A Appendix

A.1 Impact of training sequence length on nuScenes.

In Sec. 4.5 **Impact of training sequence length** of main paper, we have showed the results on Waymo Open Dataset [2]. This section we show the results on nuScenes [1], as depicted in Fig. 7, which depicts the relationship between frames and performance for the proposed INT-Pillar and CenterPoint-Pillar [3] model. Looking at the left panel first, INT performance improves as the number of frames increases, although there is saturation after a certain point; as for the right panel, the time consumed by INT is slightly higher than that of single-frame CenterPoint, but it does not increase with the number of frames.

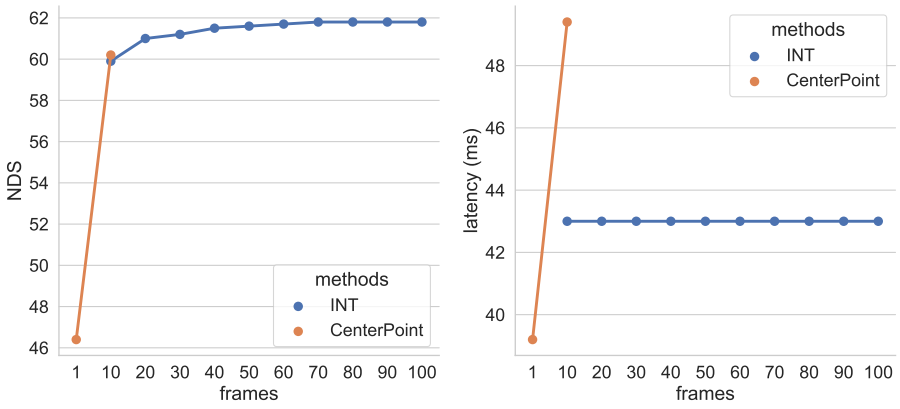

Fig. 7. Impact of training frame length on nuScenes *val* set. While CenterPoint's performance improves as the number of frames grows, the latency also increases dramatically. On the other hand, our INT keeps the same latency while increasing performance.

A.2 Impact of fusion methods for image-style data.

As indicated in **images-style fusion** of Sec. 3.2 in the main paper, we propose four fusion algorithms for image-style data. This section examines the performance and latency of these fusion methods. As shown in Table 7, the *Concat* and *GRU-like* solutions provide better performance, whereas *Add* and *Max* require less amount of latency. Taking both performance and efficiency into account, we choose the *Concat* as the default temporal fusion method for image-style data fusion experiments in both the main paper and the appendix. To accelerate experiments in Table 7, we use 1/4 training sequences for both Waymo Open Dataset and nuScenes, and we use 1/4 validation sequences for Waymo and the whole validation sequences for nuScenes.

Table 7. Impact of different fusion methods for image-style data on Waymo Open Dataset [2] and nuScenes [1] *val* set. INT-Pillars is employed in these experiments. To accelerate experiments, only 1/4 training sequences are used.

Fusion	nuScenes			Waymo				
methods	mAP↑	NDS↑	latency↓	VEL↑	PED↑	CYC↑	mAPH↑	latency↓
Baseline	26.8	35.5	**39.2**	59.1	48.6	41.1	49.6	**57.4**
Add	31.5	42.0	39.6	58.9	54.6	54.8	56.1	58.3
Concat	**33.7**	**45.3**	40.5	**59.8**	58.3	62.5	**60.2**	59.0
Max	30.3	42.9	39.6	58.7	54.9	55.1	56.2	58.4
GRU-like	31.9	44.8	41.0	58.1	**58.9**	**63.8**	**60.2**	59.7

A.3 Impact of Dynamic Training Sequence Length (DTSL).

Dynamic Training Sequence Length (DTSL) in Sec 3.1 of main paper explains why DTSL is required for INT training, and this part confirms DTSL's function. Training a 100-frame INT-Pillars network on nuScenes improves the NDS with DTSL by 2.5% compared to no DTSL, as demonstrated in Table 8.

Table 8. Impact of Dynamic Training Sequence Length (DTSL) on nuScenes *val* set.

	mAP	NDS
w/o DTSL	50.2	59.3
w/ DTSL	**52.3**	**61.8**

A.4 A typical example of INT-Voxel framework.

The point-style data in MB is initialized as a $B \times N \times C$ tensor, and image-style data is a $B \times C \times H \times W$ tensor, where B is batch size, C is feature dimension, N is points number (50,000 in the paper), H/W is the feature map size. Here we take INT-Voxel as an example, and Fig. 8 shows the positions of fusion modules, which are commonly described in **Fusion settings** of Sec. 4.2.

A.5 Explanation on performance saturation.

From Fig. 1 and 7, we can see a performance saturation while frames increases. The reason for this phenomenon is that: (1) Point-style data in MB has a limited number of points (current 50,000), so points that are too old do not remain in the MB any more. Enlarging the point MB or improving the update policy may ease the problem. (2) Image-style data is exponentially decaying in current fusion methods, so the contribution of too old information is almost negligible. Improving the fusion methods may alleviate this problem.

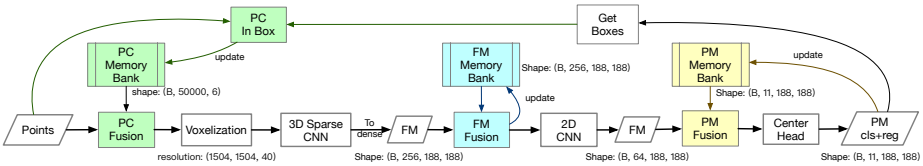

Fig. 8. Pipeline of INT-VoxelNet on Waymo. PC, FM and PM represent Point Cloud, Feature Map and Prediction Map, respectively.

A.6 Latency details for CenterPoint in Table 1

As illustrated in **Latency settings** of Sec. 4.2, we put the voxelization to GPU. The voxelization implementation follows OpenPCDet for E-Pillar, and Ceneter-Point for E-Voxel. Table 9 and 10 show the detailed cost of each module in CenterPoint [3].

Table 9. Detailed latency of CenterPoint-Pillar on Waymo Open Dataset.

latency(ms)	copy to GPU	voxelization	MLP + scatter_to_bev	2D CNN + post	total
E-Pillar 1f	0.6	1.8	5.5	49.8	57.7
E-Pillar 2f	1.4 (+0.8)	11.4 (+9.6)	14.6 (+9.1)	50.4 (+0.6)	77.8 (+20.1)

Table 10. Detailed latency of CenterPoint-Voxel on Waymo Open Dataset.

latency(ms)	copy to GPU	voxelization	sparse conv	2D CNN + post	total
E-Voxel 1f	0.6	1.9	43.6	25.6	71.7
E-Voxel 2f	1.3 (+0.7)	3.7 (+1.8)	60.1 (+16.5)	25.8 (+0.2)	90.9 (+19.2)

A.7 Prediction Visualizations

We compared the prediction results of 2-stage CenterPoint-Voxel and INT-Voxel on Waymo Open Dataset *val* set in Figure 9.

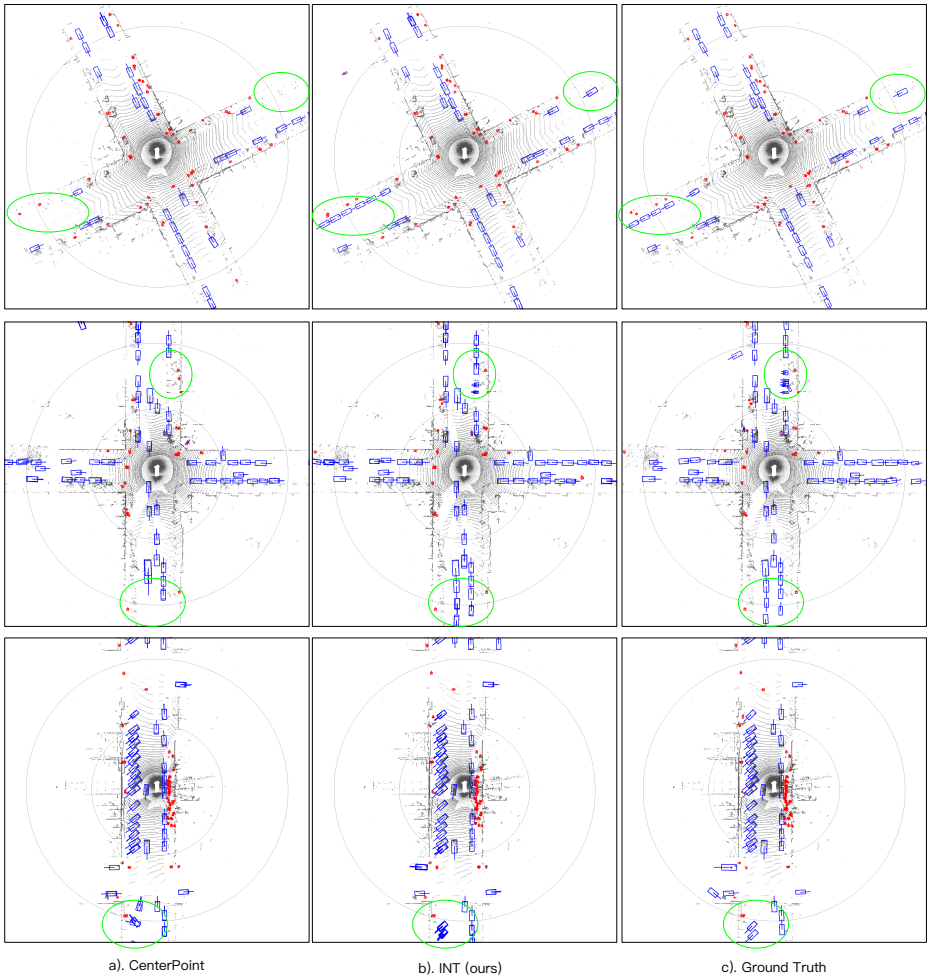

Fig. 9. The detection results on Waymo Open Dataset *val* set. The bounding boxes of VEHICLE, PEDESTRIAN and CYCLIST are in the color blue, red and magenta respectively. As pointed out in green circles, INT is obviously more advantageous in detecting long-range targets because it has more historical information.

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11621–11631 (2020) 1, 2
2. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2446–2454 (2020) 1, 2
3. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021) 1, 3