Appendix

A. Motion Forecasting Model

In Sec. 4.1, we mentioned that a forward and a backward MultiPath++ are trained for generating MoDAR points. In this section, we provide more details about training and evaluating the MultiPath++ models.

Close the domain gap between WOMD and WOD. When constructing Waymo Open Motion Dataset (WOMD) and training the motion forecasting models, people intentionally mines the interesting trajectories, such as the following pairwise cases: merges, lane changes, unprotected turns, intersection left turns, intersection right turns, pedestrian-vehicle intersections, cyclist-vehicle in intersections, intersections with close proximity, and intersections with high accelerations [9]. Different from WOMD, most trajectories in Waymo Open Dataset (WOD) are less interesting: cars are usually parked or moving with a constant velocity [39].

Therefore, to close the trajectories sampling gap, when training MultiPath++ [42] on WOMD, we change the original sampling strategy to a dense sampling strategy, which uses all tracks for training instead of sampling the interesting tracks. Tab. 6 shows the performance comparison when training with different sampling strategies and testing on different dataset. When training with the original WOMD, the results are better on the original WOMD validation set. This is because both original WOMD training and validation sets sample the interesting trajectories. However, when training with the dense sampled WOMD, the results on WOD validation set is better. For example, the Average Displacement Error (ADE) is reduced from 1.83 to 1.17 on WOD validation set, by changing the original sampling strategy to the dense sampling strategy.

Forward and Reverse Motion Forecasting Models. Besides the past point cloud sequence, the offboard detection set up also takes the information from the future point cloud sequence. To propagate future object information to the current frame, we train a reverse motion forecasting model. Specifically, we prepared the reversed training set based on the WOMD, and also prepared the reversed WOD for generating MoDAR points. When preparing the training set, we resplit all 91 frame trajectories to 11 frame input track and 80 frame ground truth track as training label. Different from the forward dataset, the backward dataset take the last 11 frames as the input, and guide the model to predict the first 80 frame trajectories. Besides, we reverse the velocity vector of each object accordingly. When preparing the WOD inference set, instead of using the original timestamp T_{original}, we assign a virtual (negative) timestamp T_{virtual} for each detection. The timestamp will be normalized before feeding into the motion forecasting models. After we reassign the virtual timestamp to each detection box, we proceed the tracking and motion forecasting as forward counterpart. Finally, we convert the virtual timestamp back to the original timestamp by $T_{\text{original}} = -T_{\text{virtual}} + c$. Finally, we compare the forward and reverse motion forecasting model in Tab. 7, showing that the reverse model is as good as (or even slightly better than) the forward model.

B. Sharing 3D Detectors for LiDAR-MoDAR detection and Motion Forecasting

As we illustrated in Fig. 3, our pipeline needs two 3D detection models: (1) a LiDAR 3D object detection model to prepare the input tracks for motion forecasting model, and (2) a LiDAR-MoDAR 3D object detection model for the final detection results. Although the architecture of these two models are the same, their weights are trained separately. In this section, we discuss the possibility to consolidate these two models, *i.e.*, use a shared model with the same weights for both LiDAR-MoDAR detection and motion forecasting input. We explore the impact to performance if the LiDAR-MoDAR detector model takes the MoDAR points generated by itself (MoDAR from its detection boxes).

The results are shown in Tab. 8. We use the 1-frame SWFormer as the baseline model (#W1), and use an in-house motion forecasting model that is slightly stronger than MultiPath++ reported in the main paper. We observe that when the MoDAR points are generated differently during training and validation, the performance will drop. For example, when training and evaluating with MoDAR points generated by #W1, the L2 3D mAPH is 74.5. However, if evaluating this model with the MoDAR points generated by #W2, the performance drops by 3.7 (from 74.5 to 70.8) L2 3D mAPH, even though the MoDAR points from #W2 is more accurate than #W1. We also observe that retraining the detector model again helps reduce this gap. Specifically, for model #W3, when training with MoDAR points from #W2 and evaluate with MoDAR points from #W3, the performance only drops by 1.4 (from 74.8 to 73.4) L2 3D mAPH, which is smaller than the 3.7 L2 3D mAPH gap for model #W2. Therefore, we hypothesize iterative training can potentially mitigate this problem. However iterative re-training would make the training process more complex. As a future work, we can explore other techniques (such as adding noise to MoDAR points during training, or generating MoDAR points on-the-fly) to improve the robustness of taking MoDAR points from different models.

C. Implementation Details of Detectors

For CenterPoints and SWFormer LiDAR-only models, we apply data augmentations during training following the

Training Set	Validation Set	ADE	FDE	minADE	minFDE
WOMD (Original)	WOMD Val. WOD Val.	3.34 1.83	10.2 9.22	1.40 0.82	4.01 3.67
WOMD (Dense)	WOMD Val. WOD Val.	3.61 1.17	11.1 5.45	1.42 0.55	3.74 2.27

Table 6. Compare different sampling strategies when training the motion forecasting model, MultiPath++, on Waymo Open Motion Dataset (WOMD). We test the trained model on the validation set of both WOMD and WOD. We observe that the dense sampling strategy leads to lower error on WOD validation set. ADE, FDE, minADE, and minFDE are evaluation metrics (lower is better) for the motion forecasting task.

	ADE	FDE	minADE	minFDE
Forward MP++	1.17	5.45	0.55	2.27
Reverse MP++	1.11	4.70	0.51	1.76

Table 7. Compare the performance of the forward and the reverse motion forecasting models. We observe that the reverse motion forecasting model is as good as (or even slightly better than) the forward one.

Model ID	Model	MP++ inputs		Veh. L1 3D		Veh. L2 3D		Ped. L1 3D		Ped. L2 3D		**L2 3D mAPH**
		@train	@eval	AP	APH	AP	APH	AP	APH	AP	APH	
#W1	SWFormer [40]	-	-	77.0	76.5	68.3	67.9	80.9	72.3	72.3	64.4	66.2
#W2	+MoDAR	#W1	#W1	83.2	82.6	75.9	75.3	84.0	80.5	76.7	73.7	74.5 (+8.3)
			#W2	80.2	79.6	73.2	72.6	80.0	76.0	72.7	69.0	70.8 (+4.6)
#W3	+MoDAR	#W2	#W2	83.6	83.0	76.4	75.9	84.4	80.8	77.1	73.7	74.8 (+8.6)
			#W3	82.3	81.7	75.3	74.8	82.6	79.0	75.3	71.9	73.4 (+7.2)

Table 8. The performance comparison when generating MoDAR points by different models during evaluation. We observe that using different model to (feed to MP++ as inputs to) generate MoDAR points during training harms the final detection performance. Iterative training can mitigate this performance drop.

original SWFormer implementation [40]: randomly rotating the world by yaws, randomly flipping the world along y-axis, randomly scaling the world, and randomly dropping points. For the MoDAR-LiDAR fusion model, we first combine MoDAR and LiDAR points together, and then apply data augmentation to the fused point cloud. Note that these data augmentation only change the 3D coordinate of points, but keep the point feature unchanged.

D. MoDAR-LiDAR Fusion

Late fusion implementation details. We implemented the MoDAR-LiDAR late fusion by a weighted box fusion strategy [37]. Since LiDAR signal shows better performance, we set the weight of the LiDAR predictions as 0.9 and set the weight of MoDAR predictions as 0.1. We finally keep top 300 boxes sorted by the confidence scores.

Fusing MoDAR from different frames. In Tab. 5, we directly get the detection results from MoDAR. In this section, we introduce more details about how to generate detection boxes from MoDAR. As we mentioned, each MoDAR point represents a predicted 3D box. The location of the MoDAR point is the predicted center of the object, while the object size is stored in the MoDAR point feature. Therefore, we have a large number of 3D boxes predicted by different motion forecasting models. We also use the weighted box fusion strategy [37] to fuse these boxes together. Specifically, the boxes generated by recent predictors will have higher weights. Take the 5×2 predictions in Tab. 10 as an example: we take the boxes from the closest 5 past and 5 future predictors, with the weight of 1.0, 0.8, 0.6, 0.4, and 0.2. The results are shown in Tab. 10, and we call this method as late fusion because it is a box-level fusion strategy. We observe that using the closest 5 past and 5 future predictors achieves the best results. Fusing boxes from more predictors does not help because the long-term predictors predict less accurate boxes.

On the other hand, in this section, we also explore the

Model	Frame	Offline	Veh. L1 3D		Veh. L2 3D		Ped. L1 3D		Ped. L2 3D		**L2 3D mAPH**
	[-p, +f]	Method?	AP	APH	AP	APH	AP	APH	AP	APH	
MVF++ [33]†	[-4, 0]		79.7	-	-	-	81.8	-	-	-	-
+3DAL [33]	[-∞, ∞]	✓	84.5	84.0	75.8	75.3	82.9	79.8	73.6	70.8	73.1
LidarAug [16]*	[-2, 0]		81.4	80.9	73.3	72.8	84.1	80.4	76.5	72.9	72.9
+MoDAR	[-91, 91]	✓	**86.3**	**85.8**	**79.5**	**79.0**	**87.7**	**84.6**	**81.1**	**78.0**	**78.5**

Table 9. MoDAR based on a stronger detection LidarAug [16]. †: ensemble with 10 times test-time-augmentation. *: our re-implementation.

Number of	Fusion	Veh. L2		Ped. L2	
Predictions	Method	AP	APH	AP	APH
1 × 2	Late	65.6	65.0	**69.6**	63.7
5 × 2	Late	**67.4**	**66.8**	**69.6**	**63.8**
10 × 2	Late	67.1	66.5	62.9	57.6
15 × 2	Late	66.1	65.6	52.5	48.2
5 × 2	Early	70.3	68.6	74.5	70.2
10 × 2	Early	70.4	69.3	75.5	71.4
20 × 2	Early	**71.2**	**70.5**	**75.8**	**72.0**
40 × 2	Early	70.9	70.2	74.8	70.8
80 × 2	Early	69.0	68.4	74.3	70.3

Table 10. Fusing MoDAR from different predictors. We compare the early and the late fusion strategies, and explore to fuse different number of predictions (”×2” means fusing the predictions from both past and future predictors).

LiDAR	MoDAR	Latency (ms)
3 frames	✗	172
5 frames	✗	247
7 frames	✗	276
3 frames	✓	221

Table 11. Latency comparison between LiDAR-MoDAR fusion and LiDAR-only models. The latency of LiDAR-MoDAR fusion model is between 3-frame and 5-frame LiDAR-only models.

early fusion strategy to fuse the MoDAR points from different predictors. Specifically, we put all MoDAR points (but no LiDAR points) as the input of the 3D object detection model. According to the results shown in Tab. 10, early fusion is more effective than the late fusion, and it can take the MoDAR points from more predictors even if the predictors are not close to the current frame. For example, our best MoDAR-only early fusion model achieves 70.5 Vehicle L2 APH and 72.0 Pedestrian L2 APH, which is already better than the LiDAR-only model with 69.7 Vehicle L2 APH and 70.1 Pedestrian L2 APH (shown in Tab. 5 in the main paper).

E. Latency

In this section, we compared the latency of our MoDAR-LiDAR fusion detection model with the LiDAR-only detection model, based on our re-implementation of the 3-frame SWFormer. We measure the latency using an in-house GPU. The average latency of our baseline 3-frame SWFormer is 172ms per frame. Note that this latency is considerably higher than the 20ms latency reported in the original SWFormer paper [39], which is mainly because our research-oriented implementation is not optimized with respect to the fused transformer kernels [39] and the hardware devices are different. However, the comparisons below are under the same hardware devices and under the same implementation.

We measure the latency of three LiDAR-only models, the LiDAR-only SWFormer with 3-, 5-, or 7-frame LiDAR point cloud input, and our MoDAR-LiDAR fusion model that takes 3-frame LiDAR point cloud and MoDAR points from 160 predictors. The latency are shown in Tab. 11. As we can see, the latency of our LiDAR-MoDAR fusion detector is between 3-frame and 5-frame LiDAR-only model, indicating the marginal computational complexity for using MoDAR points. Note that for the onboard system, we can cache the motion forecasting signal with little overhead, because motion forecasting is usually an important module of an autonomous driving system. For the offboard application, the latency of our motion forecasting model MultiPath++ is 217 ms, which is similar to the detection model. Compared with 3DAL [33] that takes 15min to process a 200-frame sequence, our offboard system takes about $221 + 172 + 217 * 2 = 827$ms to process a frame, *i.e.*, 3 minutes per 200-frame sequence, which is about 5× faster than 3DAL. As future work, by implementing customized kernels and optimizing network architectures, we expect to further reduce the latency.

Model	mAPH	Veh AP/APH 3D		Ped AP/APH 3D	
	L2	L1	L2	L1	L2
SWFormer	73.4	82.9/82.5	75.0/74.7	82.1/78.1	75.9/72.1
+MoDAR	**78.9**	88.0/87.5	81.2/80.8	85.8/82.5	80.2/77.0

Table 12. Compare WOD test set results with our baseline method, SWFormer [40]. mAPH/L2 is the offical ranking metric on the WOD leaderboard.

F. Results on the WOD Test Set

Tab. 12 shows vehicle and pedestrain detection results comparison with our baseline, SWFormer [40]. We observe a similar improvement compared with the results on validation set. This further indicates the effectiveness of our proposed method.

G. Generalizing to Stronger Detectors

To show our method generalizes, we use LidarAug-SWFormer [16] as a stronger baseline. Shown in Tab. 9, adding MoDAR leads to consistent gains and significantly outperforms previous methods. For example, we achieves 78.5 L2 3D mAPH, which is significantly better 3DAL by 5.4 L2 3D mAPH.