CoAnchor: Robust Collaborative Perception under Spatio-Temporal Misalignment via Object-Level Anchors

Chi Li
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
lichi@bupt.edu.cn

Rui Lin
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
lr_507@bupt.edu.cn

Aobo Ji
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
bobojassp@bupt.edu.cn

Dongzhu Xu*
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
xudongzhu@bupt.edu.cn

Abstract

Collaborative perception extends the sensing range of a single vehicle by fusing observations from nearby agents, which improves the robustness of autonomous driving. In realistic deployments, however, the received collaborator messages are often affected by both communication delay and relative-pose noise, which jointly cause stale observations, spatial misalignment, and unstable feature fusion. Existing methods usually address these issues from either the spatial or temporal side, but handling them jointly in a unified and efficient manner remains challenging. In this paper, we propose *CoAnchor*, an anchor-centric spatio-temporal alignment framework for asynchronous collaborative perception. Instead of directly reasoning on dense BEV features, *CoAnchor* builds sparse object-level spatio-temporal anchors as a shared interface for pose correction and tightly connects spatial refinement, temporal propagation, and current-time verification within one unified loop, while keeping the overall correction process lightweight. Extensive experiments on both simulated and real-world datasets illustrate that *CoAnchor* remains competitive under clean settings and improves the robustness under joint delay and pose perturbations with a favorable practical accuracy-efficiency trade-off.

CCS Concepts

• **Computing methodologies** → **Object detection**; *Multi-agent systems*.

Keywords

Collaborative Perception, Spatio-Temporal Feature Alignment

*Corresponding author.

MM '26, Rio de Janeiro, Brazil

ACM ISBN 979-8-4007-2213-4/2026/11
https://doi.org/10.1145/3767308.3835460

ACM Reference Format:
Chi Li, Rui Lin, Aobo Ji, and Dongzhu Xu. 2026. *CoAnchor*: Robust Collaborative Perception under Spatio-Temporal Misalignment via Object-Level Anchors. In *Proceedings of the 34th ACM International Conference on Multimedia (MM '26), November 10–14, 2026, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3767308.3835460

1 Introduction

Multi-agent collaborative perception has become a significant advantage for autonomous driving, because neighboring agents can share complementary observations beyond the field of view of a single vehicle [15, 25, 26, 29]. By exchanging rich perception information through the V2X communication technology [6, 20], the ego vehicle can better detect distant, occluded, or partially visible objects, thereby improving the scene understanding in challenging traffic environments [5, 8, 28].

Despite the huge potential, collaborative perception is highly sensitive to the synchronization of cross-agent communications [13, 14, 19, 20, 29]. As illustrated in Fig. 1, in realistic deployments, the received collaborator messages are rarely perfectly aligned with the ego vehicle's current observation due to two types of perception errors. (i) *Temporal misalignment*: due to communication delay, collaborator messages from neighboring agents are captured earlier than the ego timestamp, so the shared observations may already be stale when they arrive at the ego vehicle. (ii) *Spatial misalignment*: localization noise and relative-pose error can place the collaborator observations at incorrect locations in the ego frame. More seriously, these two factors interweave in practice. If the ego vehicle directly fuses the temporally- and spatially-misaligned collaborator information, this can easily cause shifted detections, duplicated responses, or motion ghosts in the final object perception.

Existing methods usually address these issues by treating temporal and spatial misalignment separately. *On the one hand*, spatially oriented methods [2, 8, 15, 36] refine cross-agent alignment before feature fusion by correcting noisy relative poses or matching object-level observations across agents. They are effective when collaborator observations remain sufficiently synchronized and when reliable correspondences can still be established. *On the other*

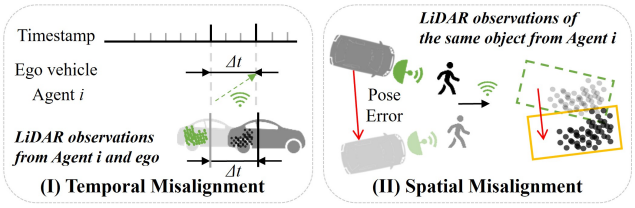


Figure 1: Temporal and spatial misalignment between ego vehicle and cooperative agents.

hand, temporally oriented methods [14, 19, 23, 25] compensate stale collaborator information by propagating or reconstructing delayed features toward the ego vehicle's current time. These methods are strong when the spatial reference is already reliable. Nevertheless, in practical collaborative perception, delay and pose noise are coupled rather than isolated: a biased pose estimate may change the starting point of temporal propagation, while propagated collaborator information should also be re-evaluated at the current ego time before the object feature fusion process. Therefore, simply handling spatial correction and temporal compensation in sequence is often insufficient under realistic spatio-temporal misalignment. This motivates us to seek a unified representation that can couple delayed cross-agent correspondence, temporal propagation, current-time verification, and feature fusion.

In this paper, we propose ***CoAnchor***, an anchor-centric collaborative perception framework for spatio-temporally misaligned messages. Our key observation is that, although high-dimensional BEV features are powerful for object detection, they do not explicitly expose cross-agent object correspondence or temporal structure under joint delay and pose perturbations. By contrast, a low-dimensional object-level representation can provide a shared interface for jointly reasoning about spatial alignment, temporal propagation, and reliability estimation. Based on this observation, we represent delayed collaborator objects as *spatio-temporal anchors*, each of which corresponds to a matched cross-agent object hypothesis together with its motion state and reliability cues over time. These anchors connect delayed-time correspondence and pose refinement, current-time propagation and verification, closed-loop reliability feedback, and anchor-guided feature fusion within a unified pipeline.

Compared with directly performing pose corrections in dense BEV feature space, *CoAnchor*'s anchor-centric design has the following advantages. *(i)* It couples the temporal propagation with current-time posterior verification tightly, so the collaborator perceptive information can be calibrated before the object fusion, to mitigate the effects of communication delay and pose noise. *(ii)* It mainly acts on sparse anchors rather than repeatedly invoking heavy-cost BEV feature processing modules. Thus, it remains efficient in practice, and one feedback round can provide a better accuracy-efficiency trade-off.

Extensive experiments on OPV2V [30] and real-world V2V4Real [28] demonstrate that *CoAnchor remains competitive in clean settings and becomes consistently more robust under coupled pose perturbation and communication delay*. In particular, on V2V4Real under a delay of 200 ms and pose noise of $(0.6\,\text{m}, 0.6^\circ)$, *CoAnchor* achieves 67.04 AP@0.5, outperforming the state-of-the-art spatio-temporal cascade baseline by 6.00 AP@0.5, corresponding to a 9.83% relative improvement.

We next summarize the contributions. *(i)* We analyze why collaborative perception degrades under joint spatio-temporal misalignment caused by communication delay and relative-pose noise. *(ii)* We propose *CoAnchor*, an anchor-centric framework that uses currently verifiable object-level anchors to tightly connect spatial correction, temporal propagation, and collaborative fusion, instead of treating them as independent stages. *(iii)* Extensive evaluations show that *CoAnchor* achieves improved robustness with a favorable accuracy–efficiency trade-off.

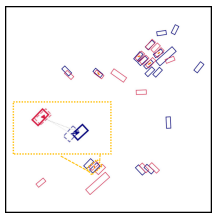
2 Related Work

Collaborative perception. Multiple agents can improve the scene understanding by exchanging sensory information or intermediate representations [7, 22]. According to the stage at which information is fused, existing methods are commonly divided into early [3], intermediate, and late fusion [17, 18]. Intermediate-fusion methods offer a better balance between detection accuracy and transmission cost, and have therefore become the dominant paradigm [6, 27, 31, 32]. However, practical collaborative perception needs to cope with communication delay, localization noise, and imperfect cross-agent synchronization, which make multi-agent fusion significantly more challenging in real deployments [1, 4].

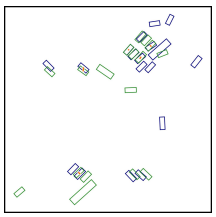
Spatial calibration under pose noise. A line of research focuses on the degradation caused by relative-pose noise [16, 21, 34]. CoAlign [15] formulates collaborative calibration through agent-object pose graph optimization and improves robustness without requiring precise relative-pose supervision. RoCo [8] further models pose correction as iterative object matching and pose adjustment, emphasizing the importance of reliable cross-agent correspondences under noisy conditions. These methods are effective when observations remain approximately synchronized and cross-agent correspondences are reliable. However, under asynchronous collaboration, spatial refinement alone cannot resolve temporal staleness, and its performance may degrade when the matched landmarks are noisy or incorrect.

Temporal compensation under communication delay. Another line of work studies how to compensate stale collaborator information under communication delay [25, 33]. SyncNet [14] reconstructs current-time features from historical observations through temporal feature interaction. TraF-Align [19] introduces trajectory-aware feature alignment, where low-dimensional motion cues guide deformable attention along temporally aligned object trajectories. These methods substantially improve robustness to latency when the delayed collaborator message has already been placed in a reliable spatial frame. However, they usually assume the incoming feature as a valid starting point, so residual pose bias can still be propagated forward and appear as duplicated responses or motion ghosts after fusion.

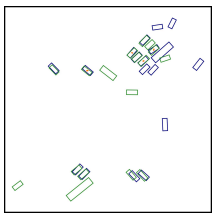
Optimization under spatio-temporal misalignment. Recent methods attempt to improve robustness under both spatial and temporal perturbations [24, 35, 37]. V2X-ViT [29] exhibits partial robustness to noisy collaboration through transformer-based feature aggregation. CoST [20] introduces temporal context by treating historical features as delayed collaborators and retrieving them through deformable sampling. CoDiff [9] uses conditional diffusion to denoise and progressively refine fused feature representations



(a) Raw detection.



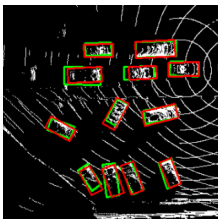
(b) Compensated result.



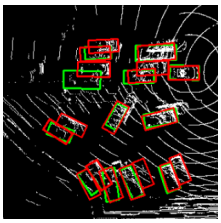
(c) Ideal detection.

Original neighbor box
Current ego box
CoAlign-corrected neighbor box
Velocity
Neighbor box two frames earlier
Ego box two frames earlier
Cluster matching relation

Figure 2: Showcase of spatial alignment outputs (detected bounding boxes) for raw detections, CoAlign-compensated detections, and ideal detections.



(a) w/o pose noise with a 200 ms delay.



(b) Delay: 200 ms Pose noise: $(0.6\,\text{m}, 0.6^\circ)$

Figure 3: Because of residual spatial errors, many motion ghosts remain even after temporal compensation.

corrupted by delay and pose errors. Nevertheless, under large spatio-temporal perturbations, these methods still predominantly depend on high-dimensional feature interaction to implicitly reconstruct the aligned scene. This implicit strategy becomes unreliable under large spatio-temporal perturbations, because the model has to recover object correspondence, motion changes, and feature misalignment all from noisy high-dimensional features. Consequently, scene recovery may become unreliable when delayed observations are both temporally outdated and spatially biased.

Different from these approaches, our method is built around *object-level anchors*. Instead of performing all corrections directly in dense BEV feature space, we construct a low-dimensional object-level representation and achieve fusion within one unified pipeline.

3 Measurement and Motivation

3.1 Measurement Setups and Results

To address the challenge of spatio-temporal asynchrony in collaborative perception, a straightforward strategy is to cascade a spatial alignment module with a temporal compensation module. For instance, one can combine CoAlign [15], which mitigates spatial errors via agent-object pose graph optimization, with TraF-Align [19], which compensates communication delay through trajectory-aware feature alignment. We denote this cascade solution as *baseST*.

A pilot study shows that such a feed-forward cascade does not provide a stable plug-and-play gain under joint delay and pose noise. On V2V4Real, under a fixed 100 ms delay, inserting CoAlign before TraF-Align reduces AP@0.5 / AP@0.7 from 78.76 / 50.80 to 73.16 / 43.86 even without pose noise. When pose noise of $(0.6\,\text{m}, 0.6^\circ)$ is further introduced under the same delay, the cascaded result drops to 60.68 / 37.39, below the ego-only reference of 62.19 / 40.55. These results suggest that simply stacking spatial pose refinement and temporal compensation cannot consistently provide trustworthy collaborator evidence for current-time fusion. Please refer to Sec. 5 for more detailed comparisons.

In summary, our study leads to two observations. *(i)* CoAlign cannot fully correct the relative pose errors between the ego and the delayed neighbor, because its performance is sensitive to the quality of single-agent detections and the correctness of the matched object pairs. *(ii)* TraF-Align can compensate delay once a reasonable spatial reference is available, but it has no explicit mechanism to determine whether the collaborator features remain geometrically trustworthy after pose correction. As a result, residual spatial errors are inherited by the temporal alignment stage and may finally appear as duplicated responses or motion ghosts after propagation.

3.2 Performance Analysis

Low-quality landmarks and mismatch sensitivity in the spatial alignment module. We attribute the instability of CoAlign in realistic scenes to the fact that its pose solver is driven by a limited set of matched detection pairs, whose quality directly depends on the reliability of single-agent detections. In practice, the failure does not only come from the lack of sufficient landmarks. More significantly, the optimizer can be dominated by *bad* landmarks. We observe two sources of such bad constraints. *(i) Low-quality detections*: due to sparse observations, occlusion, and imperfect perception, some detected boxes are themselves inaccurate and therefore provide unreliable geometric references for pose refinement. *(ii) Incorrect correspondences*: even when boxes are detected, the matching stage may associate incompatible objects across agents. In both cases, the optimization is no longer guided by physically reliable constraints, and a few bad landmarks can bias the global solution and drag the refined pose away from the correct one.

Figure 2 presents one representative failure case. In this example, the compensated result is not uniformly poor because the scene itself is unalignable; rather, the main error is caused by one influential mismatched pair, together with other low-quality landmarks that reduce the reliability. Once the wrong match is removed, the global alignment becomes visibly better. The issue is not that box-based pose refinement is always ineffective, but that its reliability is highly sensitive to the quality of detected landmarks and correctness of matched object pairs. The same example also reveals a useful cue for diagnosing such bad correspondences. The wrong pair is not only spatially misleading in the delayed frame, but also exhibits the worst temporal consistency among all matched pairs in the current frame. This suggests that temporal information is useful not only for delay compensation, but also for identifying unreliable correspondences that survive single-frame matching.

Residual spatial errors in temporal delay compensation. TraF-Align can compensate communication delay effectively only when the incoming neighbor feature has already been placed into a reliable coordinate frame. When a biased relative pose is used to warp the delayed neighbor feature into the ego frame, the temporal module starts from a spatially biased input. Its temporal propagation

itself is still designed to recover delayed information, but it does not explicitly determine whether the incoming representation is already contaminated by residual spatial misalignment. As a result, the temporal branch cannot remove such spatial bias on its own: the delay-compensated feature is propagated from a wrong starting point, and the remaining spatial error is further carried into the subsequent feature aggregation stage.

As illustrated in Figure 3, these residual spatial errors may finally appear as duplicated responses and motion ghosts after temporal compensation and fusion. In other words, the main issue is not that temporal compensation becomes meaningless under pose noise, but that it lacks an explicit mechanism to identify and suppress residual spatial errors. This explains why temporal alignment alone may still behave unstably under joint pose noise and communication delay, even though it remains effective in cleaner settings.

The above observations reveal a practical gap in *baseST*. The spatial pose-refinement module can improve the relative pose only when the matched detection pairs are reliable, but it cannot guarantee such reliability in realistic scenes. The delay-compensation module can recover delayed information once the spatial reference is reasonably accurate, but it does not explicitly identify or remove residual spatial errors inherited from the upstream warping stage. Therefore, the final object fusion may simultaneously suffer from imperfect temporal recovery and remaining spatial bias. These limitations motivate a more tightly coupled design, in which spatial pose refinement, temporal propagation, and current-time reliability verification are connected within a unified closed loop.

4 Design

4.1 Problem Formulation

At ego time t, the message received from neighbor j was captured earlier at the delayed time $u_j = t - \tau_j$, where τ_j denotes the communication delay. The relative pose used for collaboration is also noisy; for simplicity, we denote the delayed collaborator pose estimate by $\tilde{\xi}_j^{u_j} = \xi_j^{u_j} \oplus \epsilon_j^{u_j}$. In a direct collaborative pipeline, the delayed collaborator feature $F_j^{u_j}$ would be warped into the ego frame by the noisy relative pose and then fused with the ego feature F_e^t:

$$\mathcal{M}_{j\to e}^{u_j} = \phi_{\text{warp}}\left(F_j^{u_j},\, T(\xi_e^t, \tilde{\xi}_j^{u_j})\right). \tag{1}$$

$T(\cdot)$ denotes the relative-pose transform from the collaborator frame to the ego frame and $\phi_{\text{warp}}(\cdot)$ denotes BEV feature warping. Under joint delay and pose noise, however, $\mathcal{M}_{j\to e}^{u_j}$ is not only spatially biased but temporally stale. Directly fusing it with F_e^t can introduce shifted responses, duplicated structures, or motion ghosts in the final prediction. The goal of *CoAnchor* is to recover a reliable current-time collaborator representation before feature fusion.

4.2 Overall Architecture

CoAnchor is an anchor-centric spatio-temporal alignment framework for collaborative perception under coupled communication delay and pose noise. It constructs a sparse object-level interface, because object states make cross-agent correspondence, motion evolution, and current-time verification much easier to control.

As shown in Fig. 4, this sparse interface connects the whole pipeline. *(i) Spatio-temporal anchor initialization* matches delayed neighbor objects with ego objects around the delayed timestamp, refines an initial relative pose, and converts the reliable matched pairs into delayed anchors. *(ii) Object temporal propagation* advances delayed neighbor objects from u_j to the ego current time t, while allowing anchor states to be corrected when a reliable current-time ego observation is available. *(iii) Closed-loop posterior scoring* uses current-time agreement together with short-history consistency to update pair reliability, and feeds the updated weights back to the next pose-refinement round. *(iv) Anchor-guided feature fusion* uses the final refined pose and propagated object hypotheses to correct collaborator features before downstream dense fusion and detection decoding.

4.3 Spatio-Temporal Anchor Initialization

High-dimensional BEV features are informative for perception, but they do not explicitly expose object correspondences or temporal structure, especially under large delay and pose noise. By contrast, low-dimensional object states naturally carry spatial layout and motion information. We therefore start from object-level states and use them as the sparse carriers of spatio-temporal alignment.

Object state and bipartite matching. In *CoAnchor*'s pipeline, each agent first runs its own single-agent detector to obtain the current object boxes, while the corresponding short box histories are retrieved from memory. For an object m detected by agent i at time u, we represent its motion state as

$$x_{i,m}^u = [p_x,\, p_y,\, v_x,\, v_y,\, \psi]^T, \tag{2}$$

where (p_x, p_y) is the box center in the BEV plane, (v_x, v_y) is the planar velocity estimated from short box histories, and ψ is the heading angle. Given the delayed observations from neighbor j at the delayed timestamp $u_j = t - \tau_j$, we first roughly warp the neighbor detections into the ego frame using the raw relative pose. We then construct cross-agent associations at u_j in two stages. *(i)* We apply a spatial-distance gate to discard obviously incompatible ego–neighbor pairs. *(ii)* On the remaining candidates, we build a bipartite matching cost using velocity similarity together with local neighborhood consistency, and solve a one-to-one assignment by the Hungarian algorithm [11]. This yields the matched-pair set $\mathcal{P}_j = \{(a, b)\}$ between ego objects and delayed neighbor objects.

Trajectory-guided pose refinement. Each matched pair now provides not only a single-frame correspondence but a short motion history. We use these trajectory-aware pairs to refine the relative pose. For each matched pair $k = (a, b) \in \mathcal{P}_j$, we first convert its association cost into an initial reliability score $q_{j,k}^{(0)}$, so that pairs with lower matching cost start with higher confidence. At closed-loop round ℓ, the refined transform $\Delta T_{j\to e}^{(\ell)}$ is obtained by iteratively reweighted least squares (IRLS):

$$\Delta T_{j\to e}^{(\ell)} = \arg\min_{\Delta T\in SE(2)} \sum_{k=(a,b)\in\mathcal{P}_j} q_{j,k}^{(\ell)} \sum_{\kappa=0}^{L_p} \rho\left(\left\|\Delta T\left(p_{j,b}^{u_j-\kappa}\right) - p_{e,a}^{u_j-\kappa}\right\|_2^2\right), \tag{3}$$

where $p_{j,b}^{u_j-\kappa}$ and $p_{e,a}^{u_j-\kappa}$ are the historical box centers of the neighbor object b and ego object a at offset κ, L_p is the history length used in pose refinement, $q_{j,k}^{(\ell)}$ is the current reliability weight of pair k, and $\rho(\cdot)$ is a robust penalty. Intuitively, Eq. (3) searches for a single 2D rigid transform that best aligns the matched short trajectories.

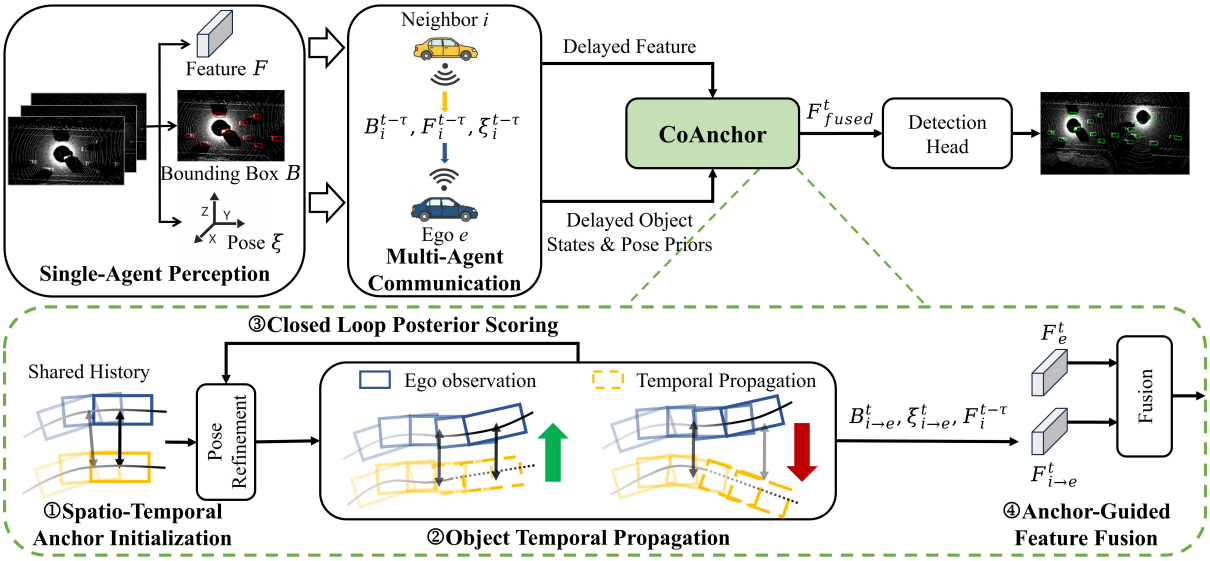


Figure 4: Proposed architecture of *CoAnchor*. Given delayed collaborator features, object cues, and noisy relative poses, *CoAnchor* constructs object-level anchors to refine the relative pose and to provide ego-guided priors for delay compensation. The propagated and verified anchor states are then fed back to later pose refinement and used to support subsequent collaborative fusion.

Anchor construction. After pose refinement, each matched pair is converted into a delayed spatio-temporal anchor. For pair k, we denote its delayed anchor state and covariance by $x_{j,k}^{u_j}$ and $\Sigma_{j,k}^{u_j}$, respectively. The initial pair score $q_{j,k}^{(0)}$, inherited from the association stage, is converted into $\Sigma_{j,k}^{u_j}$, so that low-confidence matches start with larger uncertainty before propagation. These anchors form the matched subset used for current-time correction and closed-loop feedback.

4.4 Object Temporal Propagation

Next, the delayed neighbor scene must be advanced from the delayed timestamp u_j to the current ego time t. We formulate this stage as prediction followed by optional current-time correction. Prediction uses only the delayed anchor state, while correction is applied only when a reliable current-time ego correspondence is available. Each delayed neighbor box is first expressed in the ego coordinate frame at time u_j using the refined relative pose. We then use the ego self-motion estimate to transform these delayed boxes into the ego coordinate frame at the current timestamp t, so that all later propagation, current-time correction, and feedback are performed in a unified current-time ego frame. After ego-time rewriting, we further propagate all delayed neighbor objects with a rule-based motion model, while reserving current-time correction only for the spatio-temporal anchors formed by matched pairs.

For each spatio-temporal anchor, we predict its current-time prior state and covariance as

$$x_{j,k}^{t-} = A(\tau_j) x_{j,k}^{u_j}, \tag{4}$$

$$\Sigma_{j,k}^{t-} = A(\tau_j)\Sigma_{j,k}^{u_j} A(\tau_j)^T + Q(\tau_j), \tag{5}$$

where $x_{j,k}^{u_j}$ and $\Sigma_{j,k}^{u_j}$ denote the delayed anchor state and covariance, $A(\tau_j)$ is the rule-based constant-velocity state-transition matrix over delay τ_j (with position updated by velocity and heading kept unchanged), and $Q(\tau_j)$ is the corresponding diagonal process-noise covariance that increases with delay.

For most propagated anchors, the ego still provides a reliable current-time observation. In such cases, we use the current ego observation of position and heading to perform a standard Kalman-style measurement update [10] on the propagated prior. To determine whether the current-time correction should be accepted, we compute the normalized innovation squared

$$d_{j,k}^2 = (\nu_{j,k}^t)^T (S_{j,k}^t)^{-1} \nu_{j,k}^t, \tag{6}$$

where $\nu_{j,k}^t$ is the measurement residual and $S_{j,k}^t$ is its covariance. A small $d_{j,k}^2$ indicates that the propagated anchor agrees well with the current ego observation. If $d_{j,k}^2$ exceeds a χ^2 gate, or if no reliable current-time ego correspondence is found due to missed detection, occlusion, or association failure, we do not maintain this hypothesis as an active anchor. Instead, it is downgraded to an ordinary propagated neighbor object.

This ordinary-object path also includes delayed neighbor objects that were not matched at initialization. All such objects are still advanced to time t by the same ego-motion rewriting and rule-based motion model, so that the full delayed foreground can be brought into the current ego frame. However, since they are not maintained as spatio-temporal anchors, they do not participate in ego-side correction or later closed-loop feedback. In this way, only the subset that remains reliably supported by cross-time correspondence is preserved as active anchors for subsequent refinement. When no reliable current-time ego correspondence is available, the collaborator object is retained through this ordinary propagation-and-fusion path rather than discarded. After gating, the next pose-refinement round is skipped if fewer than three valid trajectory correspondences remain.

4.5 Closed-Loop Posterior Scoring

A matched pair is considered reliable only if it exhibits consistent spatio-temporal behavior over the available time span, as supported jointly by the shared pre-delay history and the current-time ego observation. We therefore use the propagated anchors to evaluate each pair before the next pose update.

For each matched pair with a reliable current-time ego correspondence, we first measure its current-time consistency by the normalized innovation $d^2_{j,k}$. We then check whether the pair remains stable over a small shared history window around the delayed timestamp, by comparing the pose-transformed neighbor boxes with the corresponding ego boxes across several nearby historical frames. This gives a short-history disagreement term $r^{\text{hist}}_{j,k}$. We then combine the two into a unified feedback residual:

$$r^{\text{fb}}_{j,k} = d^2_{j,k} + r^{\text{hist}}_{j,k}. \tag{7}$$

The next-round pair weight is updated by preserving its initial confidence and exponentially downweighting it according to the feedback residual:

$$q^{(\ell+1)}_{j,k} = q^{(0)}_{j,k} \exp\left(-\lambda_{\text{fb}}\, r^{\text{fb}}_{j,k}\right), \tag{8}$$

where $q^{(0)}_{j,k}$ is the initial score of pair k, and λ_{fb} controls how strongly inconsistent pairs are suppressed. In this way, pairs that remain stable at both the current time and the nearby delayed-time history receive larger weights in the next IRLS pose update, while drifting correspondences are progressively downweighted.

4.6 Anchor-Guided Feature Fusion

After the closed-loop stage, we retain the final refined relative pose together with the propagated current-time object hypotheses. The delayed high-dimensional neighbor feature is first rewritten into the ego current frame using the refined pose:

$$\mathcal{M}^{u_j}_{j\to e} = \phi_{\text{warp}}\left(F^{u_j}_j, T(\xi^t_e, \tilde{\xi}^{u_j}_j) \circ \Delta T^{(\ell_{\max})}_{j\to e}\right), \tag{9}$$

where $\Delta T^{(\ell_{\max})}_{j\to e}$ is the final refined pose after the last loop round. Specifically, the delayed collaborator feature is first transformed into the ego coordinate system at the delayed time, and then brought into the ego frame at the current time. We apply a non-learnable box-wise feature mover: for each delayed neighbor box already rewritten into the current ego frame, it extracts the enclosing axis-aligned BEV rectangle and relocates it toward the corresponding propagated current-time box by bilinear sampling. This deterministic operation corrects the dominant foreground displacement before dense fusion, while leaving the rest of the BEV feature unchanged.

Downstream fusion and detection decoding. After pose correction and box-wise foreground adjustment, the neighbor features are aggregated with the ego feature by a downstream collaborative perception module. The fusion operator here is flexible and can be instantiated by any standard cooperative perception module. In our implementation, we adopt multi-scale feature fusion, since it achieves a good trade-off among computational cost, optimization stability, and multi-scale representation capability, thus providing robust detection performance in practice. The fused feature maps are then decoded into final detection outputs. Concretely, the decoder produces a regression output and a classification output for candidate boxes. The regression output describes the object geometry, including object position, box size, and heading angle, while the classification output estimates the confidence that each candidate belongs to a foreground vehicle rather than background. These predictions are finally decoded to obtain the detection results.

5 Evaluation

5.1 Experimental Setup

Datasets. We evaluate the proposed method on two widely used collaborative perception benchmarks, OPV2V [30] and V2V4Real [28]. OPV2V is a large-scale simulated benchmark for vehicle-to-vehicle collaboration, while V2V4Real is a real-world benchmark collected in realistic driving scenes.

Compared methods. We compare our method with representative baselines from both pose-robust and asynchronous collaborative perception, including V2X-ViT [29], ERMVP [36], CoAlign [15], TraF-Align [19], and CoST [20]. To verify whether spatial refinement and delayed fusion can be coordinated by simple composition, we further construct the cascaded baseline *baseST*. We also report ego-only references for TraF-Align and our method to reveal how much collaborative gain remains under heavy perturbations.

Implementation details. For a fair comparison, all methods adopt PointPillars [12] as the common BEV backbone. Following standard collaborative perception settings, the voxel size is set to 0.4 m and the communication range is limited to 70 m. We adopt a staged training pipeline. We first train a single-agent detector to produce per-agent object detections. During collaborative training, these detections are used as the object-level inputs of CoAnchor, and the whole collaborative perception model is trained under injected pose noise together with the downstream collaborative detection branch. The collaborative detector is supervised by the standard classification, box-regression, and direction losses used in PointPillars-style heads. All models are trained within the same codebase on two NVIDIA RTX 4090 GPUs. We follow the official optimization settings of each baseline whenever available.

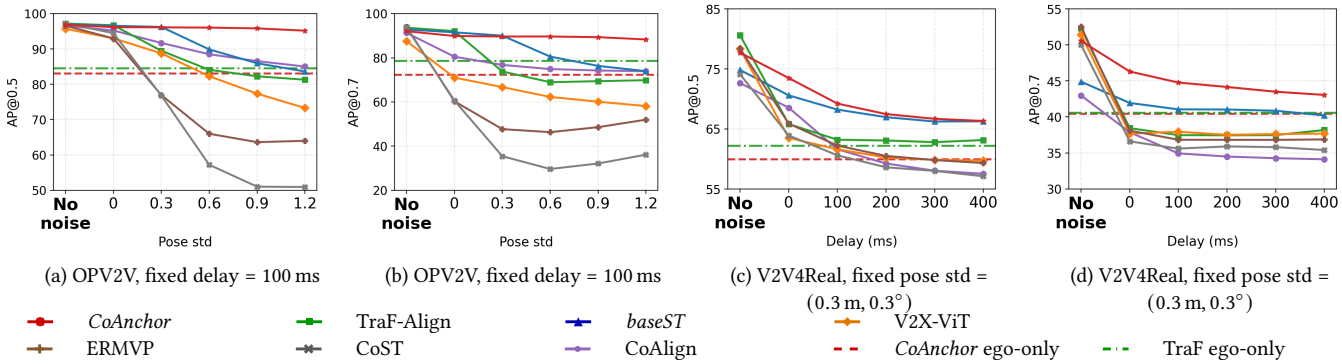
Evaluation protocol. We report detection performance using AP@0.5 and AP@0.7. To evaluate robustness under coupled perturbations, we inject Gaussian pose noise into each agent pose in the BEV plane and impose communication delay on the non-ego branch by delaying collaborator observations. Specifically, for a pose $\xi = (x, y, \theta)$, the noisy pose is constructed by independently adding noise with standard deviation σ_t to x and y, and noise with standard deviation σ_r to θ. Unless otherwise specified, pose noise is written as (σ_t, σ_r), where σ_t is measured in meters and σ_r in degrees. Our main comparison is conducted under four representative settings: *(i) No Noise*, with no injected pose perturbation or communication delay; *(ii) Joint-Hard*, with 200 ms delay and pose noise (0.6 m, 0.6°); *(iii) Pose-Hard*, with 100 ms delay and pose noise (0.9 m, 0.9°); and *(iv) Delay-Hard*, with 300 ms delay and pose noise (0.3 m, 0.3°). Unless otherwise specified, the same fixed rule-based settings are used across datasets and test conditions.

5.2 Quantitative Comparison

Main results. Table 1 reports the comparison with state-of-the-art methods. Overall, *CoAnchor* is not always the best under the clean setting, but it remains competitive and achieves the strongest

Table 1: Comparison with state-of-the-art methods on OPV2V and V2V4Real. Each entry is reported as AP@0.5 / AP@0.7.

Method	OPV2V [30]				V2V4Real [28]			
	No Noise	P-Hard	J-Hard	D-Hard	No Noise	P-Hard	J-Hard	D-Hard
CoAlign [15]	96.64 / 91.31	86.49 / 74.19	85.90 / 72.82	83.19 / 71.37	72.64 / 42.97	51.34 / 32.81	53.59 / 33.22	58.26 / 34.26
ERMVP [36]	96.59 / **94.01**	63.61 / 48.57	57.46 / 42.95	47.86 / 36.56	78.38 / **52.55**	51.63 / 34.69	53.16 / 34.55	59.81 / 36.81
V2X-ViT [29]	95.71 / 87.54	77.34 / 60.06	75.90 / 55.94	68.55 / 52.28	78.33 / 51.45	51.87 / 35.37	53.83 / 36.24	59.84 / 37.62
CoST [20]	97.20 / 93.70	51.00 / 32.10	47.90 / 27.80	42.60 / 26.90	74.10 / 50.00	49.50 / 30.60	51.80 / 32.00	58.00 / 35.80
TraF-Align (ego-only)	84.49 / 78.57	84.49 / 78.57	84.49 / 78.57	84.49 / 78.57	62.19 / 40.55	62.19 / 40.55	62.19 / 40.55	62.19 / 40.55
TraF-Align [19]	**97.22** / 93.66	82.27 / 69.43	85.87 / 72.46	90.33 / 78.58	**80.60** / 52.29	55.33 / 35.30	58.18 / 35.09	62.79 / 37.47
baseST [15, 19]	97.04 / 92.90	85.97 / 76.34	90.19 / 81.21	93.76 / 86.03	74.82 / 44.89	56.30 / 36.15	61.04 / 37.69	66.23 / 40.86
CoAnchor (ego-only)	83.09 / 72.35	83.09 / 72.35	83.09 / 72.35	83.09 / 72.35	59.93 / 40.43	59.93 / 40.43	59.93 / 40.43	59.93 / 40.43
CoAnchor	96.89 / 92.08	**95.85 / 89.36**	**95.15 / 87.96**	**94.44 / 86.40**	77.71 / 50.56	**67.76 / 45.01**	**67.04 / 44.05**	**66.68 / 43.50**



(a) OPV2V, fixed delay = 100 ms (b) OPV2V, fixed delay = 100 ms (c) V2V4Real, fixed pose std = $(0.3\,\text{m}, 0.3^\circ)$ (d) V2V4Real, fixed pose std = $(0.3\,\text{m}, 0.3^\circ)$

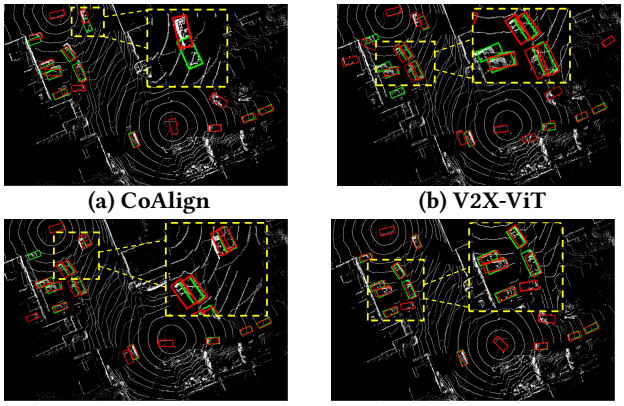
Figure 5: Robustness analysis under progressively increasing temporal and spatial perturbations. Panels (a)–(b) fix the delay at 100 ms on OPV2V and vary the pose-noise magnitude. Panels (c)–(d) fix the pose-noise standard deviations at $(0.3\,\text{m}, 0.3^\circ)$ on V2V4Real and vary the communication delay. AP@0.5 and AP@0.7 are reported separately.

robustness once pose noise and communication delay are jointly introduced. This trend is consistent across both OPV2V and V2V4Real, indicating that the proposed *CoAnchor* preserves standard-case accuracy while substantially improving robustness under coupled perturbations. *(i)* On OPV2V, our method reaches 95.15 / 87.96 under *Joint-Hard*, clearly outperforming the strongest baseline *baseST* at 90.19 / 81.21. The advantage is already visible in the more pose-dominant mixed setting *Pose-Hard*, while under *Delay-Hard* our method still remains slightly above *baseST*. *(ii)* On V2V4Real, where the collaborative pipeline is more strongly affected by realistic detection noise and unstable cross-agent correspondences, the robustness gain becomes even more pronounced. Under *Joint-Hard*, *CoAnchor* reaches 67.04 / 44.05, clearly surpassing *baseST* at 61.04 / 37.69. Under *Pose-Hard*, the margin is even larger, and under *Delay-Hard* our method still maintains an advantage, especially on AP@0.7.

Moreover, four observations are worth highlighting. *(i)* Data-driven fusion methods such as V2X-ViT remain competitive in the clean setting, but their performance drops sharply once the perturbation becomes large, especially on the harder mixed and joint settings. This suggests that implicit feature-level fitting has limited extrapolation ability when the test-time noise exceeds the regime that can be absorbed by the learned alignment. *(ii)* Spatially oriented methods such as CoAlign can benefit from explicit pose handling, but their gains shrink once delayed observations also need to be propagated to the current ego time. This is particularly clear on V2V4Real, where their clean-case accuracy is already noticeably below strong asynchronous baselines. *(iii)* The delay-oriented method TraF-Align remains very strong when the cross-agent spatial reference is reliable, but its robustness becomes sensitive to pose corruption. In both datasets, its clean or mildly perturbed performance is high, whereas the gap to our method enlarges in the harder coupled settings. *(iv)* The cascaded baseline *baseST* is indeed stronger than single-purpose modules in several noisy regimes, which confirms that simple composition is useful but insufficient: it still leaves a clear gap to *CoAnchor* under *Joint-Hard*. This behavior matches our design goal: instead of only stacking spatial and temporal corrections, we explicitly verify which propagated object hypotheses remain trustworthy at the current ego time and feed this reliability back to later pose refinement.

Robustness to increasing pose noise. We fix the delay and gradually increase pose noise on OPV2V, as shown in Figure 5 (a) and (b). At low noise, *baseST* remains competitive because temporal compensation is effective while the upstream spatial reference is still reliable. As pose noise grows, V2X-ViT and *baseST* degrade more rapidly: learned feature interaction has limited tolerance to large perturbations, while temporal propagation alone cannot determine whether incoming collaborator evidence is already spatially biased. In contrast, *CoAnchor* combines short-history pose refinement with current-time anchor verification, preserving valid hypotheses while suppressing spatially biased ones and therefore degrading more gracefully across the tested noise range.

Robustness to increasing delay. We fix the pose noise and vary the communication delay on V2V4Real, as shown in Figure 5 (c) and (d). TraF-Align drops sharply once the collaborator input is spatially biased, indicating that temporal compensation starts from an unreliable spatial reference under coupled perturbations. *baseST* degrades more slowly as the delay increases because upstream pose correction provides partial mitigation, but it starts from a noticeably lower level, consistent with negative pose optimization under realistic detection noise. By re-evaluating propagated anchors



(a) CoAlign (b) V2X-ViT (c) *baseST* (d) *CoAnchor*

Figure 6: Qualitative analysis under the *Joint-Hard* setting.

with current-time ego observations before later pose refinement and fusion, *CoAnchor* retains valid collaborator hypotheses while filtering stale or spatially biased ones, maintaining the strongest and most stable collaborative advantage across the tested delay range.

5.3 Qualitative Comparison

Figure 6 shows representative qualitative results under coupled pose noise and communication delay. *baseST* tends to produce duplicate responses together with missed detections when delayed collaborator evidence is not properly verified before fusion. CoAlign alleviates part of the spatial misalignment, but under asynchronous inputs, it can still leave delayed boxes caused by the temporal gap, and some collaborator boxes remain spatially shifted after pose correction. V2X-ViT exhibits another typical failure pattern, where missed detections and duplicate responses coexist once the learned fusion is affected by stronger spatio-temporal perturbation.

By comparison, *CoAnchor* produces cleaner final detections with fewer duplicates and more accurate box locations. During object propagation, ego-guided correction helps merge propagated co-visible hypotheses with the current ego observations and suppress duplicate ghost responses before final fusion. The closed-loop pose feedback further removes unreliable matched pairs in later pose refinement rounds, reducing the chance that spatially biased or temporally stale collaborator evidence survives into the final fused detections. Overall, the qualitative behavior is consistent with the quantitative results: the proposed framework better preserves useful collaborator information while suppressing duplicated or shifted responses under coupled perturbations.

5.4 Ablation Study

Module-wise ablation. Table 2 evaluates three anchor-level components on OPV2V under the Joint-Hard setting. Specifically, w/o historical initialization cues removes short-history information during anchor initialization; w/o ego-guided correction prevents propagated anchors from being corrected by reliable current-time ego observations; and w/o anchor feedback disables current-time reliability updates before later pose-refinement rounds.

Removing any component degrades performance. Removing historical initialization cues causes the largest drop, from 95.15/87.96 to 92.47/81.59. Removing ego-guided correction yields 94.03/85.27, while removing anchor feedback gives 94.31/86.73. These results show that short-history cues establish reliable initial anchors, while current-time ego correction and anchor feedback provide complementary gains beyond a feed-forward anchor pipeline.

Table 2: Module-wise ablation study on OPV2V.

Variant	OPV2V	
	AP@0.5	AP@0.7
w/o historical initialization cues	92.47	81.59
w/o ego-guided correction	94.03	85.27
w/o anchor feedback	94.31	86.73
***CoAnchor* (Full)**	**95.15**	**87.96**

Table 3: Performance and runtime analysis on V2V4Real.

Method	V2V4Real		Time (ms/frame)
	AP@0.5	AP@0.7	
V2X-ViT [29]	53.83	36.24	110.62
TraF-Align [19]	58.18	35.09	121.73
CoAnchor w/o feedback	66.19	43.35	49.39
CoAnchor +1 round feedback	**67.04**	44.05	51.62
CoAnchor +2 round feedback	66.96	**44.16**	52.51

Efficiency and runtime analysis. We further compare detection accuracy and runtime on V2V4Real under the *Joint-Hard* setting in Table 3. We include the external baselines V2X-ViT and TraF-Align, together with three versions of our anchor-centric pipeline: *CoAnchor* w/o feedback, *CoAnchor* +1 round feedback, and *CoAnchor* +2 round feedback. Because the closed loop operates mainly on sparse anchors rather than repeatedly invoking dense fusion modules, additional feedback incurs only limited computational overhead. Compared with *CoAnchor* w/o feedback, adding one feedback round improves accuracy with only a small runtime increase, indicating that a single anchor-feedback step already brings the main practical gain. Adding a second round yields only marginal change, so most of the benefit is already obtained in the first round. It is also worth noting that our method remains substantially more efficient than the compared baselines. Even with one feedback round, it is still much faster than V2X-ViT and TraF-Align while achieving the best detection accuracy in this setting. Therefore, one feedback round offers the most favorable accuracy–efficiency trade-off and is adopted as the default setting.

6 CONCLUSION AND FUTURE WORK

In this paper, we study asynchronous collaborative perception under coupled communication delay and relative-pose noise. We propose *CoAnchor*, an anchor-centric closed-loop spatio-temporal alignment framework that uses sparse object-level anchors to connect short-history-aware initialization, ego-guided propagation, current-time feedback, and pose-corrected feature fusion. Experiments show that *CoAnchor* remains competitive under clean settings while improving robustness under coupled perturbations. Although the adopted constant-velocity motion model provides a lightweight approximation for short communication delays, its modeling capacity may be limited under longer latency or highly nonlinear object motion. Future work will explore stronger delay-aware motion models and improve standard-case accuracy while preserving the anchor-centric design's efficiency and robustness.

Acknowledgments

We appreciate the insightful feedback from the anonymous reviewers who helped improve this work. This work was supported by the NSFC Project (62402058), the Foundation of State Key Laboratory of Networking and Switching Technology (NST20260111), and the Fundamental Research Funds for the Central Universities.

References

[1] Antoine Caillot, Safa Ouerghi, Pascal Vasseur, Rémi Boutteau, and Yohan Dupuis. 2022. Survey on Cooperative Perception in an Automotive Context. *IEEE Transactions on Intelligent Transportation Systems* 23, 9 (2022), 14204–14223. doi:10.1109/TITS.2022.3153815

[2] Gong Chen, Chaokun Zhang, Pengcheng Lv, and Xiaohui Xie. 2026. CoRA: A Collaborative Robust Architecture with Hybrid Fusion for Efficient Perception. *Proceedings of the AAAI Conference on Artificial Intelligence* 40, 4 (March 2026), 2841–2849. doi:10.1609/aaai.v40i4.37274

[3] Qi Chen, Sihai Tang, Qing Yang, and Song Fu. 2019. Cooper: Cooperative Perception for Connected Autonomous Vehicles based on 3D Point Clouds. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (Dallas, TX, USA). IEEE, Piscataway, NJ, USA, 514–524. doi:10.1109/ICDCS.2019.00058

[4] Xin Gao, Xinyu Zhang, Yiguo Lu, Yuning Huang, Lei Yang, Yijin Xiong, and Peng Liu. 2024. A Survey of Collaborative Perception in Intelligent Vehicles at Intersections. *IEEE Transactions on Intelligent Vehicles* (2024), 1–20. doi:10.1109/TIV.2024.3395783

[5] Jiaming Gu, Jingyu Zhang, Muyang Zhang, Weiliang Meng, Shibiao Xu, Jiguang Zhang, and Xiaopeng Zhang. 2023. FeaCo: Reaching Robust Feature-Level Consensus in Noisy Pose Conditions. In *Proceedings of the 31st ACM International Conference on Multimedia* (Ottawa, ON, Canada) *(MM '23)*. Association for Computing Machinery, New York, NY, USA, 3628–3636. doi:10.1145/3581783.3611880

[6] Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. 2022. Where2comm: Communication-Efficient Collaborative Perception via Spatial Confidence Maps. In *Advances in Neural Information Processing Systems* (New Orleans, LA, USA), Vol. 35. Curran Associates, Inc., Red Hook, NY, USA, 4874–4886. doi:10.52202/068431-0352

[7] Tao Huang, Jianan Liu, Xi Zhou, Dinh C. Nguyen, Mostafa Rahimi Azghadi, Yuxuan Xia, Qing-Long Han, and Sumei Sun. 2025. Vehicle-to-Everything Cooperative Perception for Autonomous Driving. *Proc. IEEE* 113, 5 (May 2025), 443–477. doi:10.1109/JPROC.2025.3600903

[8] Zhe Huang, Shuo Wang, Yongcai Wang, Wanting Li, Deying Li, and Lei Wang. 2024. RoCo: Robust Cooperative Perception By Iterative Object Matching and Pose Adjustment. In *Proceedings of the 32nd ACM International Conference on Multimedia* (Melbourne, VIC, Australia) *(MM '24)*. Association for Computing Machinery, New York, NY, USA, 7833–7842. doi:10.1145/3664647.3680559

[9] Zhe Huang, Shuo Wang, Yongcai Wang, and Lei Wang. 2025. CoDiff: Conditional Diffusion Model for Collaborative 3D Object Detection. arXiv:2502.14891 [cs.CV] https://arxiv.org/abs/2502.14891

[10] Rudolph Emil Kalman. 1960. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering* 82, 1 (1960), 35–45. doi:10.1115/1.3662552

[11] H. W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 1-2 (1955), 83–97. doi:10.1002/nav.3800020109

[12] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. 2019. PointPillars: Fast Encoders for Object Detection from Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA, USA). IEEE, Piscataway, NJ, USA, 12697–12705. doi:10.1109/CVPR.2019.01298

[13] Zixing Lei, Zhenyang Ni, Ruize Han, Shuo Tang, Dingju Wang, Chen Feng, Siheng Chen, and Yanfeng Wang. 2024. Robust Collaborative Perception without External Localization and Clock Devices. In *2024 IEEE International Conference on Robotics and Automation (ICRA)* (Yokohama, Japan). IEEE, Piscataway, NJ, USA, 7280–7286. doi:10.1109/ICRA57147.2024.10610635

[14] Zixing Lei, Shunli Ren, Yue Hu, Wenjun Zhang, and Siheng Chen. 2022. Latency-Aware Collaborative Perception. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII* (Tel Aviv, Israel) *(Lecture Notes in Computer Science, Vol. 13692)*. Springer, Cham, Switzerland, 316–332. doi:10.1007/978-3-031-19824-3_19

[15] Yifan Lu, Quanhao Li, Baoan Liu, Mehrdad Dianati, Chen Feng, Siheng Chen, and Yanfeng Wang. 2023. Robust collaborative 3D object detection in presence of pose errors. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* (London, United Kingdom). IEEE, Piscataway, NJ, USA, 4812–4818. doi:10.1109/ICRA48891.2023.10160546

[16] Zhenyang Ni, Zixing Lei, Yifan Lu, Dingju Wang, Chen Feng, Yanfeng Wang, and Siheng Chen. 2024. Self-Localized Collaborative Perception. arXiv:2406.12712 [cs.CV] doi:10.48550/arXiv.2406.12712

[17] Andreas Rauch, Felix Klanner, Ralph Rasshofer, and Klaus Dietmayer. 2012. Car2X-based perception in a high-level fusion architecture for cooperative perception systems. In *2012 IEEE Intelligent Vehicles Symposium* (Alcalá de Henares, Spain). IEEE, Piscataway, NJ, USA, 270–275. doi:10.1109/IVS.2012.6232130

[18] Zaydoun Yahya Rawashdeh and Zheng Wang. 2018. Collaborative Automated Driving: A Machine Learning-based Method to Enhance the Accuracy of Shared Information. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (Maui, HI, USA). IEEE, Piscataway, NJ, USA, 3961–3966. doi:10.1109/ITSC.2018.8569832

[19] Zhiying Song, Lei Yang, Fuxi Wen, and Jun Li. 2025. TraF-Align: Trajectory-aware feature alignment for asynchronous multi-agent perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Nashville, TN, USA). IEEE, Piscataway, NJ, USA, 12048–12057. doi:10.1109/CVPR52734.2025.01125

[20] Zongheng Tang, Yi Liu, Yifan Sun, Yulu Gao, Jinyu Chen, Runsheng Xu, and Si Liu. 2025. CoST: Efficient Collaborative Perception From Unified Spatiotemporal Perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Honolulu, HI, USA). IEEE, Piscataway, NJ, USA, 1120–1129. https://openaccess.thecvf.com/content/ICCV2025/html/Tang_CoST_Efficient_Collaborative_Perception_From_Unified_Spatiotemporal_Perspective_ICCV_2025_paper.html

[21] Nicholas Vadivelu, Mengye Ren, James Tu, Jingkang Wang, and Raquel Urtasun. 2021. Learning to Communicate and Correct Pose Errors. In *Proceedings of the 2020 Conference on Robot Learning* (Virtual Conference) *(Proceedings of Machine Learning Research, Vol. 155)*, Jens Kober, Fabio Ramos, and Claire Tomlin (Eds.). PMLR, Cambridge, MA, USA, 1195–1210. https://proceedings.mlr.press/v155/vadivelu21a.html

[22] Lei Wan, Jianxin Zhao, Andreas Wiedholz, Manuel Bied, Mateus Martinez de Lucena, Abhishek Dinkar Jagtap, Andreas Festag, Antônio Augusto Fröhlich, Hannan Ejaz Keen, and Alexey Vinel. 2026. A Systematic Literature Review on Vehicular Collaborative Perception—A Computer Vision Perspective. *IEEE Transactions on Intelligent Transportation Systems* 27, 1 (Jan. 2026), 81–118. doi:10.1109/TITS.2025.3631141

[23] Junjie Wang and Tomas Nordström. 2025. Latency Robust Cooperative Perception Using Asynchronous Feature Fusion. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (Tucson, AZ, USA). IEEE, Piscataway, NJ, USA, 4862–4871. doi:10.1109/WACV61041.2025.00476

[24] Sichao Wang, Ming Yuan, Chuang Zhang, Qing Xu, Lei He, and Jianqiang Wang. 2025. V2X-DGPE: Addressing Domain Gaps and Pose Errors for Robust Collaborative 3D Object Detection. In *2025 IEEE Intelligent Vehicles Symposium (IV)* (Cluj-Napoca, Romania). IEEE, Piscataway, NJ, USA, 2074–2080. doi:10.1109/IV64158.2025.11097385

[25] Sizhe Wei, Yuxi Wei, Yue Hu, Yifan Lu, Yiqi Zhong, Siheng Chen, and Ya Zhang. 2023. Asynchrony-Robust Collaborative Perception via Bird's Eye View Flow. In *Advances in Neural Information Processing Systems* (New Orleans, LA, USA), Vol. 36. Curran Associates, Inc., Red Hook, NY, USA, 28462–28477. doi:10.52202/075280-1236

[26] Dongzhu Xu, Rui Lin, Huanhuan Zhang, Anfu Zhou, and Huadong Ma. 2025. Bridging Cross-Layer Interactions Between 5G RAN and MEC for Latency-Critical Video Analytics. *IEEE Transactions on Networking* 33, 4 (2025), 1614–1629. doi:10.1109/TON.2025.3542456

[27] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. 2023. CoBEVT: Cooperative Bird's Eye View Semantic Segmentation with Sparse Transformers. In *Proceedings of the 6th Conference on Robot Learning* (Auckland, New Zealand) *(Proceedings of Machine Learning Research, Vol. 205)*, Karen Liu, Dana Kulic, and Jeff Ichnowski (Eds.). PMLR, Cambridge, MA, USA, 989–1000. https://proceedings.mlr.press/v205/xu23a.html

[28] 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. 2023. 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)* (Vancouver, BC, Canada). IEEE, Piscataway, NJ, USA, 13712–13722. doi:10.1109/CVPR52729.2023.01318

[29] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. 2022. V2X-ViT: Vehicle-to-Everything Cooperative Perception with Vision Transformer. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX* (Tel Aviv, Israel) *(Lecture Notes in Computer Science, Vol. 13699)*. Springer, Cham, Switzerland, 107–124. doi:10.1007/978-3-031-19842-7_7

[30] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. 2022. OPV2V: An Open Benchmark Dataset and Fusion Pipeline for Perception with Vehicle-to-Vehicle Communication. In *2022 International Conference on Robotics and Automation (ICRA)* (Philadelphia, PA, USA). IEEE, Piscataway, NJ, USA, 2583–2589. doi:10.1109/ICRA46639.2022.9812038

[31] Yunjiang Xu, Lingzhi Li, Jin Wang, Benyuan Yang, Zhiwen Wu, Xinhong Chen, and Jianping Wang. 2025. CoDynTrust: Robust Asynchronous Collaborative Perception via Dynamic Feature Trust Modulus. In *2025 IEEE International Conference on Robotics and Automation (ICRA)* (Atlanta, GA, USA). IEEE, Piscataway,

NJ, USA, 336–342. doi:10.1109/ICRA55743.2025.11127779

[32] Kun Yang, Dingkang Yang, Jingyu Zhang, Mingcheng Li, Yang Liu, Jing Liu, Hanqi Wang, Peng Sun, and Liang Song. 2023. Spatio-Temporal Domain Awareness for Multi-Agent Collaborative Perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Paris, France). IEEE, Piscataway, NJ, USA, 23383–23392. doi:10.1109/ICCV51070.2023.02137

[33] Haibao Yu, Yingjuan Tang, Enze Xie, Jilei Mao, Ping Luo, and Zaiqing Nie. 2023. Flow-based feature fusion for vehicle-infrastructure cooperative 3D object detection. In *Advances in Neural Information Processing Systems* (New Orleans, LA, USA), Vol. 36. Curran Associates, Inc., Red Hook, NY, USA, 34493–34503. doi:10.52202/075280-1497

[34] Yunshuang Yuan, Hao Cheng, and Monika Sester. 2022. Keypoints-Based Deep Feature Fusion for Cooperative Vehicle Detection of Autonomous Driving. *IEEE Robotics and Automation Letters* 7, 2 (2022), 3054–3061. doi:10.1109/LRA.2022.3143299

[35] Yunshuang Yuan, Yan Xia, Daniel Cremers, and Monika Sester. 2025. SparseAlign: A Fully Sparse Framework for Cooperative Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Nashville, TN, USA). IEEE, Piscataway, NJ, USA, 22296–22305. doi:10.1109/CVPR52734.2025.02077

[36] Jingyu Zhang, Kun Yang, Yilei Wang, Hanqi Wang, Peng Sun, and Liang Song. 2024. ERMVP: Communication-Efficient and Collaboration-Robust Multi-Vehicle Perception in Challenging Environments. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA, USA). IEEE, Piscataway, NJ, USA, 12575–12584. doi:10.1109/CVPR52733.2024.01195

[37] Zewei Zhou, Hao Xiang, Zhaoliang Zheng, Seth Z. Zhao, Mingyue Lei, Yun Zhang, Tianhui Cai, Xinyi Liu, Johnson Liu, Maheswari Bajji, Xin Xia, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. 2025. V2XPnP: Vehicle-to-Everything Spatio-Temporal Fusion for Multi-Agent Perception and Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Honolulu, HI, USA). IEEE, Piscataway, NJ, USA, 25399–25409. doi:10.1109/ICCV51701.2025.02356

A Evaluation Protocol and Baseline Construction

A.1 AP Computation

We report AP@0.5 and AP@0.7 using dataset-level global sorting. Concretely, after decoding and non-maximum suppression, all predicted boxes over the entire evaluation split are pooled together and sorted by confidence to form a single precision–recall curve at each IoU threshold. In other words, the ranking is established once over the whole benchmark, rather than separately within each scene. This evaluation setup is also consistent with prior methods such as CoAlign [15] and RoCo [8].

By contrast, some implementations use scene-wise sorting with scene-level aggregation: predictions are first ranked within each scene, and true/false positives are accumulated locally before the results are aggregated scene by scene. Accordingly, the absolute AP values under the two protocols can differ numerically. All results in the main text and this appendix follow the same dataset-level global-sorting protocol.

A.2 Construction of the Cascaded Baseline *baseST*

We clarify how the cascaded baseline *baseST* is constructed from CoAlign and TraF-Align. Starting from the original TraF-Align pipeline, we insert a CoAlign-style pose-refinement stage before temporal delay compensation. To preserve the co-visibility assumption underlying box-based cross-agent matching, the relative pose is refined only from the delayed neighbor observation and the ego observation at the same delayed timestamp $u_j = t - \tau_j$. In other words, detections from different timestamps are not jointly fed into the CoAlign solver.

After this delayed-time pose refinement, the refined relative pose is used consistently for the delayed collaborator message and for the historical features required by TraF-Align. Specifically, we transform the entire short history into the ego frame using the same refined delayed-time pose together with ego-motion, rather than optimizing each historical frame independently. This design avoids introducing additional frame-wise temporal inconsistency from the inserted spatial-alignment stage, and keeps the cascaded implementation as faithful as possible to the original temporal-compensation pipeline of TraF-Align.

B Implementation and Reproducibility Details

This appendix section provides implementation details beyond the main text, including the detector architecture, method-specific training configurations, runtime pipeline of the object-level branch, and the full set of rule-based hyperparameters. We first summarize the detector instantiations and optimization settings used in our experiments. We then describe the inference-time flow of the object-level branch and present the corresponding explicit formulation.

B.1 Method-specific Training Configurations

Table 4 summarizes the optimization schedules and training-time perturbation settings used by the compared methods. All methods use AdamW with weight decay 10^{-4}, and all experiments share the same standard point-cloud augmentation, including scaling, rotation, and flipping.

Although all methods are implemented within the same PointPillar family, their detector-side configurations and training schedules are not strictly identical. The single-agent detector adopts a PointPillar-style architecture. Voxelized LiDAR points are first encoded by a 64-channel PillarVFE, scattered to the BEV plane, and processed by a three-stage ResNet-style BEV backbone. The multi-scale features are then decoded by an upsampling path, compressed by a 256-channel shrink head, and passed to the standard classification, box-regression, and direction-classification heads. In our implementation, the single-agent detector uses the same backbone widths as the collaborative detector (64–128–256), so that the comparison isolates the effect of collaboration rather than changes in the detector trunk.

The collaborative detector shares the same voxel encoder and BEV backbone, but introduces attention-based intermediate fusion at the three BEV scales. Specifically, per-agent BEV features with channel dimensions 64, 128, and 256 are transformed into the ego frame, fused scale by scale via self-attention, decoded into a shared BEV representation, compressed to 256 channels, and finally fed to the same detection heads. The detector architecture is kept unchanged across datasets.

For our single-agent detector and our collaborative model, the detection loss follows the standard PointPillar formulation used in the codebase. Specifically, the positive classification weight is set to 1.0. The classification branch uses Sigmoid Focal Loss with $\alpha = 0.25$, $\gamma = 2.0$, and loss weight 1.0. The box-regression branch uses codewise Weighted Smooth L1 Loss with $\sigma = 3.0$ and loss weight 2.0. The direction branch uses Weighted Softmax Classification Loss with loss weight 0.2. The baselines follow the official loss definitions in their released implementations.

B.2 Formulation in Practice

The learnable detector/fusion backbone remains unchanged; the proposed object-level module is an inference-time rule-based branch described below.

Matching cost. Let Ω_{ab} be the set of common history offsets shared by ego track a and delayed collaborator track b, after rotating the collaborator velocity into the ego frame using the initial pose rotation R_0. The velocity consistency term is defined as

$$d_{ab}^{\text{vel}} = \frac{1}{|\Omega_{ab}|} \sum_{\kappa \in \Omega_{ab}} \left\| v_{a,\kappa}^{e} - R_0 v_{b,\kappa} \right\|_2 . \tag{10}$$

For local-graph consistency, the implementation constructs two signatures from the K nearest neighbors of each track center: an edge-distance signature and a pairwise-neighbor signature. Let $D_{\text{sort}}(\cdot,\cdot)$ denote the mean absolute difference between the sorted values of two signatures truncated to their common length. The graph term is

$$d_{ab}^{\text{graph}} = \frac{w_{\text{edge}} D_{\text{sort}}(s_a^{\text{edge}}, s_b^{\text{edge}}) + w_{\text{pair}} D_{\text{sort}}(s_a^{\text{pair}}, s_b^{\text{pair}})}{w_{\text{edge}} + w_{\text{pair}}}. \tag{11}$$

The final matching cost before Hungarian assignment is

$$c_{ab} = \lambda_{\text{vel}} \frac{d_{ab}^{\text{vel}}}{T_{\text{vel}}} + \lambda_{\text{graph}} \frac{d_{ab}^{\text{graph}}}{T_{\text{graph}}}. \tag{12}$$

Table 4: Method-specific training configurations.

Method	Epochs	LR / scheduler	Perturbation during training
CoAnchor (single-agent)	30	Multi-step decay; initial lr = 0.002 $\gamma = 0.1$, milestones [10, 20]	perfect setting throughout
CoAnchor CoAlign	30	Multi-step decay; initial lr = 0.002 $\gamma = 0.1$, milestones [10, 20]	pose noise $(0.2\,\text{m}, 0.2^\circ)$
V2X-ViT	60 clean + 10 fine-tune	Multi-step decay; initial lr = 0.001 $\gamma = 0.1$, milestones [15, 50]	Clean pretraining; fine-tune with pose noise $(0.2\,\text{m}, 0.2^\circ)$ and random delay 200–300 ms
TraF-Align *baseST*	60	One-cycle schedule; peak lr = 10^{-4} initial lr = 10^{-5}; ramp-up fraction = 0.4	random delay 0–400 ms
ERMVP CoST	60	Cosine annealing with warmup warmup lr = 2×10^{-4} warmup epochs = 10 minimum lr = 2×10^{-5}	perfect setting throughout

Cost-to-pair-weight mapping. Pose refinement is executed only when the matched set for neighbor j contains at least $N_{\text{trk}}^{\min} = 3$ trajectory correspondences, i.e., at least three matched trajectories are available. For a matched pair $k = (a, b)$ in neighbor j's matched set P_j, the implementation assigns the initial pair weight based on the matching cost c_{ab}

$$q_{j,k}^{(0)} = \exp\left(-\frac{c_{ab}}{T_q}\right), \tag{13}$$

which is used both for Kalman-prior initialization and as the pair weight entering pose refinement.

IRLS pose solver. Both the initial pose estimation and the later pose-refinement rounds driven by Closed-Loop Posterior Scoring use weighted rigid alignment with IRLS. Following Eq. (3) in the main text, at loop round ℓ the refined pose is estimated from the matched trajectory points by

$$\Delta T_{j\to e}^{(\ell)} = \arg\min_{\Delta T \in SE(2)} \sum_{k\in P_j} q_{j,k}^{(\ell)} \sum_{\kappa=0}^{L_p} \rho_\delta\left(\left\| \Delta T\, p_{j,b}^{u_j-\kappa} - p_{e,a}^{u_j-\kappa} \right\|_2\right), \tag{14}$$

where each $k = (a, b) \in P_j$, $p_{j,b}^{u_j-\kappa}$ and $p_{e,a}^{u_j-\kappa}$ are the neighbor and ego box centers at the shared history offset κ, and $L_p = k_{\text{hist}}$ in our implementation. Here ρ_δ is the Huber cost

$$\rho_\delta(r) = \begin{cases} \frac{1}{2} r^2, & r \le \delta, \\ \delta(r - \frac{1}{2}\delta), & r > \delta. \end{cases} \tag{15}$$

Thus, pair weight $q_{j,k}^{(\ell)}$ serves as the per-pair weight in IRLS, while ρ_δ provides robustness to outlier point residuals. In the released rule-based configuration, the solver uses 10 iterations, $\delta = 0.5$, and convergence tolerance 10^{-4}.

Anchor state, prior covariance, and propagation. For a matched pair $k = (a, b) \in P_j$, the delayed anchor state is represented as

$$x_{j,k}^{u_j} = [p_x, p_y, v_x, v_y, \psi]^T. \tag{16}$$

The planar velocity components v_x and v_y are expressed in meters per frame, obtained by converting physical velocities using the frame interval. Let τ_j denote the communication delay of neighbor j, so the current time satisfies $t = \tau_j + u_j$, and let Δt_{frame} denote the frame interval. For both datasets, $\Delta t_{\text{frame}} = 100$ ms, corresponding to a sampling rate of 10 Hz. We convert the delay to frame units as $\Delta n_j = \tau_j / \Delta t_{\text{frame}}$. The initial covariance converted from $q_{j,k}^{(0)}$ is

$$\Sigma_{j,k}^{u_j} = \text{diag}\left(\frac{\alpha_p}{q_{j,k}^{(0)} + \varepsilon_q}, \frac{\alpha_p}{q_{j,k}^{(0)} + \varepsilon_q}, \frac{\alpha_v}{q_{j,k}^{(0)} + \varepsilon_q}, \frac{\alpha_v}{q_{j,k}^{(0)} + \varepsilon_q}, \frac{(\alpha_\psi^{\text{rad}})^2}{q_{j,k}^{(0)} + \varepsilon_q}\right), \tag{17}$$

where $\alpha_\psi^{\text{rad}} = (\pi/180)\alpha_\psi^{\text{deg}}$ is the yaw scale expressed in radians.

The constant-velocity state transition used in the implementation is

$$A(\Delta n_j) = \begin{bmatrix} 1 & 0 & \Delta n_j & 0 & 0 \\ 0 & 1 & 0 & \Delta n_j & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{18}$$

and the process noise is

$$\begin{aligned} Q(\Delta n_j) = \text{diag}\Big(&\sigma_a^2 \max(\Delta n_j^2, 1), \sigma_a^2 \max(\Delta n_j^2, 1), \\ &\sigma_a^2 \max(\Delta n_j, 1), \sigma_a^2 \max(\Delta n_j, 1), \\ &(\sigma_{\psi,Q}^{\text{rad}})^2 \max(\Delta n_j, 1)\Big). \end{aligned} \tag{19}$$

The propagated prior state and covariance are then

$$x_{j,k}^- = A(\Delta n_j) x_{j,k}^{u_j}, \qquad \Sigma_{j,k}^- = A(\Delta n_j)\Sigma_{j,k}^{u_j} A(\Delta n_j)^T + Q(\Delta n_j). \tag{20}$$

Observation model, NIS gate, and update. The current-time ego observation measures only (x, y, ψ), so the implementation uses

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \qquad R = \text{diag}\left(\sigma_{z,p}^2, \sigma_{z,p}^2, (\sigma_{z,\psi}^{\text{rad}})^2\right). \tag{21}$$

Let $z_{j,k}^t = [z_{j,k,x}^t, z_{j,k,y}^t, z_{j,k,\psi}^t]^T$ be the current-time ego observation associated with pair k. Following Eq. (6) in the main text, the

Table 5: Shared structural coefficients used in all experiments.

Module	Parameter	Value	Role
Tracking	k_{hist}	3	Number of historical frames used to build short tracklets.
Matching	d_{gate}	4.0	Position gate applied before a candidate pair is considered.
	λ_{vel}, λ_{graph}	1.0, 0.5	Relative weights of the velocity and local-graph terms in Eq. (12).
	T_{vel}, T_{graph}	2.0, 3.0	Normalizers for the velocity and graph terms in Eq. (12).
	K, w_{edge}, w_{pair}	3, 1.0, 0.5	Neighbor count and edge/pair signature weights for local layout consistency.
Pose	$N_{\text{trk}}^{\min}$	3	Minimum number of matched trajectories required before running pose refinement.
	n_{IRLS}, δ, ϵ_{IRLS}	10, 0.5, 10^{-4}	Shared Huber-IRLS configuration using pair weight $q_{j,k}^{(\ell)}$ as the per-pair weight.
Statistical gate	τ_χ	7.815	Fixed $\chi^2_{0.95}(3)$ gate for the 3-DoF observation (x, y, ψ).
Feedback residual	$\lvert\mathcal{K}_{j,k}\rvert$	3	Number of historical frames used in the short-history term of Eq. (24).
Closed-Loop Posterior Scoring	N_{loop}	1	Number of outer refinement rounds driven by Eq. (26).
	$q_{\min}$	10^{-4}	Minimum pair weight retained for the next pose-refinement round after Eq. (26).
Numerical constants	ε_q	10^{-6}	Small stabilizers in Eqs. (17).

Table 6: Calibrated Kalman-style uncertainty parameters.

Group	Parameter	Value	Role
Prior mapping	T_q	1.5	Cost-to-initial-weight temperature in Eq. (13).
	α_p	0.1	Position prior scale in Eq. (17).
	α_v	0.03	Velocity prior variance in Eq. (17).
	α_ψ^{deg}	3.0	Yaw prior std factor in Eq. (17).
Process noise Q	σ_a	0.03	Translational process noise in Eq. (19).
	$\sigma_{\psi,Q}^{\text{deg}}$	4.0	Yaw process noise in Eq. (19).
Measurement noise R	$\sigma_{z,p}$	0.7	Position observation noise in Eq. (21).
	$\sigma_{z,\psi}^{\text{deg}}$	8.0	Yaw observation noise in Eq. (21).
Feedback-decay coefficient	λ_{fb}	1.5	Feedback-decay coefficient in Eq. (26).

measurement residual and its covariance are

$$v_{j,k}^t = \begin{bmatrix} z_{j,k,x}^t - x_{j,k,x}^- \\ z_{j,k,y}^t - x_{j,k,y}^- \\ \text{wrap}_{1/2}(z_{j,k,\psi}^t - x_{j,k,\psi}^-) \end{bmatrix}, \qquad S_{j,k}^t = H\Sigma_{j,k}^- H^T + R. \quad (22)$$

$\text{wrap}_{1/2}(\cdot)$ denotes half-period yaw wrapping, i.e., mapping an angle difference to its principal value in the interval $[-\pi/2, \pi/2)$ under the π-periodic box-orientation ambiguity. This design choice is made to accommodate the common front-rear direction ambiguity in upstream 3D bounding box predictions. The normalized innovation squared is

$$d_{j,k}^2 = \left(v_{j,k}^t\right)^T \left(S_{j,k}^t\right)^{-1} v_{j,k}^t. \quad (23)$$

The measurement update is accepted only when $d_{j,k}^2 \le \tau_\chi$, where we fix $\tau_\chi = \chi^2_{0.95}(3) = 7.815$.

Feedback residual and Closed-Loop Posterior Scoring. For each matched pair, the implementation first computes a short-history disagreement term over the delayed frame and several nearby history frames:

$$r_{j,k}^{\text{hist}} = \frac{1}{|\mathcal{K}_{j,k}|} \sum_{\kappa \in \mathcal{K}_{j,k}} \Big(\frac{\| \Delta T_{j\to e}^{(\ell)} p_{j,b}^{u_j-\kappa} - p_{e,a}^{u_j-\kappa} \|_2^2}{\sigma_{z,p}^2} + \frac{\text{wrap}_{1/2}(\hat{\psi}_{j,b}^{u_j-\kappa,(\ell)} - \psi_{e,a}^{u_j-\kappa})^2}{(\sigma_{z,\psi}^{\text{rad}})^2} \Big), \quad (24)$$

where $\mathcal{K}_{j,k}$ contains the available history offsets and the delayed frame itself when $\Delta n_j > 0$, and $\hat{\psi}_{j,b}^{u_j-\kappa,(\ell)}$ denotes the yaw of the neighbor box after applying the current loop-round transform

$\Delta T_{j\to e}^{(\ell)}$. The feedback residual is then defined as

$$r_{j,k}^{\mathrm{fb}} = d_{j,k}^2 + r_{j,k}^{\mathrm{hist}}. \tag{25}$$

In Closed-Loop Posterior Scoring, the next-round pair weight is updated as

$$q_{j,k}^{(\ell+1)} = q_{j,k}^{(0)} \exp\left(-\lambda_{\mathrm{fb}}\, r_{j,k}^{\mathrm{fb}}\right). \tag{26}$$

Pairs whose pair weight falls below the minimum threshold $q_{\min}$ are discarded before the next pose-refinement round. The updated pair weight $q_{j,k}^{(\ell+1)}$ is then fed back to the IRLS pose solver in Eq. (14) for the next round of spatial refinement, explicitly closing the spatio-temporal alignment loop.

B.3 Shared and Calibrated Rule-based Parameters

Table 5 lists the structural coefficients that are fixed throughout all experiments. These parameters control the matching cost, local graph signatures, gating rule, and IRLS behavior. We keep them unchanged because their role is mainly structural rather than dataset-specific.

Table 6 lists the small set of Kalman-style uncertainty parameters used by the Kalman-style update, including the prior mapping, process noise, measurement noise, and closed-loop decay terms. Unlike the matching coefficients above, these parameters directly determine the uncertainty scale of delayed anchors and current-time observations, and are therefore the only groups that require light calibration.

B.4 Parameter Calibration

We calibrate the Kalman-style uncertainty parameters in Table 6 only once on the training split of V2V4Real, and then keep the resulting values unchanged in all reported experiments, including OPV2V and all test conditions. We choose V2V4Real as the calibration benchmark because it is the noisier and more realistic dataset, making it a more suitable reference point for setting the uncertainty scale of the object-level module.

The calibration is carried out under the representative **Joint-Hard** setting used in the main text, namely a communication delay of 200 ms together with pose noise (0.6 m, 0.6°). This operating point reflects the coupled spatio-temporal perturbation regime that our method is primarily designed to handle, while avoiding benchmark-specific tuning across multiple conditions.

The calibration itself is intentionally lightweight. We first fix the statistical gate to $\chi^2_{0.95}(3) = 7.815$, corresponding to a 95% confidence gate for the 3-DoF observation (x, y, ψ). We also use a target normalized innovation scale of 3 as the coarse initialization reference, since the current-time ego observation contains exactly three observed degrees of freedom. Starting from this reference scale, we then perform a small local sweep over the uncertainty-related groups in Table 6 on the V2V4Real training split, while keeping all matching and IRLS coefficients fixed.

The resulting parameter set is then reused unchanged on OPV2V and under all evaluation settings. Although this one-time calibration is not guaranteed to be strictly optimal for OPV2V, we observe no material degradation in our experiments. This suggests that the rule-based object-level module is not highly sensitive to benchmark-specific retuning once the uncertainty scale is set to a reasonable range.

Finally, the object-level closed-loop module itself contains no learnable parameters. The learned part of the overall system remains the detector and downstream fusion network described in the main text.

B.5 Hyperparameter Sensitivity

To verify that the reported gain is not tied to a narrow rule-based parameter choice, we vary the main structural and uncertainty-related hyperparameters individually on V2V4Real under *Joint-Hard*. All other settings are held at their default values. As summarized in Table 7, all tested changes except the stricter 2 m distance gate remain within 0.29 AP@0.5 of the default configuration. The result is consistent with the one-time calibration strategy above and indicates that the object-level module is insensitive to moderate parameter changes.

Table 7: Hyperparameter sensitivity on V2V4Real under *Joint-Hard*. The default AP@0.5 is 67.04.

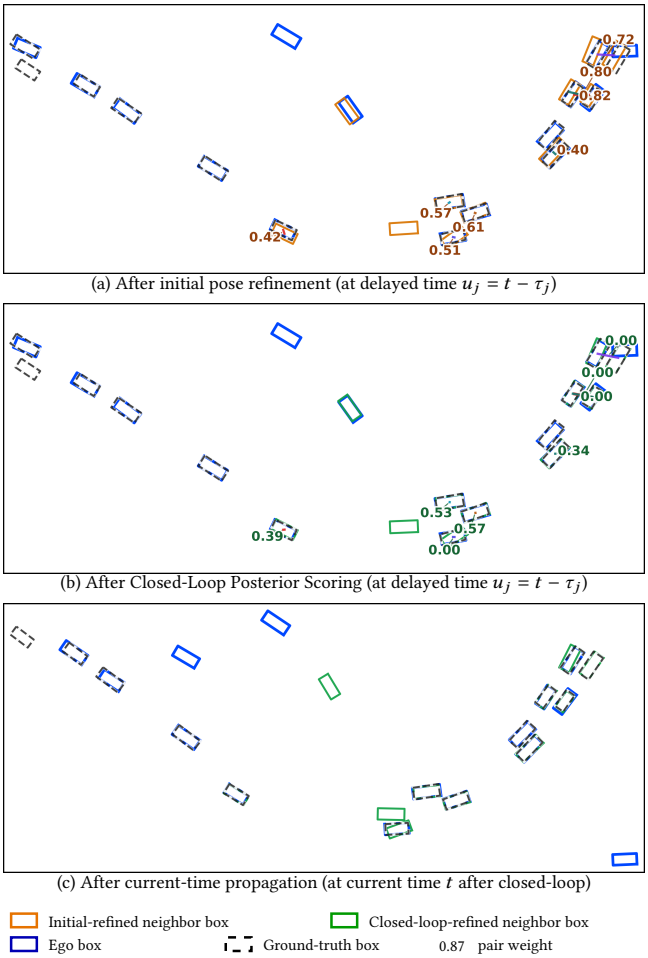
Parameter	Default	Test values	ΔAP@0.5
Distance gate (m)	4	2 / 6	−0.62 / −0.05
History length (frames)	3	2 / 4	−0.10 / +0.07
NIS gate (%)	95	90 / 99	−0.08 / −0.07
Feedback decay	1.5	1.0 / 2.0	−0.08 / −0.04
Huber penalty	0.5	0.25 / 1.0	−0.17 / −0.29
Minimum solver support	3	4 / 5	−0.10 / −0.15

C Additional Qualitative Results

Figure 7 visualizes three representative stages of our object-level pipeline. All three panels are taken from an OPV2V example under a 400 ms communication delay and pose noise (0.6 m, 0.6°).

Together, the panels illustrate both the closed-loop reliability update and the remaining limitation of the simple rule-based propagation model. The panel (a) shows the initial pose-refinement result before Closed-Loop Posterior Scoring. In this example, the estimated pose is still poor, and several mismatched pairs remain because the delayed collaborator boxes are affected jointly by inaccurate detections and initial relative-pose error. This case illustrates the limitation of relying only on one-shot delayed-frame matching and pose refinement: when the input correspondences are already corrupted, the resulting alignment can still be unreliable even if the optimization itself is well defined.

The panel (b) shows the pose after Closed-Loop Posterior Scoring. Compared with the initial result, the refined alignment becomes much cleaner and most corresponding boxes are brought into close agreement. Importantly, four previously matched pairs are assigned extremely small pair weight values (below 10^{-2}), which effectively suppresses their influence in the next pose-refinement round. This example directly reflects the role of Closed-Loop Posterior Scoring: instead of trusting all delayed-frame correspondences equally, the



(a) After initial pose refinement (at delayed time $u_j = t - \tau_j$)

(b) After Closed-Loop Posterior Scoring (at delayed time $u_j = t - \tau_j$)

(c) After current-time propagation (at current time t after closed-loop)

Figure 7: Representative qualitative example of the object-level pipeline in *CoAnchor*. This example visualizes how *CoAnchor* progressively improves delayed collaborator alignment through object-level pose refinement, Closed-Loop Posterior Scoring, and temporal propagation.

method re-evaluates them using current-time agreement and short-history consistency, and then downweights pairs that are no longer trustworthy.

The panel (c) shows the propagated collaborator boxes after rewriting and temporal propagation to the current ego time. Most delayed neighbor boxes are now well aligned with the ego-side observations, indicating that the object-level propagation is effective in bringing stale collaborator evidence closer to the current frame. At the same time, several boxes still exhibit residual offsets rather than perfect overlap. This limitation is expected: under delays of several hundred milliseconds, assuming approximately constant-velocity straight-line motion is a reasonable and computationally efficient approximation, but it cannot fully model all real object motions. As a result, while the propagation stage is practically useful and aligns most objects well, its simplicity can still become a bottleneck in some hard cases.

D Extended Experimental Results

D.1 Additional Motivation Results

Table 8 provides the detailed measurement results for the pilot study discussed in Sec. 3 of the main text.

Table 8: Additional motivation measurements. Method comparison under fixed 100ms communication delay with varying Gaussian pose noise.

AP@0.5 / AP@0.7 in OPV2V (100ms fixed)			
Method	**None**	(0.3 **m**, 0.3°)	(0.6 **m**, 0.6°)
CoAlign [15]	95.24 / 80.50	91.61 / 76.39	88.52 / 74.90
TraF-Align (ego-only)	84.49 / 78.57	84.49 / 78.57	84.49 / 78.57
TraF-Align [19]	**96.72 / 92.12**	89.50 / 73.81	84.10 / 68.95
baseST [15, 19]	96.58 / 91.55	**96.26 / 90.06**	**89.89 / 80.55**

AP@0.5 / AP@0.7 in V2V4Real (100ms fixed)			
Method	**None**	(0.3 **m**, 0.3°)	(0.6 **m**, 0.6°)
CoAlign [15]	69.86 / 40.30	61.60 / 34.93	55.16 / 33.44
TraF-Align (ego-only)	62.19 / 40.55	62.19 / 40.55	**62.19 / 40.55**
TraF-Align [19]	**78.76 / 50.80**	63.20 / 37.47	57.82 / 34.89
baseST [15, 19]	73.16 / 43.86	**68.22 / 41.05**	60.68 / 37.39

The OPV2V results indicate that the cascaded baseline *baseST* can be beneficial on the cleaner simulated benchmark, although it does not yield a gain in every case: at zero pose noise, it is slightly below TraF-Align (96.58 / 91.55 vs. 96.72 / 92.12), but once the pose perturbation increases to 0.3 and 0.6, it becomes clearly stronger than TraF-Align (96.26 / 90.06 vs. 89.50 / 73.81, and 89.89 / 80.55 vs. 84.10 / 68.95). This suggests that under relatively clean detections and correspondences, explicit spatial refinement can still provide a beneficial upstream reference for temporal compensation, but its benefit is conditional rather than unconditionally guaranteed.

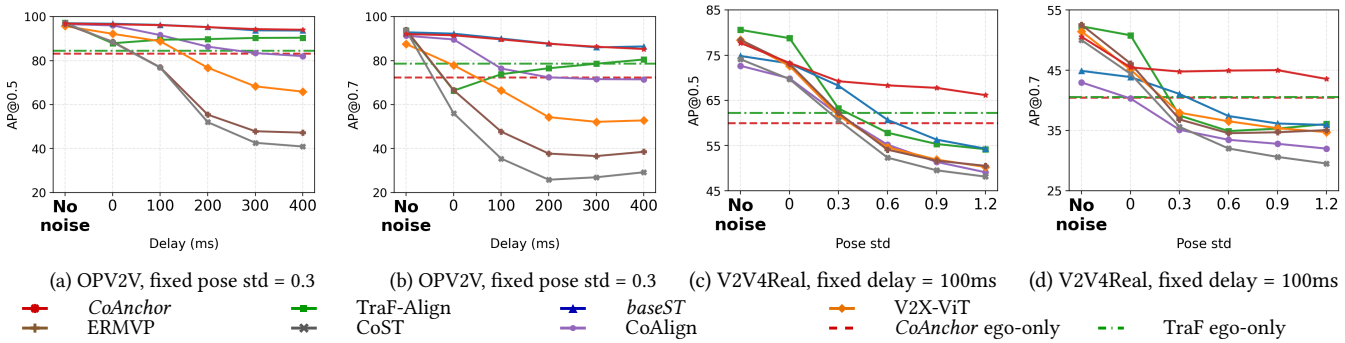
On V2V4Real, the pattern is consistent with the measurement study in the main text: direct cascading does not provide a stable gain over TraF-Align, and under stronger pose perturbation it can even fall below the ego-only reference. This further supports that simply stacking spatial pose refinement and temporal compensation is insufficient in realistic scenes.

D.2 Additional Robustness Curves

Figure 8 complements the robustness analysis in the main text by adding the four omitted sweep directions. Together with the corresponding figure in the main text, they complete the delay/noise sweeps on both datasets.

Here we focus on one specific phenomenon: the non-monotonic delay behavior of TraF-Align. A mild version of this pattern is already visible in the V2V4Real delay sweep in the main text, and the OPV2V delay sweep in Fig. 8(a)–(b) makes it much clearer. Under fixed pose noise, TraF-Align first drops substantially as delay increases from the no-noise point, but then partially rebounds when the delay becomes even larger, especially at AP@0.7.

One possible explanation is that TraF-Align explicitly encodes delay in the collaborator branch, so the model can learn to reduce the effective weight of neighbor information once the delay becomes very large. This suppression may reduce false positives



(a) OPV2V, fixed pose std = 0.3 (b) OPV2V, fixed pose std = 0.3 (c) V2V4Real, fixed delay = 100ms (d) V2V4Real, fixed delay = 100ms

Figure 8: Additional robustness curves omitted from the main text. Comparison with state-of-the-art methods on OPV2V and V2V4Real. Each entry is reported as AP@0.5 / AP@0.7.

introduced by stale collaborator features, and the effect is particularly visible at AP@0.7, where small localization biases are more likely to turn a seemingly plausible prediction into a mismatch. However, this strategy is still implicit and coarse. It is not tied to an explicit object-level notion of trustworthiness, and it reduces collaborator influence in a relatively non-selective manner. As a result, it may suppress some harmful stale evidence, but it also discards many useful collaborative cues and therefore weakens the potential collaborative gain.

The curves also suggest that under coupled pose and delay perturbations, pose noise cannot be handled merely by reducing the influence of collaborator information. In the V2V4Real pose-noise sweep of Fig. 8(c)–(d), our method maintains comparatively robust performance as the pose noise increases under fixed delay. A reason is that our method does not treat collaborator information in an all-or-nothing way. Instead, it propagates delayed hypotheses at the object level and re-checks them against current-time ego observations, so that collaborator evidence can be filtered more selectively according to its current spatial trustworthiness.

D.3 High-Motion Subset Analysis

We further examine the constant-velocity prior on a high-motion subset of V2V4Real under the *Joint-Hard* setting, which uses a 200 ms communication delay and pose noise $(0.6\,\text{m}, 0.6^\circ)$. The subset is selected exclusively from *ground-truth object trajectories*, rather than predictions from any evaluated method, so all methods are evaluated on exactly the same frames. A frame is selected if any co-visible ground-truth object has a yaw change larger than 15° across adjacent annotated frames, or a center residual larger than 0.5 m when its two preceding ground-truth states are used to form a constant-velocity prediction of the current state. This procedure selects 379 out of 1,993 test frames (19.0%).

Table 9: Results on the V2V4Real high-motion subset under *Joint-Hard*. Each entry is AP@0.5 / AP@0.7.

Method	All frames	High-motion subset
TraF-Align [19]	58.18 / 35.09	56.35 / 33.50
baseST [15, 19]	61.04 / 37.69	58.83 / 36.01
CoAnchor	**67.04 / 44.05**	**62.76 / 40.63**

As shown in Table 9, the selected subset is harder for all methods. Nevertheless, *CoAnchor* retains a clear advantage: it outperforms TraF-Align by 6.41 / 7.13 AP and *baseST* by 3.93 / 4.62 AP on the high-motion subset. The result indicates that nonlinear motion degrades the constant-velocity prior but does not eliminate the benefit of object-level verification: unreliable innovations can still be rejected by the statistical gate or assigned small closed-loop weights before they dominate pose refinement.