BEVSync: Asynchronous Data Alignment for Camera-based Vehicle-Infrastructure Cooperative Perception Under Uncertain Delays

Wentao Wang[1], **Jiaqian Wang**[1,2], **Yuxin Deng**[1], **Guang Tan**[1]*,

[1] School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-sen University
[2] Peng Cheng Laboratory
{wangwt66, wangjq86, dengyx35}@mail2.sysu.edu.cn, tanguang@mail.sysu.edu.cn

Abstract

Vehicle-to-infrastructure (V2I) cooperative perception systems can enhance the sensing abilities of autonomous vehicles. Existing V2I solutions often consider LiDAR devices instead of cameras, the most prevalent sensors with low cost and wide installation. In addition, a major challenge that has been underexplored is the time asynchrony between image frames from different sources. This asynchrony arises because of clock differences, varying times involved in data processing and transmission, causing *uncertain delays* that complicate data alignment and potentially reduce perception accuracy. We propose BEVSync, a camera-based V2I cooperative perception system that adaptively aligns frames from the ego-vehicle and infrastructure by compensating for motion deviations. Specifically, we develop an extractor-compensator model to extract and predict perceptual features using historical frames, thereby smoothing out the data misalignment. Experiments on the real-world dataset DAIR-V2X show that our approach surpasses existing methods in terms of performance and robustness.

Introduction

Autonomous driving relies on complete and precise environmental perception (Wang, Xu, and Tan 2024; Li et al. 2022a). The advancement of Vehicle-to-everything (V2X) communication technology has enabled the connected autonomous vehicles and infrastructure to exchange information, thereby enhancing the vehicles' perception capability (Lin et al. 2022; Fan et al. 2023; Xu et al. 2022b, 2023). In particular, the vehicle-to-infrastructure (V2I) approach emerges as a viable solution due to the widespread deployment of roadside units (Shi et al. 2022). In this paper, we focus on utilizing the V2I approach to improve the perception performance of autonomous vehicles.

We consider cameras as the primary sensors of the infrastructure. Cameras can obtain rich semantic information (Li, Tan, and Gou 2024), are cost-effective, and have been widely installed in the real world. When mounted on the poles, cameras can provide longer-range perception and are less affected by occlusion compared to vehicle-mounted cameras. Recent studies have shown that roadside cameras can perform 3D object detection with decent accuracy (Zou et al. 2022; Yang et al. 2023). This success has led to further research into camera-based cooperative perception (Wang et al. 2023; Runsheng Xu 2022).

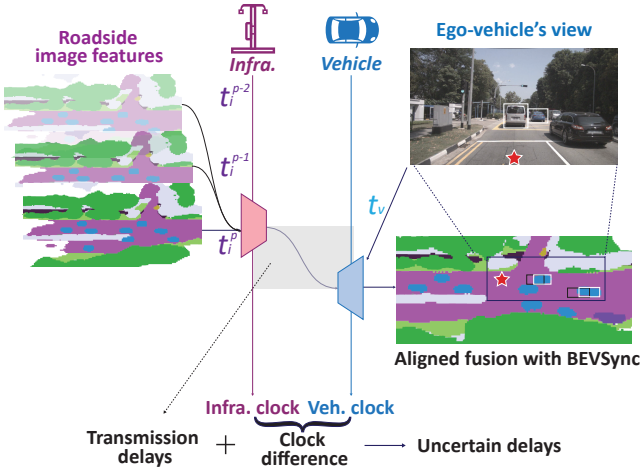

Figure 1: Data fusion under uncertain delays in a V2I environment. The data from the infrastructure are subject to various delays when arriving at the ego-vehicle. The left part of the diagram shows the multi-frame features to be transmitted; the *star* in the right part represents the ego-vehicle. The black boxes indicate the unaligned detection results produced by a naive fusion method, whereas the white boxes show the aligned results achieved using BEVSync.

Current cooperative perception approaches commonly rely on the assumption of temporal synchrony between the ego-vehicle and infrastructure (Wei et al. 2024). That means data frames from different sources can be perfectly aligned in time, requiring only spatial alignment for data fusion. However, this assumption may not hold in real-world scenarios. The infrastructure clock may not be precisely synchronized with the ego-vehicle's clock, especially when using traditional camera devices. Furthermore, an image from the infrastructure typically goes through several phases: encoding, transmission to servers, decoding, application-related processing (e.g., privacy protection), and finally, transmission to the ego-vehicle. The clock difference, combined with the highly dynamic processing and transmission de-

*Corresponding author.

lays, makes it challenging to align and fuse the data in a static way. Simply employing an existing fusion technique without considering the issue of such uncertain delays may result in suboptimal perception accuracy. Figure 1 illustrates the cooperation process and highlights the delay factor.

We present BEVSync, a novel camera-based cooperative perception approach that is able to adaptively fuse image features from the ego-vehicle and infrastructure in the presence of uncertain delays. The key idea is to compensate for the motion reflected by the delayed data for alignment. Specifically, BEVSync extracts motion information from the historical infrastructure-side frames and encodes it into the messages sent to the ego-vehicle. Using the received messages, the ego-vehicle can align the asynchronous features using an extractor-compensator model. This model is trained based on the sequences of historical feature frames and is capable of predicting future features.

We extensively evaluate the proposed BEVSync framework on the DAIR-V2X real-world dataset (Yu et al. 2022). Experimental results show that BEVSync outperforms existing camera-based fusion methods under varying delays.

The main contributions of this paper are three-fold:

- We propose a novel asynchronous perception fusion framework to tackle the challenge of uncertain temporal misalignment in camera-based vehicle-infrastructure cooperative systems.
- We design an attention-based extractor-compensator model to address the issue of data asynchrony. It can effectively counteract the delay-induced perception bias by predicting future features.
- We conduct a series of experiments to show that our proposed method achieves significant improvements in cooperative perception scenarios under uncertain delays compared with the existing methods.

Related Work

Cooperative Perception

In the context of V2X communications, V2VNet (Wang et al. 2020) uses multi-round message passing via graph neural networks to achieve better perception performance. DiscoNet adopts knowledge distillation to train a fusion module for intermediate fusion (Li et al. 2021). Coopernaut (Cui et al. 2022) uses V2V sharing information to generate control policies for end-to-end autonomous driving. Some works adopt roadside devices to assist vehicles in achieving extended perception range via cooperation. DAIR-V2X (Yu et al. 2022) is a pioneering work in V2X cooperation, which introduces the vehicle-infrastructure target detection task and provides early and late fusion benchmarks. V2X-ViT (Xu et al. 2022a) designs a heterogeneous attention mechanism to aggregate information from both connected vehicles and infrastructures. VIMI (Wang et al. 2023) proposes an intermediate fusion method for multi-view camera fusion in camera-based vehicle-infrastructure cooperative scenarios.

Camera-based BEV perception

Learning powerful features in the bird's-eye-view (BEV) for perception tasks is drawing extensive attention from industry and academia (Ma et al. 2023; Liu et al. 2023; Li et al. 2023a). BEVFormer (Li et al. 2022b) constructs BEV queries and explores spatial cross-attention and temporal self-attention to recurrently refine the BEV features. BEVDepth (Li et al. 2023b) asserts that the quality of intermediate depth estimation is crucial for enhancing multi-view 3D object detection. To this end, the method introduces explicit depth supervision using ground-truth depth data derived from point clouds. BEVHeight (Yang et al. 2023) investigates the camera-based roadside 3D Object detection and claims that the height to the ground achieves a distance agnostic formulation to ease the optimization process of camera-only methods.

Asynchronous Perception Fusion

Considering transmission delay, some methods have been proposed to achieve robust cooperative perception. VIPS (Shi et al. 2022) directly transmits detection results to the ego vehicles and compensates for delays using Kalman filtering (Welch, Bishop et al. 1995). SyncNet leverages historical multi-frame information and employs the Convolution LSTM (Shi et al. 2015) to compensate for the current frame (Lei et al. 2022). Some works apply flows to asynchronous detection tasks, which contain the motion information over time (Deville and Gatski 2012; Zhu et al. 2017). FFNet (Yu et al. 2024) leverages the flow of features to enable the transmitted information with linear prediction ability. CoBEVFlow (Wei et al. 2024) adopts BEV flow with time series forecasting to handle the temporal irregular messages from cooperative agents.

Methodology

Problem Definition

In our system, the input consists of two parts:

- Images $X_v(t_v)$, captured by the ego-vehicle camera with timestamp t_v according to its own clock, along with its pose parameters;
- Images $X_i(t_i)$, captured by the roadside camera with timestamp t_i according to the roadside clock, along with its pose parameters.

Although the infrastructure and vehicle are both equipped with cameras, the difference of camera poses results in varying viewing perspectives. Therefore, we use side-specific encoders to extract corresponding BEV features. We adopt BEVHeight (Yang et al. 2023) to process the input images, which can predict per-pixel height for roadside 3D object detection. For better computational efficiency, we use the ResNet-50 as the image encoder to extract features from the input image $X_i(t_i)$. Then the BEV features $F_i(t_i) \in R^{H \times W \times C}$ that combine extracted image features and the predicted height distribution are generated via the voxel pooling operation.

The ego-vehicle is equipped with a forward-facing camera, producing images $X_v(t_v)$. We aim to generate a BEV

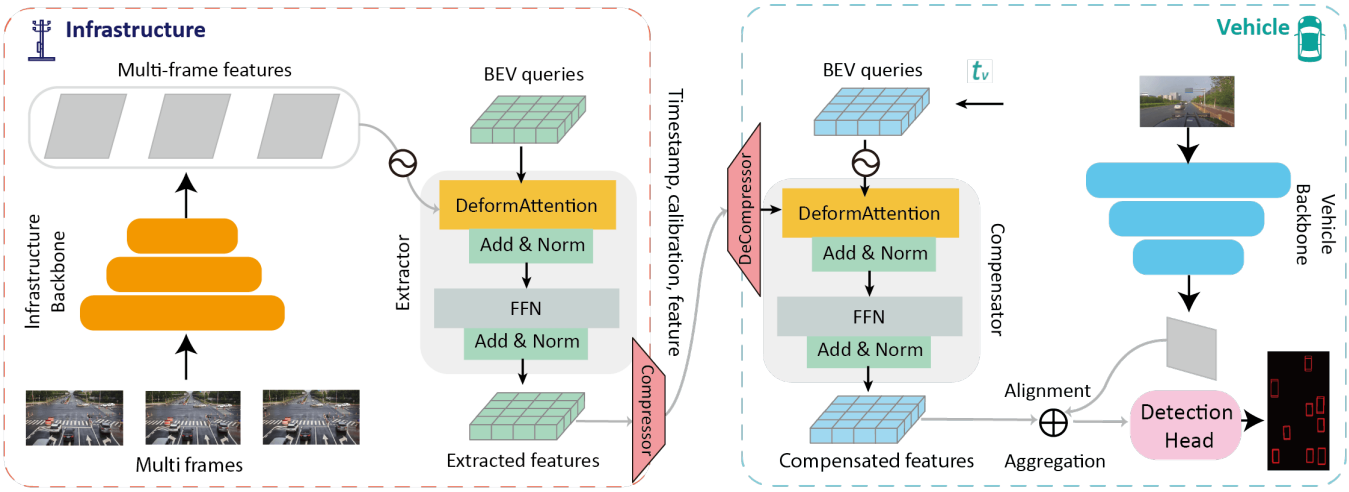

Figure 2: **BEVSync framework.** On the infrastructure side, we extract motion information from the historical BEV frames. On the vehicle side, we utilize the received features to achieve spatio-temporal alignment. The aligned features are then fused with the ego-vehicle features to output detection results.

representation that allows fusion with the shared features from the infrastructure. We adopt a similar architecture to BEVDepth (Li et al. 2023b) with no depth supervision. The ResNet-50 is used as the image backbone to extract 2D image features. The refined BEV feature map $F_v^{t_v} \in R^{H\times W\times C}$ is under the coordinate system of ego-vehicle.

Note that, compared to LiDAR point clouds, RGB images lack sufficient spatial information, making it more challenging to extract the motion information of objects in complex traffic scenarios.

Overall Architecture

As depicted in Figure 2, BEVSync consists of three main modules. The first one generates BEV features along with motion information, the second transmits and aligns the generated BEV features, and the third aggregates the aligned features and outputs the detection results.

Firstly, let the $X_i(t_i)$ be the raw observation at timestamp t_i of the infrastructure. The main computations involved are as follows:

$$F_i(t_i) = \text{ENC}_i(X_i(t_i)), \tag{1a}$$
$$F_v(t_v) = \text{ENC}_v(X_v(t_v)), \tag{1b}$$
$$\hat{M}_i(t_i) = \text{EXTR}(\{\{F_i(t_i^p)\}\}_{p=i-k+1,\ldots,i}, \tag{1c}$$
$$\hat{F}_i(t_v) = \text{COMP}(t_v, \hat{M}_i(t_i)), \tag{1d}$$
$$\hat{H}(t_v) = \text{AGG}(\hat{F}_i(t_v), F_v(t_v)), \tag{1e}$$
$$\hat{Y}_v(t_v) = \text{HEAD}_(\hat{H}(t_v)), \tag{1f}$$

where $F_i(t_i) \in R^{H\times W\times C}$ is the BEV feature of infrastructure at time t_i, $\hat{M}_i(t_i) \in R^{H\times W\times C}$ is the fused feature of historical multi-frame BEV feature $\{F_i(t_i^p)\}_{p=i-k+1,\ldots,i}$ received from infrastructure, which contains key motion information, $\hat{H}(t_v)$ is the aggregated features of both ego-vehicle and infrastructure, $\hat{Y}_v(t_v)$ is the final output of the system.

Equations 1a and 1b extract features from raw observations, Equation 1c leverages historical infrastructure features to perceive transmitted messages with predictable motion information, Equation 1d obtains the estimated synchronous feature by compensating for the motion deviation. Equation 1e aggregates the aligned infrastructure and ego-vehicle feature. Finally, Equation 1f outputs the final detection results with a detection head.

Motion Extraction and Compensation

After obtaining a multi-frame feature sequence from the infrastructure, we aim to enable it to support extrapolation while sending it to the ego-vehicle. Thus we use a network to extract motion information from historical features. The per-frame BEV feature with size $[H, W, C]$ is obtained via the infrastructure encoder. k is set as 3, which is the minimum size required to extract sufficient motion information. Thus we use three frames of infrastructure-side BEV maps $F_i(t_i^{p-2})$, $F_i(t_i^{p-1})$ and $F_i(t_i^p)$.

Also, we use trigonometric functions to encode the time into features, where the time coding is

$$(u(t))_{2e} = \sin(\frac{t}{10000^{2e/d}}), (u(t)_{2e+1}) = \cos(\frac{t}{10000^{2e/d}}) \tag{2}$$

where e is the index of temporal encoding. In this way, the temporal information can be added to the BEV feature sequence. We implement this encoding process through deformable DETR (Zhu et al. 2020). The deformable attention (DeformAttn) mechanism is used to interact with local regions of interest, which samples only K points in the vicinity of each reference point and computes the attention results, resulting in high efficiency and dramatically reducing the training time. When dealing with BEV features, the multi-head attention mechanism in the original Transformer can be highly expensive since the number of queries is likely to be large. In practice, the vehicles typically travel cohe-

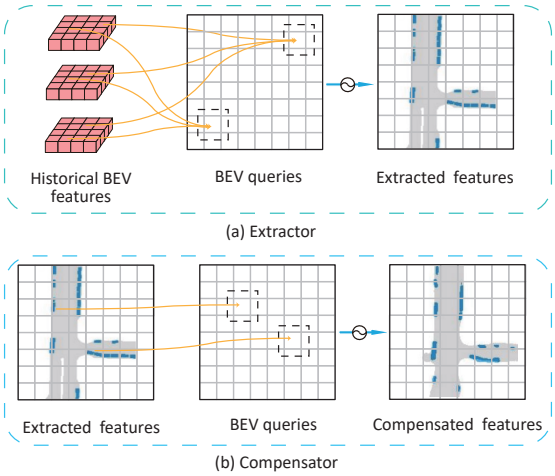

Figure 3: The extractor and compensator run separately on the infrastructure and the vehicle, respectively. (a) Extractor: each BEV query interacts with historical features from the infrastructure. (b) Compensator: each BEV query is combined with temporal information and interacts with the extracted features.

sively along the lanes. Thus extracting motion information only requires querying within these regions.

As depicted in 3 (a), We set a group of grid-shaped learnable parameters $Q \in R^{H\times W\times C}$ as the queries of DeformAttn, where learnable positional embedding is added before inputting them. The key and value both are the sum of the historical feature sequences and their corresponding time code set $V(t)$:

$$\hat{M}_i(t_i) = \text{DfmAttn}(Q, \text{MLP}(\mathcal{F}_i) + U(t), \text{MLP}(\mathcal{F}_i) + U(t)) \quad (3)$$

where $\hat{M}_i(t_i) \in R^{H\times W\times C}$ is the generated feature with motion information, $\text{MLP}(\cdot)$ is the encoding function of historical sequences $\mathcal{F}_i$.

To reduce transfer size, a convolutional auto-encoder (Masci et al. 2011) is applied to compress and decompress the generated features. A series of 1×1 convolutions are used to compress the feature maps along the channel dimension. After receiving the compressed features, the ego-vehicle uses 1×1 convolutions to project features back.

Together with the compressed features, corresponding timestamps under the infrastructure clock and calibration information are transmitted to the ego-vehicle. The vehicle can obtain precise clock through a GPS receiver (Zheng and Liu 2017), while the infrastructure can typically obtain a more coarse-grained clock via some network services, such as the Network Time Protocol (NTP) (Liang et al. 2018). Considering the local clock rate errors are very small, the clock differences can be considered fixed. We utilize the timestamps from the ego-vehicle's clock to predict future roadside features. The model is trained in such a way that it is enabled to adapt to such clock differences automatically.

Constructing the precise association of the same objects between the BEV features of different times is challenging.

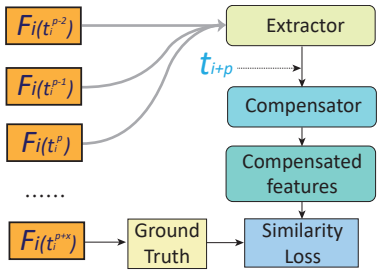

Figure 4: Training process of the extractor and compensator.

Therefore, similar to the extractor, we model this temporal connection through the deformable attention mechanism, which is shown in Figure 3 (b). To predict the features of the current timestamp, i.e., to compensate for motion information, the query is the sum of BEV query and the time code of the current timestamp.

$$\hat{F}_i(t_v) = \text{DfmAttn}(Q + U_L, \hat{M}_i(t_i), \hat{M}_i(t_i)) \quad (4)$$

where Q denotes the BEV query at the current time and U_L is the corresponding time code.

Once obtaining the compensated feature map, the ego-vehicle applies a differentiable spatial transformation operator τ, to geometrically warp the features onto the ego's coordinate system: $H_i = \tau(F_i)$. Then the BEV features of ego-vehicle and infrastructure are both of the ego coordinate system, resulting in spatial alignment. We concatenate them and leverage a self-attention block to aggregate the concatenated features.

Training Details

We divide the training process into two tasks: the asynchrony compensation task and the basic feature fusion task. In the first stage, a basic fusion model is trained end-to-end without accounting for latency. The objective of this stage is to enable BEVSync to fuse infrastructure-side features with ego-vehicle features. Secondly, we train the motion extraction and compensation modules, as depicted in Figure 4.

Specifically, we let the historical BEV features sequence $\{F_i(t_i^{p-2}), F_i(t_i^{p-1}), F_i(t_i^{p})\}$ be a training input and set the target future feature $F_i(t_i^{p+x})$ as the ground truth, which is shown in Fig. 4. The set of training pairs can be formulated as $\mathcal{D} = \{d_{p,x} = \{F_i(t_i^{p-2}), F_i(t_i^{p-1}), F_i(t_i^{p}), F_i(t_i^{p+x})\}\}$. The idea is to endow the compensator with the ability to predict the feature at the current timestamp based on the time interval. We aim to produce new features that aligns predicted feature $\hat{F}_i(t_i^{p+x})$ with ground truth $F_i(t_i^{p+x})$ as closely as possible. We leverage perceptual content similarity (Justin, Alexandre, and Li 2016) to measure the similarity between the compensated BEV feature map and ground truth. Thus the loss function can be formulated as:

$$L_{similarity} = \sum_{\mathcal{D}} \frac{||\hat{F}_i(t_i^{p+x}) - F_i(t_i^{p+x})||_1}{||F_i(t_i^{p+x})||_1} \quad (5)$$

where $||\cdot||_1$ denotes L1 norm, which can measure the difference on the feature maps (Zhao et al. 2016).

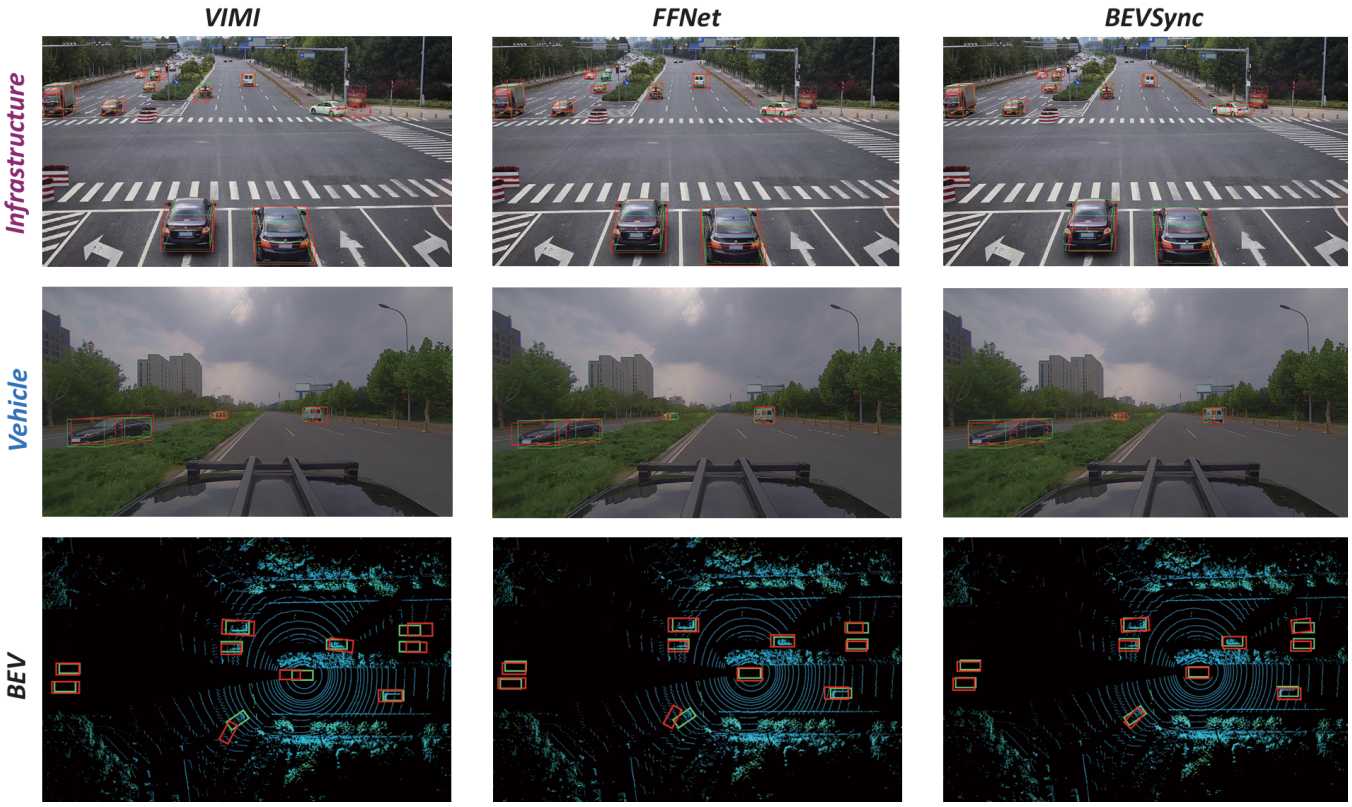

Figure 5: Visualization results of VIMI (left column) (Wang et al. 2023), FFNet (middle column) (Yu et al. 2024) and BEVSync (right column) under a transmission delay of 300 ms. The red boxes indicate *Detected* objects, whereas the green ones represent *Ground truth*. Note that the point cloud is used merely for decoration and not as perceptual observations; the center of the point cloud corresponds to the position of the ego-vehicle.

Experiments

Experiment Settings

Dataset We utilized the publicly available and real-world DAIR-V2X dataset (Yu et al. 2022), which provides more than 100 scenarios and 18,000 data pairs collected from both infrastructure and vehicle-mounted sensors, including cameras and LiDARs at complex traffic intersections. The perception range for evaluation is set to [0, -39, 100, 39] following the official settings. The input images are resized to 1440×900. This dataset provides cooperative 3D annotations with a vehicle-infrastructure cooperative perspective for 9311 pairs, with each object meticulously labeled with its respective categories, such as Car, Bus, Truck, or Van. A data split of 50%, 20% and 30% are used for training, validation and testing, respectively.

Implement details We conduct experiments on the camera-based part of DAIR-V2X (Yu et al. 2022). For each vehicle-infrastructure image pair, The timestamp of the data vehicle and infrastructure side is synchronized. And the time interval of the frame sequence from both sides is 100. Thus We use the past infrastructure frames and the current vehicle frames to simulate asynchronous cooperative messages with delays of $k \times 100$ ms ($k = 1, 2,...$). Considering that the precision of the NTP protocol is typically on the order of 50 ms, we assume the range of clock differences are from -50 ms to 50 ms.

The implementation is based on the MMDetection3D framework (Chen et al. 2019). The basic fusion model is trained on the DAIR-V2X training set for 50 epochs, with a learning rate of 0.001. All training and evaluation were performed on an NVIDIA GeForce RTX 3090 GPU. To train the extractor and compensator, we generate asynchronous sequences based on the infrastructure part of the training set.The detection performance was measured using KITTI (Geiger, Lenz, and Urtasun 2012) evaluation metrics.

Comparison with Different Methods

BEVSync is compared with five baselines:

- Vehicle-only approach without cooperation;
- Early-fusion-based cooperative perception which transmits raw images;
- Naive asynchronous late-fusion-based cooperative perception by using the Kalman Filter (Shi et al. 2022);
- Intermediate-fusion based synchronous V2I cooperation method, VIMI (Wang et al. 2023);
- Flow-based asynchronous V2I cooperation method, FFNet (Yu et al. 2024).

Models	Clock diff. (ms)	Trans. delay (ms)	mAP@3D		mAP@BEV	
			$IoU_{0.3}$	$IoU_{0.5}$	$IoU_{0.3}$	$IoU_{0.5}$
Vehicle-only	–	–	15.83	9.37	18.10	10.84
Early Fusion	–	100	18.92	9.98	26.36	12.82
VIMI (Wang et al. 2023)	–	100	25.74	13.69	33.61	18.93
Late + Kalman (Shi et al. 2022)	0	100	20.63	11.06	28.95	14.63
FFNet (Yu et al. 2022)	0	100	30.18	13.92	36.06	19.30
BEVSync (Ours)	0	100	31.59	14.02	38.86	20.34
Early Fusion	–	300	16.20	9.65	18.63	11.27
VIMI (Wang et al. 2023)	–	300	19.05	11.73	25.60	13.03
Late +Kalman (Shi et al. 2022)	0	300	18.67	10.12	25.74	12.40
FFNet (Yu et al. 2022)	0	300	26.59	13.14	29.52	15.37
BEVSync(Ours)	0	300	28.33	13.80	33.91	16.76
Late + Kalman (Shi et al. 2022)	[-50, 50]	300	18.51	10.06	25.21	12.32
FFNet (Yu et al. 2022)	[-50, 50]	300	26.20	13.01	29.07	15.05
BEVSync (Ours)	[-50, 50]	300	28.25	13.54	33.49	16.30

Table 1: Comparison with different cooperative methods. The best results are highlighted in gray background.

Table 1 presents a summary of the experimental results. We evaluate the aforementioned methods under 100 ms and 300 ms delays. Firstly, it is shown that BEVSync achieves the best performance on the DAIR-V2X dataset, which is notable under a 300 ms transmission delay. Secondly, comparing the vehicle-only perception system, we can see that vehicle-infrastructure cooperation can significantly improve perception performance. Thirdly, it can be observed that the late fusion based on the Kalman filter and BEVSync can alleviate the perception error caused by increasing communication delays. This result indicates that both of them are capable of predicting motion.

As shown in the third section of the table, we further tested the performance under conditions where there were random clock differences between ego-vehicle and infrastructure. To ensure generality, in the tests, the values of the clock difference are generated from a uniform distribution ranging from [-50, 50] ms, and these values are introduced into the compensator. It can be seen that within the error range of the NTP protocol, our model is able to primarily maintain its performance.

We provide a visualization example in Figure 5 to show the results of VIMI, FFNet and BEVSync from three views: the ego-vehicle, the infrastructure, and the BEV. From the samples in (a) and (b), under a 300 ms delay, both VIMI and FFNet's results exhibit substantial deviations to the ground truth. We can see that VIMI cannot effectively detect dynamic targets in the roadside field of view. FFNet performs relatively well for linearly moving objects due to its reliance on a first-order linear prediction approach. However, this method becomes ineffective when objects follow a curved trajectory, resulting in errors when predicting non-linear motion. In contrast, BEVSync handles more complex cases effectively by compensating for motion gaps using historical motion information. This allows BEVSync to maintain more accurate detection and alignment.

Ablation Study

Effectiveness of Extractor and compensator. We study the effectiveness of the Extractor and Compensator modules. We consider two different scenarios with 0 ms and 300 ms delays. We conducted the comparison by separately removing the two modules from BEVSync and instead directly fusing the infrastructure feature. From Table 2, we can see that:

- Both BEVSync without extractor and BEVSync without compensator exhibit an obvious drop in performance under a 300 ms delay. Without the extractor, the data sent from the infrastructure lacks crucial motion information, rendering it impossible to predict future features. Similarly, without the compensator, the model cannot utilize the timestamp to offset the motion gap;
- The experiments under the synchronous condition (i.e.,0 ms delay) indicate that the extractor can slightly improve the performance. That is because even in a synchronized system, due to the complex traffic scenarios, objects that are occluded in the current frame may appear in historical frames and can be captured by the extractor, resulting in enhanced cooperative detection.

Effect of frame window size. We conduct experiments to evaluate the effect of frame window sizes. Table 3 presents the experimental results. We can see that larger window sizes lead to better performance. That is because increasing the window size allows for capturing more motion information, thereby potentially improving the capability of prediction. Note that when the window size is reduced to 2, the performance drops significantly. A window size of 2 leads to the feature sequence containing only linear motion information, causing BEVSync to degrade into linear prediction. For example, when a vehicle in the scenario is turning, inputting more frames can help the model better determine where it will go.

Robustness under larger delays. We additionally evaluate the performance under larger delays as might be encountered in more challenging environments. Figure 6 presents the results, where the left and right graphs depict the mAP@3D

Models	Comm. delay (ms)	mAP@3D		mAP@BEV	
		$IoU_{0.3}$	$IoU_{0.5}$	$IoU_{0.3}$	$IoU_{0.5}$
BEVSync (No extractor)	0	31.72	13.68	36.77	20.34
BEVSync	0	33.40	14.24	40.03	21.28
BEVSync (No extractor)	300	18.91	11.43	24.92	12.51
BEVSync (No compensator)	300	19.29	11.68	26.03	13.08
BEVSync	300	28.33	13.80	33.91	16.76

Table 2: Comparison with model variants without feature extracting and compensating module.

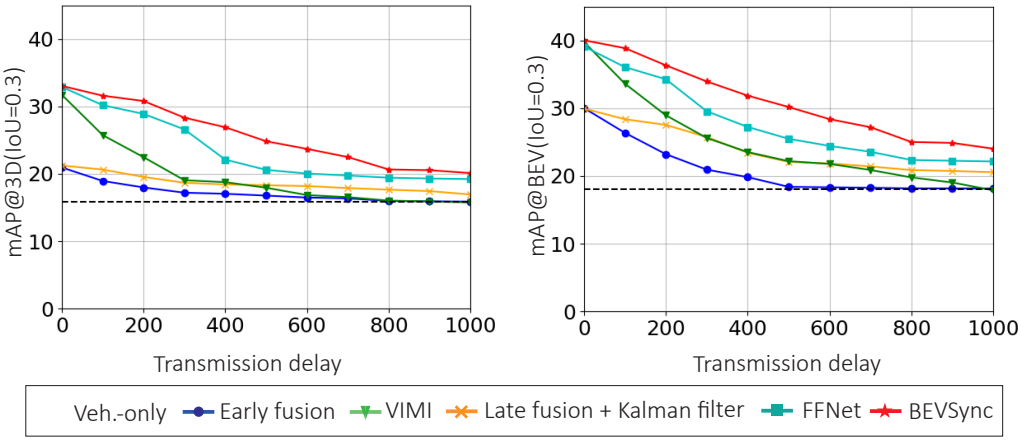

Figure 6: Performance results under larger latency.

Window size	mAP@3D		mAP@BEV	
	$IoU_{0.3}$	$IoU_{0.5}$	$IoU_{0.3}$	$IoU_{0.5}$
2	26.19	13.16	28.90	15.41
3	28.33	13.80	33.91	16.76
4	28.76	13.92	34.32	16.87
5	29.08	13.98	34.45	17.01

Table 3: Performance for various frame window sizes.

and mAP@BEV ($IoU_{0.3}$) results, respectively. From the results, a number of observations can be made:

- All cooperative methods exhibit a consistent decline in performance as the delay increases gradually from 100 ms to 1000 ms. With larger delays, the synchronous intermediate fusion method, i.e., VIMI, may perform even worse than the non-fusion method. In such cases, the significantly delayed intermediate features contribute more noise than useful information during feature fusion;
- Late fusion benefits from the use of explicit detection results, which helps to avoid the negative impact of asynchronous intermediate features that rely heavily on learning. Meanwhile, the Kalman filter ensures that it can alleviate perception errors. However, the absence of shared features ultimately leads to lower detection performance;
- FFNet shows relatively good performance under small delays below 300 ms, but its performance drops sharply as the delay increases. This sharp decline occurs because the first-order expansion struggles to compensate for nonlinear motion, especially in complex environments;
- For sufficiently large delays above 700 ms, prediction-based methods tend to stabilize. This is because dynamic vehicles become more difficult to predict, especially in such a complex intersection. While the perception of static objects, such as vehicles waiting for the green light, stays relatively constant. Therefore, the performance remains higher than vehicle-only perception.

In comparison, BEVSync deals with large delays more gracefully, exhibiting a milder decline as the delay increases. This robustness is attributed to its ability to effectively compensate for motion gaps, maintaining more accurate perception even under challenging conditions.

Conclusion

This paper introduces BEVSync, a camera-based V2I perception system, with a focus on dealing with uncertain delays. BEVSync can extract motion information from feature sequences and compensate for the motion gaps by predicting future features. Extensive experiments conducted on the DAIR-V2X real-world dataset show that BEVSync achieves superior performance compared to existing methods and robustness under various communication delays. Since our method is based on BEV perception, it can be extended to LiDAR-based cooperative fusion. In the future, we will explore heterogeneous sensor fusion in cooperative perception to balance application cost and perception performance.

Acknowledgments

This work was supported by Shenzhen Natural Science Foundation under grant 202412023000612.

References

Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. 2019. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*.

Cui, J.; Qiu, H.; Chen, D.; Stone, P.; and Zhu, Y. 2022. Coopernaut: End-to-end driving with cooperative perception for networked vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17252–17262.

Deville, M.; and Gatski, T. B. 2012. *Mathematical modeling for complex fluids and flows*. Springer Science & Business Media.

Fan, S.; Yu, H.; Yang, W.; Yuan, J.; and Nie, Z. 2023. Quest: Query stream for vehicle-infrastructure cooperative perception. *arXiv preprint arXiv:2308.01804*.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, 3354–3361. IEEE.

Justin, J.; Alexandre, A.; and Li, F.-F. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, 694–711. Springer.

Lei, Z.; Ren, S.; Hu, Y.; Zhang, W.; and Chen, S. 2022. Latency-aware collaborative perception. In *European Conference on Computer Vision*, 316–332. Springer.

Li, H.; Sima, C.; Dai, J.; Wang, W.; Lu, L.; Wang, H.; Zeng, J.; Li, Z.; Yang, J.; Deng, H.; et al. 2023a. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Li, Y.; Ge, Z.; Yu, G.; Yang, J.; Wang, Z.; Shi, Y.; Sun, J.; and Li, Z. 2023b. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 1477–1485.

Li, Y.; Ren, S.; Wu, P.; Chen, S.; Feng, C.; and Zhang, W. 2021. Learning distilled collaboration graph for multi-agent perception. *Advances in Neural Information Processing Systems*, 34: 29541–29552.

Li, Y.; Tan, G.; and Gou, C. 2024. Cascaded iterative transformer for jointly predicting facial landmark, occlusion probability and head pose. *International Journal of Computer Vision*, 132(4): 1242–1257.

Li, Y.-K.; Yu, Y.-Z.; Liu, Y.-L.; and Gou, C. 2022a. MS-GCN: Multi-stream graph convolution network for driver head pose estimation. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 3819–3824. IEEE.

Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Qiao, Y.; and Dai, J. 2022b. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, 1–18. Springer.

Liang, H.; Jagielski, M.; Zheng, B.; Lin, C.-W.; Kang, E.; Shiraishi, S.; Nita-Rotaru, C.; and Zhu, Q. 2018. Network and system level security in connected vehicle applications. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 1–7. IEEE.

Lin, Z.; Wang, L.; Ding, J.; Xu, Y.; and Tan, B. 2022. Tracking and Transmission Design in Terahertz V2I Networks. *IEEE Transactions on Wireless Communications*.

Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D. L.; and Han, S. 2023. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE international conference on robotics and automation (ICRA)*, 2774–2781. IEEE.

Ma, X.; Ouyang, W.; Simonelli, A.; and Ricci, E. 2023. 3d object detection from images for autonomous driving: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Masci, J.; Meier, U.; Cireşan, D.; and Schmidhuber, J. 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*, 52–59. Springer.

Runsheng Xu, H. X. W. S. B. Z. J. M., Zhengzhong Tu. 2022. CoBEVT: Cooperative Bird's Eye View Semantic Segmentation with Sparse Transformers. In *Conference on Robot Learning (CoRL)*.

Shi, S.; Cui, J.; Jiang, Z.; Yan, Z.; Xing, G.; Niu, J.; and Ouyang, Z. 2022. VIPS: Real-time perception fusion for infrastructure-assisted autonomous driving. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 133–146.

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28.

Wang, T.-H.; Manivasagam, S.; Liang, M.; Yang, B.; Zeng, W.; and Urtasun, R. 2020. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 605–621. Springer.

Wang, W.; Xu, H.; and Tan, G. 2024. InterCoop: Spatio-Temporal Interaction Aware Cooperative Perception for Networked Vehicles. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 14443–14449. IEEE.

Wang, Z.; Fan, S.; Huo, X.; Xu, T.; Wang, Y.; Liu, J.; Chen, Y.; and Zhang, Y.-Q. 2023. VIMI: Vehicle-Infrastructure Multi-view Intermediate Fusion for Camera-based 3D Object Detection. *arXiv preprint arXiv:2303.10975*.

Wei, S.; Wei, Y.; Hu, Y.; Lu, Y.; Zhong, Y.; Chen, S.; and Zhang, Y. 2024. Asynchrony-Robust Collaborative Perception via Bird's Eye View Flow. *Advances in Neural Information Processing Systems*, 36.

Welch, G.; Bishop, G.; et al. 1995. An introduction to the Kalman filter.

Xu, R.; Xia, X.; Li, J.; Li, H.; Zhang, S.; Tu, Z.; Meng, Z.; Xiang, H.; Dong, X.; Song, R.; et al. 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*, 13712–13722.

Xu, R.; Xiang, H.; Tu, Z.; Xia, X.; Yang, M.-H.; and Ma, J. 2022a. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *European conference on computer vision*, 107–124. Springer.

Xu, R.; Xiang, H.; Xia, X.; Han, X.; Li, J.; and Ma, J. 2022b. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 International Conference on Robotics and Automation (ICRA)*, 2583–2589. IEEE.

Yang, L.; Yu, K.; Tang, T.; Li, J.; Yuan, K.; Wang, L.; Zhang, X.; and Chen, P. 2023. BEVHeight: A Robust Framework for Vision-based Roadside 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21611–21620.

Yu, H.; Luo, Y.; Shu, M.; Huo, Y.; Yang, Z.; Shi, Y.; Guo, Z.; Li, H.; Hu, X.; Yuan, J.; et al. 2022. 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*, 21361–21370.

Yu, H.; Tang, Y.; Xie, E.; Mao, J.; Luo, P.; and Nie, Z. 2024. Flow-Based Feature Fusion for Vehicle-Infrastructure Cooperative 3D Object Detection. *Advances in Neural Information Processing Systems*, 36.

Zhao, H.; Gallo, O.; Frosio, I.; and Kautz, J. 2016. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1): 47–57.

Zheng, J.; and Liu, H. X. 2017. Estimating traffic volumes for signalized intersections using connected vehicle data. *Transportation Research Part C: Emerging Technologies*, 79: 347–362.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *International Conference on Learning Representations*.

Zhu, X.; Wang, Y.; Dai, J.; Yuan, L.; and Wei, Y. 2017. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE international conference on computer vision*, 408–417.

Zou, Z.; Zhang, R.; Shen, S.; Pandey, G.; Chakravarty, P.; Parchami, A.; and Liu, H. X. 2022. Real-time full-stack traffic scene perception for autonomous driving with roadside cameras. In *2022 International Conference on Robotics and Automation (ICRA)*, 890–896. IEEE.