

MoDAR: Using Motion Forecasting for 3D Object Detection in Point Cloud Sequences

Yingwei Li* Charles R. Qi* Yin Zhou Chenxi Liu Dragomir Anguelov
Waymo LLC

Abstract

Occluded and long-range objects are ubiquitous and challenging for 3D object detection. Point cloud sequence data provide unique opportunities to improve such cases, as an occluded or distant object can be observed from different viewpoints or gets better visibility over time. However, the efficiency and effectiveness in encoding long-term sequence data can still be improved. In this work, we propose MoDAR, using motion forecasting outputs as a type of virtual modality, to augment LiDAR point clouds. The MoDAR modality propagates object information from temporal contexts to a target frame, represented as a set of virtual points, one for each object from a waypoint on a forecasted trajectory. A fused point cloud of both raw sensor points and the virtual points can then be fed to any off-the-shelf point-cloud based 3D object detector. Evaluated on the Waymo Open Dataset, our method significantly improves prior art detectors by using motion forecasting from extra-long sequences (e.g. 18 seconds), achieving new state of the arts, while not adding much computation overhead.

1. Introduction

3D object detection is a fundamental task for many applications such as autonomous driving. While there has been tremendous progress in architecture design and LiDAR-camera sensor fusion, occluded and long-range object detection remains a challenge. Point cloud sequence data provide unique opportunities to improve such cases. In a dynamic scene, as the ego-agent and other objects move, the sequence data can capture different viewpoints of objects or improve their visibility over time. The key challenge though, is how to efficiently and effectively leverage sequence data for 3D object detection.

Existing multi-frame 3D object detection methods often fuse sequence data at two different levels. At scene level, the most straightforward approach is to transform point clouds of different frames to a target frame using known

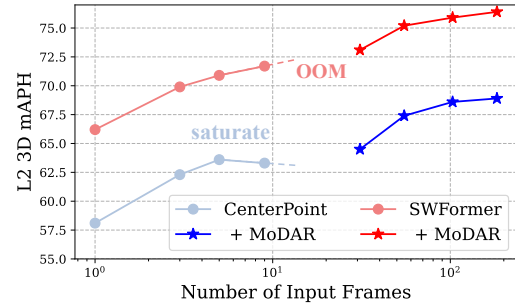


Figure 1. **3D detection model performance vs. number of input frames.** Naively adding more frames to existing methods, such as CenterPoint [59] and SWFormer [40], quickly plateaus the gains while our method, MoDAR, scales up to many more frames and gets much larger gains. L2 3D mAPH is computed by averaging vehicle and pedestrian L2 3D APH.

ego motion poses [3, 40, 55, 59]. Each point can be decorated with an extra time channel to indicate which frame it is from. However, according to previous studies [7, 33] and our experiments shown in Fig. 1, it is difficult to further improve the detection model by including more input frames due to its large computation overhead as well as ineffective temporal data fusion at scene level (especially for moving objects). On the other side, 3D Auto Labeling [33] and MPPNet [7] propose to aggregate longer temporal contexts at object level, which is more tractable as there are much less points from objects than those from the entire scenes. However, they also fail to scale up temporal context aggregation to long sequences due to efficiency issues or alignment challenges.

In our paper, we propose to use motion forecasting to propagate object information from the past (and the future) to a target frame. The output of the forecasting model can be considered another (virtual) sensor modality to the detector model. Inspired by the naming of the LiDAR sensor, we name this new modality *MoDAR*, Motion forecasting based Detection And Ranging (see Fig. 2 for an example).

Traditionally 3D object detection is a pre-processing step for a motion forecasting model, where the detector boxes are either used as input (for past frames) or learning targets (for future frames). In contrast, we use motion forecasting outputs as input to LiDAR-MoDAR multi-modal 3D object

*equal contributions

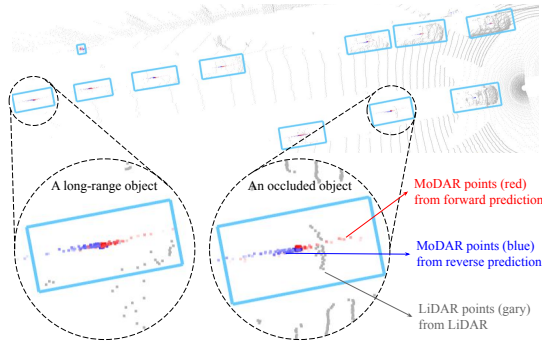


Figure 2. **3D object detection from MoDAR and LiDAR points.** MoDAR points (red and blue) are predicted object centers with extra features such as sizes, semantic classes and confidence scores. Compared to LiDAR-only detectors, a multi-modal detector taking both LiDAR (gray) and MoDAR points can accurately recognize occluded and long-range objects that have few observed points.

detectors. There are two major benefits of using a MoDAR sensor for 3D object detection from sequence data. First, motion forecasting can easily transform object information across very distant frames (8 seconds or longer). Such propagation is especially robust to occlusions as the forecasting models do not assume successful tracking for trajectory forecasting. Second, considering forecasting output as another sensor data source for 3D detection, it is a lightweight sensor modality, making long-term sequence data processing possible without much computation overhead.

Specifically, in MoDAR, we represent motion forecasting output at the target frame as a set of virtual points (named as MoDAR points), one for each object from a way-point on a forecasted trajectory. The predicted object location is the 3D coordinate of the virtual point, while additional information (such as object type, size, predicted heading, and confidence score) is encoded into the virtual point features. Each virtual point is appended with a time channel to indicate the context frames it uses for the motion forecasting. For a target frame, we can use forecasted outputs from multiple context frames easily through a union of corresponding virtual points. In an offboard/offline detection setup, we can use both forward prediction and reverse prediction (use future frames as input to the forecasting model) to combine information from the past and the future. For detection, we fuse the raw sensor points (from LiDARs) and the virtual points (from forecasting), and feed them to any off-the-shelf point cloud based 3D detector.

In experiments, we use a MultiPath++ [42] motion forecasting model trained on the Waymo Open Motion Dataset [9] to generate MoDAR points from past 9 seconds for online detection; and from past and future 18 seconds for offline detection. With minimum changes, we adapt CenterPoints [59] and SWFormer [40] detectors for LiDAR-MoDAR 3D object detection.¹ Evaluated on the Waymo

¹Although we experiment with point-cloud based detectors, MoDAR

Open Dataset [39], we show that adding MoDAR significantly improves detection quality, improving CenterPoints and SWFormer by 11.1 and 8.5 mAPH respectively; and it especially helps detection of long-range and occluded objects. Using MoDAR with a 3-frame SWFormer detector, we have achieved state-of-the-art mAPH on the Waymo Open Dataset. We further provide extensive ablations and analysis experiments to validate our designs and show impacts of different MoDAR choices.

2. Related Work

3D object detection on point clouds. Most work focuses on using single-frame input. They can be categorized to methods using different representations such as voxels or pillars [8, 14, 15, 17, 36, 38, 45, 46, 50, 52, 57, 63], point clouds [26, 31, 34, 35, 53, 54], range images [2, 18, 24, 41], etc. Liu et al. [21] did a review to put those methods in a unified framework. Among those methods, CenterPoint [59] using anchor-free detection heads [62] becomes one of the popular single-stage 3D detectors. On the other hand, more recent methods explore to use transformers for 3D detection [10, 40]. For example, SWFormer [40] used sparse window based transformers to achieve new state-of-the-art performance. In this work we use these two representative detector architectures for our experiments.

Multi-frame 3D object detection. Early multi-frame 3D detectors aggregate features from different frames using convolution layers [23]. More recent methods use a simple point concatenation strategy, which transform short-term point cloud sequences into the same coordinate (using ego-car poses) and then feed the merged points to deep networks [11, 19, 40, 41, 59]. They usually use point cloud sequences that are up to 5 frames due to memory/computation costs. Another drawback of point concatenation is that it cannot align moving objects. Later methods explore how to use more frames and model alignment at the intermediate feature level. For example, 3D-MAN [55] uses an attention module to align different frames while MPPNet [7] designs both intra-group feature mixing and inter-group feature attention. However, these methods are difficult to scale up to more frames due to their large computation overhead. On the other hand, recent methods take bounding boxes from all frames and points from a small set of context frames (for moving objects) as input, but not explicitly handling the alignment issue [33, 51].

Multi-modality fusion for 3D object detection. A robotics system (such as an autonomous driving car) often has multiple sensors, such as LiDARs, cameras, and radars, which provide complementary information. LiDAR-camera fusion is arguably the most common and well-studied modality fusion configuration [19, 29, 32, 43, 44]. There is also

can be fused with perspective view or camera or radar based detectors as using MoDAR can be considered as a sensor fusion process.

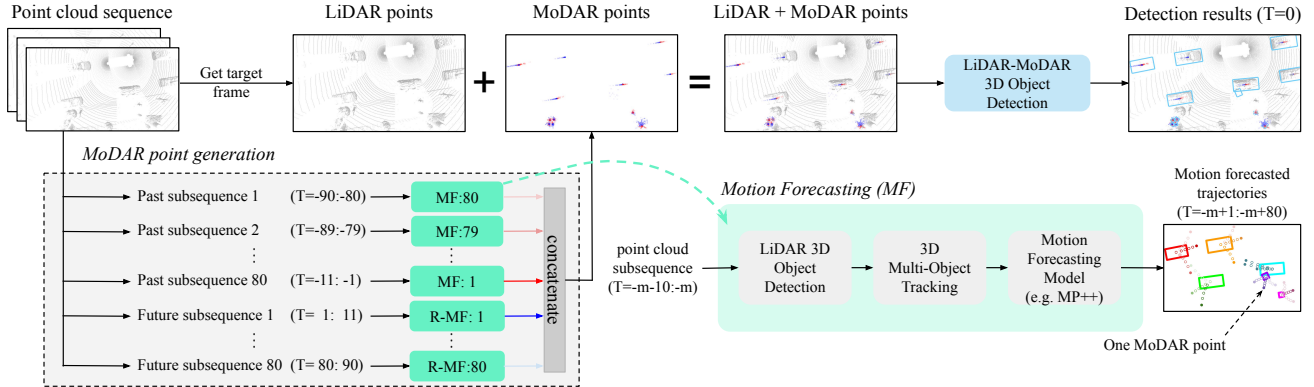


Figure 3. **Using MoDAR for 3D object detection.** Given a point cloud sequence around a target frame, our goal is to estimate 3D object bounding boxes at the target frame $T=0$. To generate MoDAR points, we run motion forecasting (MF) on various subsequences of the input. MF:m means given a subsequence of frames $T=-m-10:-m$ (11 frames), we run motion forecasting to predict object locations m frames ahead of the subsequence, i.e. predict object location and states at frame $T=0$. R-MF means reverse motion forecasting, which is only used in the offline use case. The bottom right figure shows how motion forecasting typically works – it involves running a pre-trained point cloud based detector, a multi-object tracker and a pre-trained motion forecasting model (output colors to indicate different instances). The motion forecasting outputs at the target frame ($T=0$) from various subsequences are concatenated to form the final set of MoDAR points (with the color indicating from which frame they are predicted). We then union the LiDAR and MoDAR points and train/use a LiDAR-MoDAR multi-modal 3D detector (can be any off-the-shelf 3D detector) to get final detection results at the target frame.

work on camera-radar fusion [13]. Fusion methods generally combine information at input level (early fusion), feature level, or decision level (late fusion) [30]. Input level fusion is usually computationally cheap but needs good cross modality alignment [43]. Specifically, people usually convert camera images to virtual points and fuse virtual points and lidar points as the input [49, 58]. On the other extreme, decision level fusion can tolerate modality misalignment, but has to a large compute cost or sub-optimal performance [6]. Our work considers motion forecasting as an additional modality for 3D detection, which provides complementary (temporal) information to LiDAR. By carefully designing the format of MoDAR, it aligns well to LiDAR and can be exploited using the efficient input level fusion strategies. Our proposed MoDAR point can be considered as a variant of virtual point, but different from previous works, our MoDAR points are generated from motion forecasting model, and with rich point features.

Motion forecasting. Given past observations of objects in a dynamic scene, motion forecasting aims to predict future trajectories of the objects. Current state-of-the-art methods [12, 22, 25, 27, 42, 56, 60] learn the complex and nuanced interactions from data through deep neural networks. Some other methods study joint 3D object detection and motion forecasting [1, 4, 5, 20, 23, 28, 48, 61], where detection can be an intermediate task. Among them, Fast and Furious [23] relates to our work as they used motion forecasting to improve detection. In their tracklet decoding module, they aggregate motion forecasting and detector boxes through direct box averaging, which can be considered as a late fusion of motion forecasting and detection results. Compared

to them, this paper proposes an early fusion approach to leverage both motion forecasting (as MoDAR points) and LiDAR data and demonstrate superior performance to the late fusion alternative.

3. Using MoDAR for 3D Object Detection

In this section, we first introduce how to produce the virtual modality MoDAR, and then discuss how to fuse this new modality with a LiDAR point cloud for 3D object detection. Fig. 3 illustrates the entire pipeline.

3.1. MoDAR Point Generation

We propose a virtual sensor modality, MoDAR, which represents object information propagated from past (or/and from future, in an offline setting) to the current frame. As shown in Fig. 3, MoDAR points at $T = 0$ are generated from motion forecasting on a set of history subsequences (in the online setting) or/and a set of future subsequences (in the offline setting). Specifically, given a history subsequence of frames $T = -m - K : -m$ ($K + 1$ context frames), forward motion forecasting predicts future trajectories of all detected objects at frame $T = -m$. We pick predictions that are m frames into the future as the MoDAR points at the current frame. A MoDAR point’s XYZ coordinates are the predicted object center locations while the point features can include object bounding box size, heading, semantic class as well as other metadata from the motion forecasting model (e.g. confidence scores). In an offline setting, we can access future sensor data, which allows us to take a future subsequence and run reverse motion forecasting to predict object locations backwards.

Motion forecasting (MF). To predict object locations into the future (or past), the motion forecasting (MF) involves three typical steps as shown in the bottom right of Fig. 3: detection, tracking, and prediction. We firstly run a pre-trained LiDAR based 3D object detector to localize and classify objects at every frame in the subsequence (results can be cached for overlapping subsequences), then we run multi-object tracking using a Kalman-filter based tracker [47]. Finally, a trajectory prediction model takes the tracked object boxes and predicts future object locations and headings. The trajectory prediction (or motion forecasting model) can be as simple as a constant velocity model [9]. It can also be a pre-trained deep network model such as MultiPath++ [42] (MP++), which is more accurate especially for moving agents in a complex scene. Note that although the motion forecasting output is later used for detection, there is no cyclic dependency of the LiDAR-MoDAR detector and the motion forecasting, as we use a separate pre-trained LiDAR detector to generate the motion forecasting input. It is possible to share the same detector for LiDAR-MoDAR detection and motion forecasting input but requires iterative re-training to converge the detector and motion forecasting model – see supplementary for more discussion and results.

Extensions beyond a single prediction. To make our proposed MoDAR virtual modality more informative, we propose another two extensions. First, to fully leverage the point cloud sequence data, we can combine motion forecasting from separate history (or/and future) subsequences by taking a union of the MoDAR points generated from each subsequence. To distinguish their sources, we add an extra channel of the closet frame timestamp (in the subsequence) to the current frame. Second, given an object track, data-driven motion forecasting models can predict several future trajectories, to handle the uncertainty in object behaviors. For example, MultiPath++ [42] predicts 6 possible trajectories with different confidence scores. MoDAR can include all these predictions. To distinguish them, the trajectory confidence can be added as an addition field of the a MoDAR point.

3.2. LiDAR-MoDAR 3D Object Detection

The generated MoDAR points can be combined with LiDAR points at the current frame (or from a short time window around the current frame) for LiDAR-MoDAR multi-modal 3D object detection. Since MoDAR is based on motion forecasting, it provides less accurate information than LiDARs for areas with good visibility. Therefore a MoDAR-only detection model would have unfavorable detection quality. However, we observe that MoDAR can provide complementary information to the LiDAR sensor especially when LiDAR points are sparse (long-range) or when objects are occluded. For example, when it is hard to estimate an object’s size and heading when there are very few

points, MoDAR points can help provide such information propagated from history (or future frames). When an object becomes occluded, the motion forecasting can still generate a virtual MoDAR point at the occluded region.

To leverage both LiDAR and MoDAR, we use an early fusion at the input level, for two reasons: First, compared to feature level fusion that often requires non-trivial detector architecture update, early fusion is more flexible and can be easily adapted to nearly any off-the-shelf 3D object detectors; Second, compared to late fusion, early fusion is more effective in combining the complementary information from MoDARs and LiDARs. Besides early fusion, we find that adding another late fusion on top of it can further improve pedestrian detection, see Sec. 4.3.5 for more discussion and results.

As MoDAR points are light-weight, we can use many more context frames for our MoDAR-LiDAR detector than alternative methods that rely on point cloud based temporal data fusion. Compared to the number of LiDAR points in a single frame (around 200K in a frame from the Waymo Open Dataset [39]), the number of MoDAR points is marginal. There are $(N \times J)$ MoDAR points from one motion forecasting prediction, where N is the number of objects (usually less than 100), and J is the number of trajectories for each object (e.g. 6). Therefore, to representing information from one frame, MoDAR is around $300 \times$ more efficient than LiDAR. Due to its efficiency, MoDAR helps to include information from more context frames — we use up to 180 frames (18 seconds: 9 seconds in history and 9 seconds in future) in our experiments.

4. Experiments

We evaluate LiDAR and MoDAR fusion detectors on the Waymo Open Dataset (WOD) [39], a large scale autonomous driving dataset with challenging measurements covering different visibility levels. It contains 798 training sequences and 202 validation sequences. Each sequence is around 20 seconds (with around 200 frames at 10Hz).

We evaluate and compare methods with the recommended metrics, Average Precision (AP) and Average Precision weighted by Heading (APH), and report the results on both LEVEL_1 (L1, easy only) and LEVEL_2 (L2, easy and hard) difficulty levels for both vehicles and pedestrians.

4.1. Implementation Details

Generating MoDAR points. To generate the proposed virtual modality MoDAR, we need to prepare (1) a detection and a tracking model to recognize objects from past (and future) point cloud sequences, and (2) a motion forecasting model to predict the future (and past) trajectories.

To prepare the training data for the motion forecasting model, we train LiDAR-only detectors (CenterPoint [59] or SWFormer [40]) on the Waymo Open Dataset train set and

Model	Frame [-p, +f]	Offline Method?	Veh. L1 3D		Veh. L2 3D		Ped. L1 3D		Ped. L2 3D		L2 3D mAPH
			AP	APH	AP	APH	AP	APH	AP	APH	
3D-MAN [55]	[-15, 0]		74.5	74.0	67.6	67.1	71.7	67.7	62.6	59.0	63.1
MPPNet [7]	[-3, 0]		81.5	81.1	74.1	73.6	84.6	81.9	77.2	74.7	74.2
MPPNet [7]	[-15, 0]		82.7	82.3	75.4	75.0	84.7	82.3	77.4	75.1	75.1
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
CenterPoint [59]*	[0, 0]		72.9	72.3	64.7	64.2	71.9	58.3	64.3	51.9	58.1
+MoDAR	[-91, 0]		76.1	75.6	68.9	68.4	73.8	68.7	66.9	62.1	65.3 (+7.2)
+MoDAR	[-91, 91]	✓	80.1	79.5	73.7	73.2	76.4	71.4	69.9	65.0	69.2 (+11.1)
SWFormer [40]*	[0, 0]		77.0	76.5	68.3	67.9	80.9	72.3	72.3	64.4	66.2
+MoDAR	[-91, 0]		80.6	80.1	72.8	72.3	83.5	79.5	75.7	71.8	72.1 (+5.9)
+MoDAR	[-91, 91]	✓	82.9	82.3	75.6	75.1	85.2	81.3	78.0	74.3	74.7 (+8.5)
SWFormer [40]*	[-2, 0]		78.5	78.1	70.1	69.7	82.0	78.1	73.8	70.1	69.9
+MoDAR	[-91, 0]		81.0	80.5	73.4	72.9	83.5	79.4	76.1	72.1	72.5 (+2.6)
+MoDAR	[-91, 91]	✓	84.5	84.0	77.5	77.0	86.3	82.5	79.5	75.8	76.4 (+6.5)

Table 1. **3D object detection results on the WOD val set.** Complementary to LiDAR, our proposed virtual modality MoDAR significantly improves state-of-the-art 3D object detection models, CenterPoint and SWFormer. Our proposed method achieves state-of-the-art compared to previous methods. The Frame column illustrates the indices of the frames that are used for detection. We also mark a method as offline if it uses information from the future. [†]: ensemble with 10 times test-time-augmentation. *: our re-implementation.

then run inference to get detection results on all frames at both train set and validation set frames. Then, to get object tracks, we use a simple Kalman Filter as the multi-object tracker to associate detection results across frames. Finally we apply a data-driven motion forecasting model MultiPath++ [42] on the object tracks to predict their future (and past) trajectories.

The MultiPath++ [42] is trained on the Waymo Open Motion Dataset (WOMD) [9] train set which has more than 70K sequences. Roadgraphs are not used because they are not available in WOD for inference. MultiPath++ takes tracked object boxes from 10 past frames and 1 current frames as input, and predicts object trajectories for the future 80 frames. To run inference on WOD, we pad and resegment each WOD sequence to 91 frames in an overlapping manner. Therefore, given a 200 frame WOD sequence, we take each frame as a current frame to construct a 91-frame segment, obtaining 200 91-frame segments. Instead of using the original track sampling strategy in WOMD, we use all tracks during both training and testing. We train two models: a forward MultiPath++ that takes the past tracks to predict the future, and a backward MultiPath++ that takes the future tracks to predict the past. Note that the backward model is only used for the offline setting, while the forward model is used for both online and offline models. On the WOD val set, the 8 second Average Displacement Error (ADE) [9] are 1.17 and 1.11 for the forward model and the backward model, respectively (see supplementary for more details).

Fusing MoDAR and LiDAR. We will firstly introduce the details of the MoDAR points, and then introduce how we

fuse MoDAR and LiDAR together.

A MoDAR point is structurally similar to a LiDAR point, including a 3D point coordinate and its feature. Specifically, MoDAR points are placed at the center of the predicted object location, and its feature has 13 channels that including object size (normalized by prec-omputed mean and std values), heading (represented by a unit vector), class (one hot encoding with depth 3), object tracking score (*i.e.*, the average of object detection scores over 11 past/current frames), trajectory score, trajectory standard deviation (normalized by its mean and std values), and the timestamp of the closest frame in the input track. In the offline setting, the MoDAR points for a current frame are generated from 160 motion predictions (80 for future prediction, 80 for past prediction) that take different 11-frame input tracks. Therefore, we use the information from 181 frames. In the online setting, the MoDAR points for a current frame are from 80 motion predictions. Besides, the motion forecasting model, MultiPath++ [42], predicts 6 trajectories for each input track.

When fusing MoDAR with LiDAR, we first pad the LiDAR features and MoDAR features to the same length, and then add an additional field (0 for LiDAR and 1 for MoDAR) to indicate the modality of a point.

Detection Models. We re-implemented two popular 3D point cloud detection models, the convolution-based CenterPoint [59] and the transformer-based SWFormer [40]. For CenterPoint, we train 160k steps with a total batch size of 64. For SWFormer, we train 80k steps with a total batch size of 256. The fusion of both LiDAR and MoDAR points are fed into these two models. During training, we apply data augmentations to both LiDAR and MoDAR points.

Predictor	L2 APH		Veh. L2 APH		
	Veh.	Ped.	STN	FST	VFST
-	64.2	51.9	60.5	73.3	79.9
Stationary	73.0	62.1	72.0	73.2	77.6
Constant Velocity	69.2	61.4	67.5	74.6	79.0
1 traj. MP++ [42]	73.5	64.4	71.4	75.5	82.3
6 traj. MP++ [42]	73.2	65.0	70.5	76.2	81.3

Table 2. **Effects of motion forecasting model choices.** The metrics are vehicle L2 3D APH on the WOD val set. The first row is the CenterPoint detector using LiDAR data only. The other four rows are the same detector using LiDAR and MoDAR points (with different trajectory predictors). Results are also broken down by object speed (STN: stationary. FST: fast. VFST: very fast).

4.2. Main Results

Tab. 1 shows adding MoDAR points can improve off-the-shelf 3D object detectors and compares our LiDAR-MoDAR detectors with prior art methods.

From the bottom half of Tab. 1, we see adding MoDAR points to two popular and powerful 3D detectors, CenterPoint [59] and SWFormer [40], leads to significant gains across all metrics for both vehicle and pedestrian detection and in both online and offline settings. For example, adding MoDAR points to the 1-frame CenterPoint base detector, we see 13.1 L2 APH improvement (from 51.9 to 65.0) on pedestrians and 9.0 L2 APH gains (from 64.2 to 73.2) on vehicles. The large gains apply to more powerful base detectors too. For the 3-frame SWFormer detector, adding MoDAR points can still lead to 7.3 L2 APH improvement (from 69.7 to 77.0) for vehicles, and 5.7 L2 APH improvement (from 70.1 to 75.8) for pedestrian.

The improvement on L2 metric is more significant than the L1 metric. For example, MoDAR improves the 3-frame SWFormer by 7.3 L2 APH and 5.9 L1 APH (for vehicle), and by 5.7 L2 APH and 4.4 L1 APH (for pedestrian). Since L1 only considers relatively easy objects (usually more than 5 LiDAR points on them) while L2 considers all objects, this shows that MoDAR helps more in detecting difficult objects with low visibility (more breakdowns in Sec. 4.3.4 and visualizations in Fig. 5).

Tab. 1 also compares our LiDAR-MoDAR detectors with prior art methods that leverage point cloud sequences for online/offline 3D object detection. Our method based on the 3-frame SWFormer gets the best mAPH results among all methods and achieves state-of-the-art numbers on all vehicle and pedestrian metrics. Note that although 3D Auto Labeling (3DAL) from Qi et al. [33] uses a stronger base detector (MVF++ with 5-frame input and test time augmentation) than our 3-frame SWFormer base detector, we can still achieve on par or stronger results than it with the extra input from MoDAR. In appendix, we demonstrate MoDAR can further improve a stronger baseline, LidarAug [16].

4.3. Ablations and Analysis Experiments

This section ablates the MoDAR design and provides more analysis results. Unless otherwise specified, all experiments in this section are based on the 1-frame CenterPoint detector using predictions from past and future 160 frames in total, and using early LiDAR-MoDAR fusion.

4.3.1 Effects of motion forecasting models

In Tab. 2, we compare detection results using MoDAR points generated from different motion forecasting models. We ablate 4 different motion forecasting models. (1) Stationary predictor: it aggressively assumes all objects are stationary, predicting objects’ future positions as their most recent positions. (2) Constant velocity predictor: it assumes all objects are moving at the constant velocity estimated from the observed frames. (3) MultiPath++ [42] (MP++) predicting the most confident trajectory (1 traj.); (4) MultiPath++ predicting the top 6 confident trajectories (6 traj.).

We see that MoDAR improves the CenterPoint baseline with all four predictors. The data-driven MultiPath++ model shows the best overall performance compared to other predictors. When looking into the velocity breakdown metrics (provided by WOD), we observe that stationary predictor achieves the best performance (72.0 L2 APH) for the stationary (STN) vehicles, regresses on very fast (VFST) vehicles (from 79.9 to 77.6). The constant velocity model is better than stationary predictor for fast (FST) and very fast (VFST) objects. Note that the constant velocity model does not perform as well as the stationary predictor because the input track is noisy: even though the objects are not moving, detection noises can lead to wrong velocity estimation. Finally, the MP++ predictors perform the best for moving (*i.e.*, fast and very fast) vehicles. 1-trajectory and 6-trajectory MP++ models lead to similar detection results. Note that we use 6-trajectory MP++ model as our final version. To seek higher model efficiency, 1-trajectory MP++ predictor can be selected which only includes 1/6 MoDAR points compared to 6-trajectory MP++ model.

4.3.2 Effects of different MoDAR point features

Tab. 3 ablates the importance of different object states in MoDAR points. Most information, includes object location, size, heading, type, confidence scores (from both tracking and motion forecasting), help improve the detection quality. By taking a closer look, we observe that the most important information is the object location, which improves 5.5 L2 APH (from 64.2 to 69.7) for vehicles and 10.7 L2 APH (from 51.9 to 61.2) for pedestrians. Since vehicle heading is relatively easy to estimate, adding headings to MoDAR features mainly helps pedestrian detection. For our main results, we used all features to get the best performance.

Location	Size	Heading	Class	Scores	Veh. L2 APH	Ped. L2 APH
✗	✗	✗	✗	✗	64.2	51.9
✓	✗	✗	✗	✗	69.7	61.2
✓	✓	✗	✗	✗	71.2	62.7
✓	✓	✓	✗	✗	70.9	64.0
✓	✓	✓	✓	✗	71.3	64.1
✓	✓	✓	✓	✓	73.2	65.0

Table 3. **Effects of different object state features in MoDAR points.** The metric is L2 3D APH on the WOD val set.

4.3.3 Are long-term point cloud sequences helpful?

In Fig. 4, we show the impact of the temporal ranges (what frames are used for motion forecasting) of MoDAR points on detection. We split the study to two settings: online and offline. In the online setting, only past frames are used to generate MoDAR points. For the offline setting, both past and future frames are used. We include the same number of future frames as past frames for the offline setting. For the cases of using K past predictions, we select past subsequences from T=-11:-1 to T=-K=10:-K (K subsequences) and use the forward motion forecasting from them to generate MoDAR points. Similarly for the reverse prediction from future subsequences.

The results are shown in Fig. 4. The red bars are the performance of the online setting, while the green bars are for the offline setting. We observe that for both setting, adding MoDARs from more past or future predictions generally lead to better detection and this improvement does not saturate until using MoDARs from 80 past and 80 future predictions. It is also noteworthy that the future frames provide unique information that significantly improves the results compared to only using past frames.

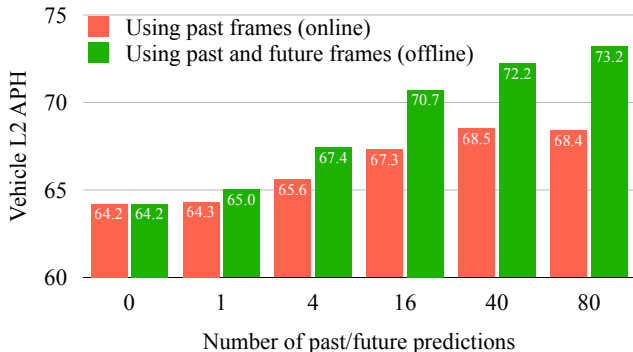


Figure 4. **Effects of MoDAR temporal context sizes on 3D object detection.** The metrics are vehicle L2 3D APH on the WOD val set. The (left) red bars are the performance of models using only MoDAR points generated from the past frames, while the (right) green bars are the performance of models that use the same number of past and future frames for MoDAR point generation.

Model	0-30m	30-50m	50m+	All
LiDAR only	90.4	69.7	45.6	69.7
LiDAR + MoDAR	92.2	76.9	58.5	77.0
	(+1.2)	(+7.2)	(+12.9)	(+7.3)

Table 4. **Distance breakdown for LiDAR-based and LiDAR-MoDAR based detection.** The metrics are vehicle L2 3D APH on the WOD val set across different ground-truth depth ranges. The base detector is a 3-frame SWFormer.

4.3.4 Performance breakdown by object distances

To better understand how MoDAR improves the LiDAR-based 3D object detection models, we provide both qualitative and quantitative analysis based on the 3-frame SWFormer model and our MoDAR variant.

Following previous works [19, 33], we divide the vehicles into three groups based on their distance to the ego-car: within 30 meters (short-range), from 30 to 50 meters (mid-range), and beyond 50 meters (long-range). Tab. 4 shows the *relative* gains by using MoDAR. MoDAR improves the results in all distance ranges. In particular, it achieves a much more significant gains for long-range vehicles (by 12.9 APH, 28.3% relatively) than short-range vehicles (by 1.8 APH, 2.0% relatively). This is likely because long-range objects have very sparse points in their observations, making it difficult to estimate their locations, headings and sizes. MoDAR fills this gap to a large extent.

4.3.5 Comparing LiDAR-MoDAR fusion methods

In Tab. 5, we compare detection results using a single modality and results using both LiDAR and MoDAR modalities using different fusion strategies.

For LiDAR-only detection, we train a 3-frame SWFormer that only takes LiDAR points as input. For MoDAR-only, we directly use the motion forecasted boxes as the detection output (assuming constant box sizes and box elevation). Note that motion predictions from nearby frames (e.g. 1 or 2 frames away) can give very similar results as to detection from the current frame, as the scene does not change dramatically between nearby frames. With some hyper parameter tuning, we select MoDAR points from the closest 10 predictions (5 past and 5 future) for the MoDAR-only detection and then apply a weighted 3D box fusion [37] to aggregate overlapping boxes (see supplementary for more details and ablations). From Tab. 5 first two rows, we can see that LiDAR-based detection gets more accurate results than MoDAR-based ones especially for pedestrians, for which motion forecasting can be noisy.

Fast and Furious [23] used a late fusion approach to combine detector and motion forecasting results through box averaging. To implement a late fusion method, we compute weighted box averaging [37] of boxes from current frame LiDAR detection and motion predictions from nearby 10

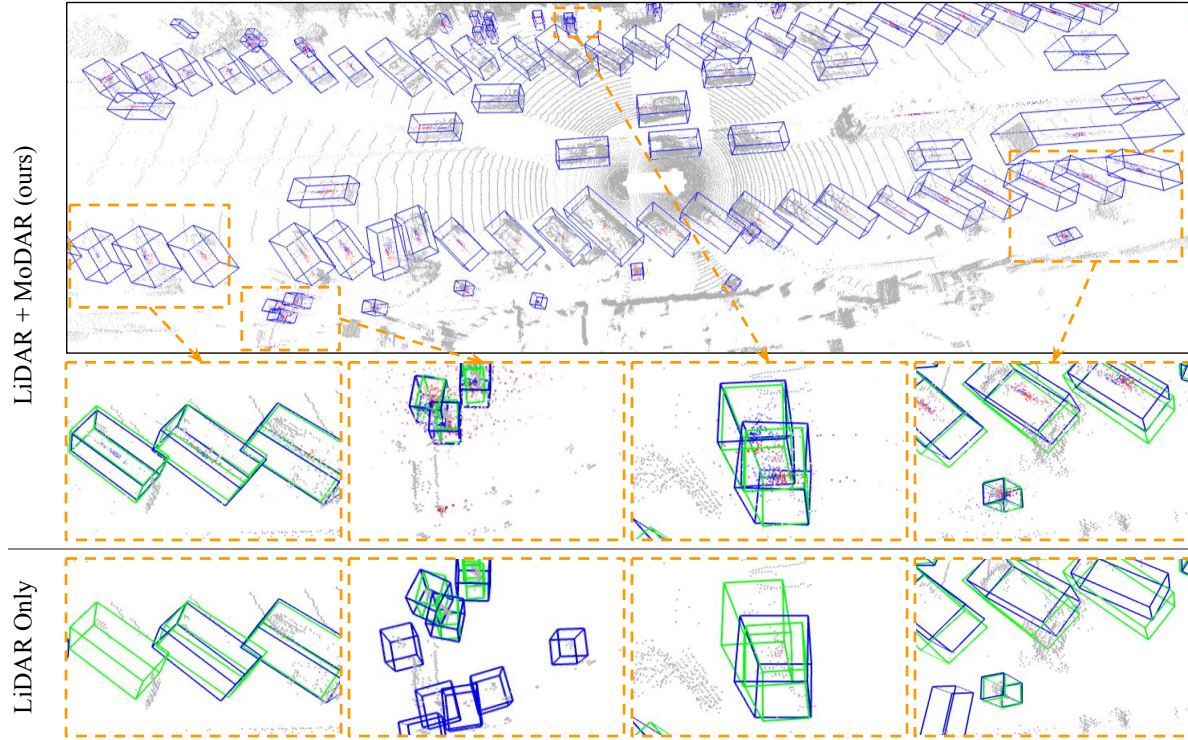


Figure 5. **Qualitative results of 3D object detection on the WOD val set.** Blue boxes: model predictions; Green boxes: ground truth boxes. We used a 3-frame SWFormer as the base detector architecture and used MoDAR points generated from 180 context frames (offline setting). Comparing the results from LiDAR-MoDAR multi-modal detector versus the LiDAR only detector, we can see that the LiDAR-MoDAR detector can recognize more heavily occluded objects or estimate their shapes and headings more accurately.

LiDAR	MoDAR	Fusion Method	Veh. L2		Ped. L2	
			AP	APH	AP	APH
✓	✗	-	70.1	69.7	73.8	70.1
✗	✓	-	67.4	66.8	69.6	63.8
✓	✓	Early	77.5	77.0	77.8	74.4
✓	✓	Late	70.9	70.4	76.4	72.3
✓	✓	Early+Late	77.6	77.1	79.5	75.8

Table 5. **Compare detection results with different modalities and different fusion methods.** The metrics are 3D and BEV L2 APH on the WOD val set. We use a 3-frame SWFormer.

predictions (past 5 and future 5). For the early fusion, we use motion forecasting from 160 predictions (past 80 and future 80) to generate MoDAR points. In Tab. 5 third and fourth rows, we can see that early fusion achieves significantly better results than the late fusion. In the last row, we show that if we combine the early and late fusion by fusing forecasted boxes from nearby 10 frames with LiDAR-MoDAR detection, we can further improve detection quality. For our main results in Tab. 1, we take advantages of late fusion for pedestrian detection (early+late fusion).

5. Conclusions

In this paper, we proposed MoDAR, a virtual sensor modality that uses motion forecasting to propagate object

states from past and future frames to a target frame. Each MoDAR point represents a prediction of an object’s location and states on a forecasted trajectory. The MoDAR points generated from a point cloud sequence can be fused with other sensor modalities such as LiDAR to achieve more robust 3D object detection especially for cases with low visibility (occluded) or in long range. Due to its simplicity, the MoDAR idea can be applied to a wide range of existing detectors not even restricted to point-cloud-based ones. Evaluated on the Waymo Open Dataset, we have demonstrated the effectiveness of the MoDAR points for two popular 3D object detectors, achieving state-of-the-art results. We have also provided extensive analysis to understand different components of the MoDAR modality.

We believe this work provides another perspective of the relationship between detection and motion forecasting. In the future, it would be appealing to study how to jointly optimize motion prediction and detection, as well as revisiting the interface design between them.

Acknowledgement. We specially thank Yurong You for idea discussion and preliminary exploration, and thank Scott Ettinger, and Chiyu “Max” Jiang for insightful discussion and technical support. Yingwei Li thank Longlong Jing, Zhaoqi Leng, Pei Sun, Tong He, Mingxing Tan, Hubert Lin, Xuanyu Zhou, Mahyar Najibi, and Kan Chen for tutorial, discussion and technical support.

References

- [1] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018. [3](#)
- [2] Alex Bewley, Pei Sun, Thomas Mensink, Dragomir Anguelov, and Cristian Sminchisescu. Range conditioned dilated convolutions for scale invariant 3d object detection. *arXiv preprint arXiv:2005.09927*, 2020. [2](#)
- [3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020. [1](#)
- [4] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Conference on Robot Learning*, pages 947–956. PMLR, 2018. [3](#)
- [5] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14403–14412, 2021. [3](#)
- [6] Vassilios Chatzis, Adrian G Bors, and Ioannis Pitas. Multi-modal decision-level fusion for person authentication. *IEEE transactions on systems, man, and cybernetics-part a: systems and humans*, 29(6):674–680, 1999. [3](#)
- [7] Xuesong Chen, Shaoshuai Shi, Benjin Zhu, Ka Chun Chung, Hang Xu, and Hongsheng Li. Mppnet: Multi-frame feature intertwining with proxy points for 3d temporal object detection. In *European Conference on Computer Vision*. Springer, 2022. [1](#), [2](#), [5](#)
- [8] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1355–1361, May 2017. [2](#)
- [9] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021. [2](#), [4](#), [5](#), [12](#)
- [10] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8458–8468, 2022. [2](#)
- [11] Runzhou Ge, Zhuangzhuang Ding, Yihan Hu, Yu Wang, Sijia Chen, Li Huang, and Yuan Li. Afdet: Anchor free one stage 3d object detection. *arXiv preprint arXiv:2006.12671*, 2020. [2](#)
- [12] Junru Gu, Chen Sun, and Hang Zhao. Densettnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15303–15312, 2021. [3](#)
- [13] Jyh-Jing Hwang, Henrik Kretzschmar, Joshua Manela, Sean Rafferty, Nicholas Armstrong-Crews, Tiffany Chen, and Dragomir Anguelov. Cramnet: Camera-radar fusion with ray-constrained cross-attention for robust 3d object detection. In *European Conference on Computer Vision (ECCV)*, pages 388–405. Springer, 2022. [3](#)
- [14] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018. [2](#)
- [15] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. [2](#)
- [16] Zhaoqi Leng, Guowang Li, Chenxi Liu, Ekin Dogus Cubuk, Pei Sun, Tong He, Dragomir Anguelov, and Mingxing Tan. Lidaraugment: Searching for scalable 3d lidar data augmentations. *arXiv preprint arXiv:2210.13488*, 2022. [6](#), [14](#), [15](#)
- [17] B. Li. 3d fully convolutional network for vehicle detection in point cloud. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1513–1518, Sep. 2017. [2](#)
- [18] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. In *RSS 2016*, 2016. [2](#)
- [19] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V Le, et al. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17182–17191, 2022. [2](#), [7](#)
- [20] Ming Liang, Bin Yang, Wenyuan Zeng, Yun Chen, Rui Hu, Sergio Casas, and Raquel Urtasun. Pnpnet: End-to-end perception and prediction with tracking in the loop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11553–11562, 2020. [3](#)
- [21] Chenxi Liu, Zhaoqi Leng, Pei Sun, Shuyang Cheng, Charles R Qi, Yin Zhou, Mingxing Tan, and Dragomir Anguelov. Lidarnas: Unifying and searching neural architectures for 3d point clouds. In *European Conference on Computer Vision (ECCV)*, pages 158–175. Springer, 2022. [2](#)
- [22] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7577–7586, 2021. [3](#)
- [23] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3569–3577, 2018. [2](#), [3](#), [7](#)
- [24] Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K. Wellington. LaserNet: An efficient probabilistic 3D object detector for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [2](#)

- [25] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. *arXiv preprint arXiv:2207.05844*, 2022. 3
- [26] Jiquan Ngiam, Benjamin Caine, Wei Han, Brandon Yang, Yuning Chai, Pei Sun, Yin Zhou, Xi Yi, Ouais Alsharif, Patrick Nguyen, Zhifeng Chen, Jonathon Shlens, and Vijay Vasudevan. Starnet: Targeted computation for object detection in point clouds. *CoRR*, 2019. 2
- [27] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified architecture for predicting multiple agent trajectories. *arXiv preprint arXiv:2106.08417*, 2021. 3
- [28] Neehar Peri, Jonathon Luiten, Mengtian Li, Aljoša Ošep, Laura Leal-Taixé, and Deva Ramanan. Forecasting from lidar via future object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17202–17211, 2022. 3
- [29] AJ Piergiovanni, Vincent Casser, Michael S Ryoo, and Anelia Angelova. 4d-net for learned multi-modal alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15435–15445, 2021. 2
- [30] Cle Pohl and John L Van Genderen. Review article multi-sensor image fusion in remote sensing: concepts, methods and applications. *International journal of remote sensing*, 19(5):823–854, 1998. 3
- [31] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9277–9286, 2019. 2
- [32] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 918–927, 2018. 2
- [33] Charles R Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3d object detection from point cloud sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6134–6144, 2021. 1, 2, 5, 6, 7, 14
- [34] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointcnn: 3d object proposal generation and detection from point cloud. *arXiv preprint arXiv:1812.04244*, 2018. 2
- [35] Weijing Shi and Ragunathan (Raj) Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2
- [36] Martin Simony, Stefan Milzy, Karl Amendey, and Horst-Michael Gross. Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2
- [37] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107:104117, 2021. 7, 13
- [38] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *CVPR*, 2016. 2
- [39] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020. 2, 4, 12, 14
- [40] Pei Sun, Mingxing Tan, Weiyue Wang, Chenxi Liu, Fei Xia, Zhaoqi Leng, and Dragomir Anguelov. Swformer: Sparse window transformer for 3d object detection in point clouds. In *European Conference on Computer Vision*. Springer, 2022. 1, 2, 4, 5, 6, 13, 15
- [41] Pei Sun, Weiyue Wang, Yuning Chai, Gamaleldin Elsayed, Alex Bewley, Xiao Zhang, Cristian Sminchisescu, and Dragomir Anguelov. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5725–5734, 2021. 2
- [42] Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, et al. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7814–7821. IEEE, 2022. 2, 3, 4, 5, 6, 12
- [43] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4604–4612, 2020. 2, 3
- [44] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11794–11803, 2021. 2
- [45] Dominic Zeng Wang and Ingmar Posner. Voting for voting in online point cloud object detection. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015. 2
- [46] Yue Wang, Alireza Fathi, Abhijit Kundu, David Ross, Caroline Pantofaru, Thomas Funkhouser, and Justin Solomon. Pillar-based object detection for autonomous driving. In *ECCV*, 2020. 2
- [47] Xinshuo Weng and Kris Kitani. A baseline for 3d multi-object tracking. *arXiv preprint arXiv:1907.03961*, 1(2):6, 2019. 4
- [48] Pengxiang Wu, Siheng Chen, and Dimitris N Metaxas. Motionnet: Joint perception and motion prediction for autonomous driving based on bird’s eye view maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11385–11395, 2020. 3
- [49] Xiaopei Wu, Liang Peng, Honghui Yang, Liang Xie, Chenxi Huang, Chengqi Deng, Haifeng Liu, and Deng Cai. Sparse

- fuse dense: Towards high quality 3d detection with depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5418–5427, 2022. 3
- [50] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2
- [51] Bin Yang, Min Bai, Ming Liang, Wenyuan Zeng, and Raquel Urtasun. Auto4d: Learning to label 4d objects from sequential point clouds. *arXiv preprint arXiv:2101.06586*, 2021. 2
- [52] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 2
- [53] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11040–11048, 2020. 2
- [54] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Ipod: Intensive point-based object detector for point cloud, 2018. 2
- [55] Zetong Yang, Yin Zhou, Zhifeng Chen, and Jiquan Ngiam. 3d-man: 3d multi-frame attention network for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1863–1872, 2021. 1, 2, 5
- [56] Maosheng Ye, Jiamiao Xu, Xunnong Xu, Tongyi Cao, and Qifeng Chen. Dcms: Motion forecasting with dual consistency and multi-pseudo-target supervision. *arXiv preprint arXiv:2204.05859*, 2022. 3
- [57] M. Ye, S. Xu, and T. Cao. Hvnnet: Hybrid voxel network for lidar based 3d object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1628–1637, 2020. 2
- [58] Tianwei Yin, Xingyi Zhou, and Philipp Kraehenbuehl. Multimodal virtual point 3d detection. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 3
- [59] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuehl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11784–11793, 2021. 1, 2, 4, 5, 6
- [60] Wenyuan Zeng, Ming Liang, Renjie Liao, and Raquel Urtasun. Lanercnn: Distributed representations for graph-centric motion forecasting. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 532–539. IEEE, 2021. 3
- [61] Zhishuai Zhang, Jiyang Gao, Junhua Mao, Yukai Liu, Dragomir Anguelov, and Congcong Li. Stinet: Spatio-temporal-interactive network for pedestrian detection and trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11346–11355, 2020. 3
- [62] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2
- [63] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings*

of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4490–4499, 2018. 2

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 re-assign 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.	3.34	10.2	1.40	4.01
	WOD Val.	1.83	9.22	0.82	3.67
WOMD (Dense)	WOMD Val.	3.61	11.1	1.42	3.74
	WOD Val.	1.17	5.45	0.55	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	#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	#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 sec-

tion, 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 [-p, +f]	Offline Method?	Veh. L1 3D		Veh. L2 3D		Ped. L1 3D		Ped. L2 3D		L2 3D mAPH
			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 Predictions	Fusion Method	Veh. L2		Ped. L2	
		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).

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

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.

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 $5 \times$ 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
3DAL	-	85.8/85.5	77.2/76.9	-	-
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 official ranking metric on the WOD leaderboard.

F. Results on the WOD Test Set

Tab. 12 shows vehicle and pedestrian 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.