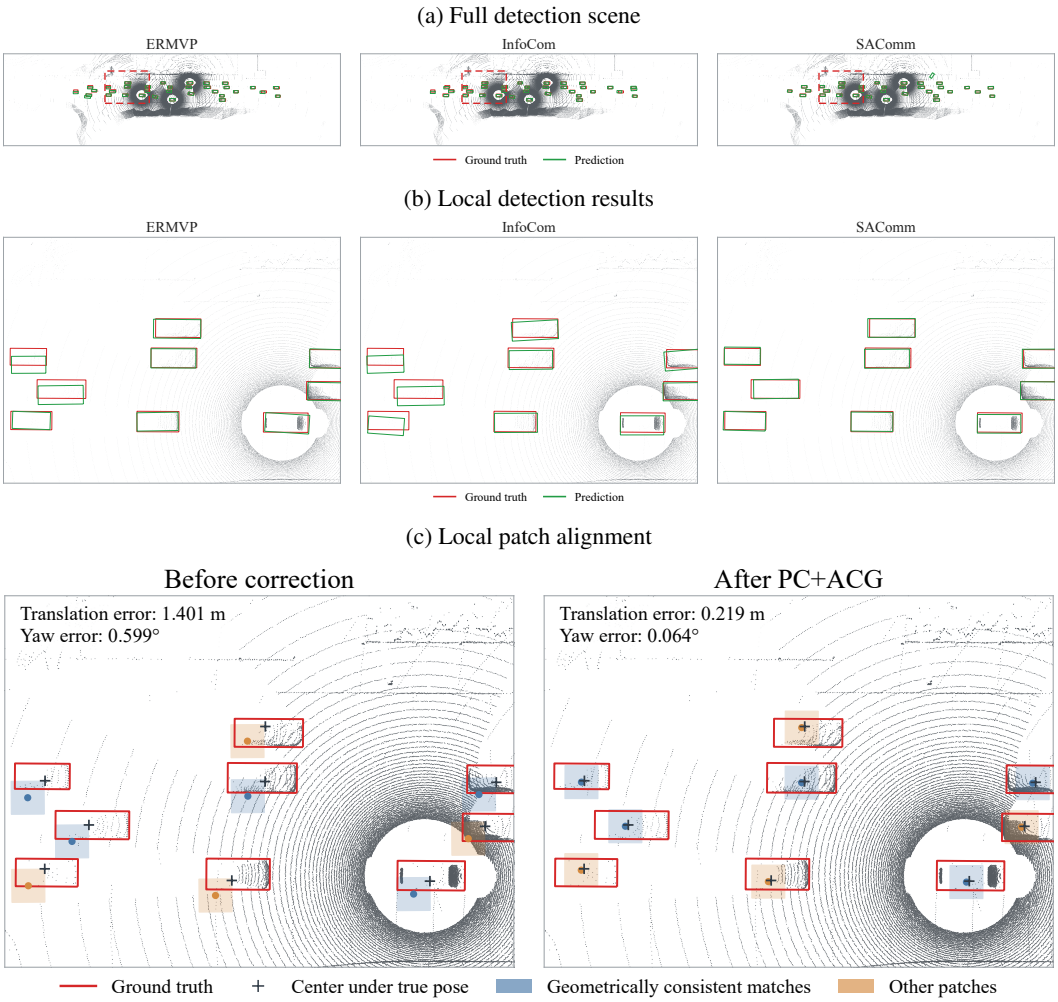


Figure 13: Detection and patch alignment on OPV2V at $s = 0.6$. Panels (b) and (c) enlarge the dashed region in (a). Patch alignment shows one sender; detection uses all participating CAVs.