Robust Asynchronous Collaborative 3D Detection via Bird's Eye View Flow

Anonymous Author(s)
Affiliation
Address
email

A Robustness to pose error

We conduct experiments to validate the performance under the impact of both asynchrony and pose error. To simulate the pose error, we add Gaussian noise $\mathcal{N}(0, \sigma_t)$ on x, y and $\mathcal{N}(0, \sigma_r)$ on θ during the inference phase, where x, y, θ are 2D centers and yaw angle of accurate global poses. Our pose noise setting follows the Gaussian distribution with a mean of 0m, a standard deviation of 0m-0.5m, a mean of 0° and a standard deviation of $0^\circ - 0.5^\circ$. And this experiment is conducted under the expectation of time interval is 300ms to simulate the time asynchrony. We compare our CoBEVFlow and other baseline methods including V2X-ViT[7], Where2comm[2] and SyncNet[4]. Table 1 shows the results on IRV2V and DAIR-V2X[8] dataset. We see that **CoBEVFlow still performs well even when both pose errors and time asynchrony appear**. CoBEVFlow consistently outperforms other methods across all noise settings on IRV2V dataset. In the case of noise levels of 0.4/0.4, our approach achieves 0.133 and 0.043 improvement over SyncNet.

Table 1: Detection performance on IRV2V and DAIR-V2X[8] dataset with pose noises following Gaussian distribution in the testing phase.

Dataset	IRV2V					DAIR-V2X				
Noise Level $\sigma_t/\sigma_r(m/\circ)$	0.0/0.0	0.1/0.1	0.2/0.2	0.3/0.3	0.4/0.4	0.0/0.0	0.1/0.1	0.2/0.2	0.3/0.3	0.4/0.4
Model / Metric	AP@0.50 ↑									
V2X-ViT	0.641	0.626	0.627	0.625	0.619	0.693	0.692	0.545	0.685	0.681
Where2comm	0.510	0.411	0.411	0.411	0.411	0.702	0.693	0.679	0.658	0.643
Where2comm+SyncNet	0.654	0.653	0.652	0.651	0.648	0.711	0.692	0.583	0.579	0.671
CoBEVFlow (ours)	**0.831**	**0.820**	**0.815**	**0.802**	**0.781**	**0.738**	**0.743**	**0.732**	**0.723**	**0.703**
Model / Metric	AP@0.70 ↑									
V2X-ViT	0.511	0.504	0.502	0.504	0.501	0.545	0.545	0.545	0.685	0.543
Where2comm	0.388	0.323	0.312	0.302	0.293	0.577	0.577	0.561	0.658	0.543
Where2comm+SyncNet	0.549	0.550	0.545	0.538	0.527	0.587	0.583	0.579	0.570	**0.567**
CoBEVFlow (ours)	**0.757**	**0.730**	**0.686**	**0.628**	**0.570**	**0.599**	**0.593**	**0.579**	**0.571**	0.560

B Visualization

Fig. 1 shows compares detection results of Where2comm[2], V2X-ViT[7], SyncNet[4], and CoBEVFlow at three asynchrony levels on the DAIR-V2X[8] dataset. The expectations of time intervals are 100, 300, and 500ms. The red boxes represent the detection results and the green boxes represent the ground-truth.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

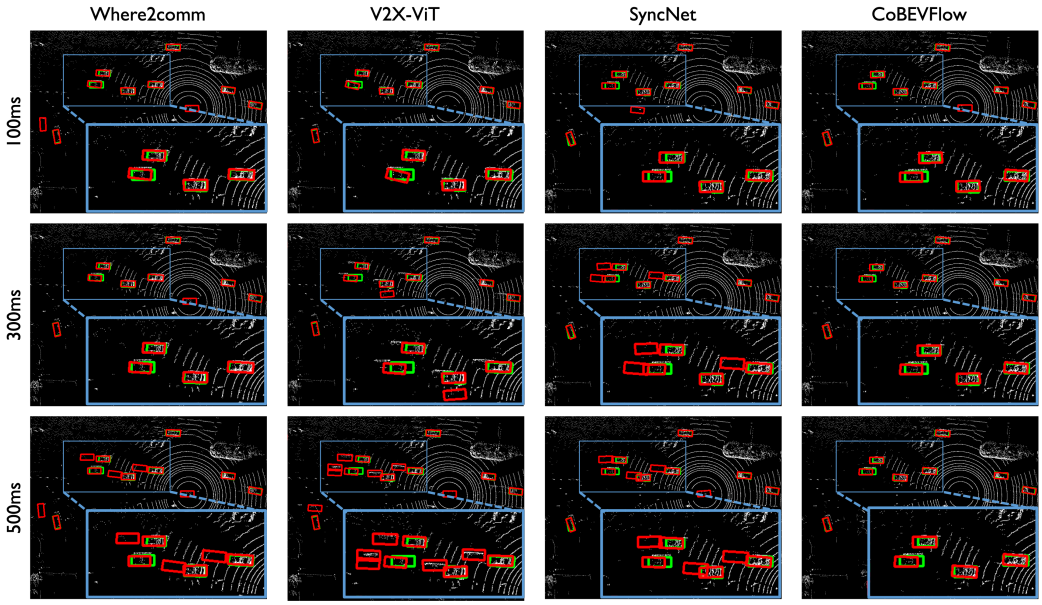
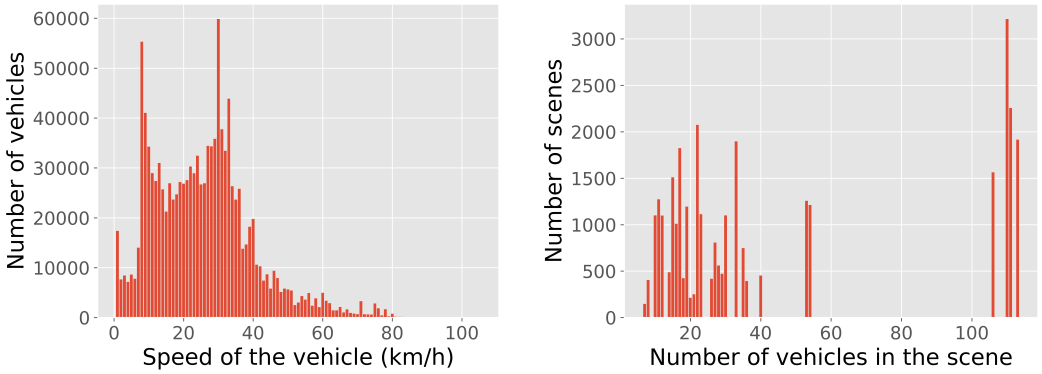


Figure 1: Visualization of detection results for Where2comm, V2X-ViT, SyncNet, and our CoBEVFlow with the expectation of time intervals are 100, 300, and 500ms on DAIR-V2X dataset. Red and green boxes denote detection results and ground-truth respectively.



(a) Vehicle speed distribution.

(b) Number of vehicles in the scene distribution.

Figure 2: Data distribution of IRV2V dataset. (a) shows the speed distribution of moving vehicles; (b) shows the vehicle numbers distribution.

C IRegular V2V (IRV2V)

To facilitate the research on asynchrony for collaborative perception, we use CARLA [1] (under MIT license) to simulate IRregular V2V (IRV2V) dataset, which is the first collaborative perception dataset with multiple asynchronies.

Asynchronous data collection. The number of collaborative vehicles in a scene ranges from 2 to 5. Each collaborative vehicle is equipt with 4 cameras for 360° view, a 32-channel LiDAR, and GPS/IMU sensors. The ideal sample interval of the sensor is 100ms. Due to different asynchronous factors, collaborative messages have asynchronous timestamps. There is a time offset $\delta_s \sim \mathcal{U}(-50, 50)$ms at the sampling starting point of non-ego vehicles. And all non-ego vehicles' collaborative messages are sampled with time turbulence $\delta_d \sim \mathcal{U}(-10, 10)$ms. Sensing information at each timestamp of each agent contains 4 camera images with resolution 600×800, and 32-channel LiDAR points.

Data size. Assuming the model requires the use of information from the past 10 frames, our dataset consists of a total of 8,449 collaborative samples, which include 8,449 point cloud inputs and 33,796

Table 2: Performance of CoBEVFlow and other baseline methods under the expectation of time interval from 0 to 500ms on IRV2V dataset. CoBEVFlow outperforms all the baseline methods and shows great robustness under any level of asynchrony.

Expectation of interval (ms)	0	100	200	300	400	500
Model / Metric	AP@0.50 ↑					
Single	0.647					
Late Fusion	0.828	0.638	0.478	0.371	0.324	0.308
V2VNet	0.811	0.747	0.710	0.663	0.626	0.591
V2X-ViT	0.781	0.737	0.692	0.641	0.598	0.575
DiscoNet	0.742	0.728	0.704	0.673	0.647	0.625
Where2comm	0.864	0.758	0.609	0.510	0.455	0.431
Where2comm+SyncNet	0.864	0.721	0.672	0.654	0.649	0.625
CoBEVFlow (ours)	**0.864**	**0.841**	**0.834**	**0.831**	**0.812**	**0.815**
Model / Metric	AP@0.70 ↑					
Single	0.535					
Late Fusion	0.751	0.385	0.285	0.235	0.219	0.223
V2VNet	0.744	0.607	0.540	0.480	0.439	0.408
V2X-ViT	0.630	0.577	0.545	0.511	0.489	0.479
DiscoNet	0.624	0.612	0.586	0.559	0.537	0.519
Where2comm	0.827	0.613	0.458	0.388	0.362	0.359
Where2comm+SyncNet	0.827	0.602	0.555	0.549	0.545	0.536
CoBEVFlow (ours)	**0.864**	**0.781**	**0.761**	**0.757**	**0.714**	**0.687**

RGB images. We have split the dataset into training, validation, and testing sets, which contain 5,445, 994, and 2,010 samples, respectively.

Data analysis. Figure 2 presents some statistical analysis results regarding the IRV2V dataset. The IRV2V dataset contains a total of 1,564,033 vehicles, with an average of 48.302 vehicles per scene. It should be noted that the figure only displays the distribution of vehicles with speeds greater than 1 km/h. Considering real-world scenarios, there are around 1,203,793 moving vehicles in the dataset. Plot (a) illustrates the distribution of moving vehicles with different speeds across all samples, ranging from 1 to 105 km/h, with an average speed of 25.586 km/h, which achieves around 15km/h faster compared to the majority of vehicles in the V2X-Sim[5]. Plot (b) shows the distribution of the total number of vehicles per sample in the dataset, with the maximum number of vehicles being 113.

D Detailed information about experimental settings

Implement details. We conduct experiments on LiDAR-based part of IRV2V and DAIR-V2X[8] dataset. Our feature encoder is PointPillars[3] based. And our backbone follows the setting in CoAlign[6]. The difference is that we change the fusion method from self-attention to max-fusion. We conduct training for a total of 60 epochs, starting with an initial learning rate of 2e-3. Subsequently, at the 10th and 20th epochs, the learning rate decreases to 10% of its previous value. For IRV2V dataset, we set the lidar range as $x \in [-140.8, +140.8]$m, $y \in [-40, +40]$m. The voxel size is $h = w = 0.4$m. The feature map's size is $H = 200, W = 704$. For DAIR-V2X dataset, we set the lidar range as $x \in [-100.8, +100.8]$m, $y \in [-40, +40]$m. The voxel size is $h = w = 0.4$m. The feature map's size is $H = 200, W = 504$.

Communication volume. Our communication volume is the same as Where2comm[2]. For CoBEVFlow, we control the communication volume by adjusting the maximum number of generated ROIs. Specifically, the average number of voxels contained in each ROI region is 40, and we limit the maximum number of generated ROIs to $K_{\|\mathcal{R}\|}$. Correspondingly, we modify the information exchange in Where2comm to include the top $40 \times K_{\|\mathcal{R}\|}$ blocks based on their scores on the spatial confidence map. In practical scenarios, the actual communication volume is influenced by factors such as the feature dimension and floating-point precision. To simplify the expression, we uniformly represent the communication volume using the logarithm to the base 2 of the voxel count. The

Table 3: Performance of CoBEVFlow and other baseline methods under the expectation of time interval from 0 to 500ms on DAIR-V2X[8] dataset. CoBEVFlow outperforms all the baseline methods and shows great robustness under any level of asynchrony.

Expectation of interval (ms)	0	100	200	300	400	500
Model / Metric	AP@0.50 ↑					
Single	0.674					
Late Fusion	0.709	0.664	0.616	0.600	0.595	0.593
V2VNet	0.626	0.623	0.622	0.617	0.612	0.608
V2X-ViT	0.725	0.714	0.704	0.693	0.681	0.681
DiscoNet	0.645	0.630	0.620	0.606	0.597	0.590
Where2comm	0.807	0.739	0.603	0.698	0.686	0.676
Where2comm+SyncNet	0.807	0.719	0.706	0.695	0.687	0.679
CoBEVFlow (ours)	**0.807**	**0.765**	**0.749**	**0.738**	**0.728**	**0.726**
Model / Metric	AP@0.70 ↑					
Single	0.587					
Late Fusion	0.538	0.479	0.467	0.464	0.466	0.465
V2VNet	0.556	0.555	0.554	0.552	0.548	0.548
V2X-ViT	0.556	0.553	0.550	0.545	0.541	0.541
DiscoNet	0.553	0.537	0.530	0.523	0.519	0.518
Where2comm	0.662	0.603	0.587	0.577	0.569	0.566
Where2comm+SyncNet	0.662	0.602	0.588	0.587	0.584	0.580
CoBEVFlow (ours)	**0.662**	**0.621**	**0.601**	**0.599**	**0.592**	**0.588**

communication volume is

$$\log_2\left(40 \times K_{\|\mathcal{R}\|}\right), \tag{1}$$

where $K_{\|\mathcal{R}\|}$ is the maximum number of generated ROIs, and 40 is the average number of voxels in each ROI.

E Benchmarks

We conduct extensive experiments on current collaborative perception methodologies. Table 2 and Table 3 present the detection performance under the expectation of time interval from 0 to 500ms on IRV2V and DAIR-V2X[8] respectively, which correspond to the numerical results shown in Figure 4 in the main text. We see that CoBEVFlow consistently achieves significant improvements over previous methods on both datasets and the leading gap is bigger when the expectation of the time interval is higher.

References

[1] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[2] Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *arXiv preprint arXiv:2209.12836*, 2022.

[3] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 12697–12705. Computer Vision Foundation / IEEE, 2019.

[4] Zixing Lei, Shunli Ren, Yue Hu, Wenjun Zhang, and Siheng Chen. Latency-aware collaborative perception. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 316–332. Springer, 2022.

[5] Yiming Li, Dekun Ma, Ziyan An, Zixun Wang, Yiqi Zhong, Siheng Chen, and Chen Feng. V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving. *IEEE Robotics Autom. Lett.*, 7(4):10914–10921, 2022.

[6] Yifan Lu, Quanhao Li, Baoan Liu, Mehrdad Dianati, Chen Feng, Siheng Chen, and Yanfeng Wang. Robust collaborative 3d object detection in presence of pose errors. *CoRR*, abs/2211.07214, 2022.

[7] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 107–124. Springer, 2022.

[8] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, and Zaiqing Nie. DAIR-V2X: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 21329–21338. IEEE, 2022.