Once Detected, Never Lost:
Surpassing Human Performance in Offline LiDAR based 3D Object Detection

Lue Fan[1,2,3] Yuxue Yang[1,2] Yiming Mao[4] Feng Wang[6] Yuntao Chen[5]
Naiyan Wang[6] Zhaoxiang Zhang[1,2,5,7]
[1]Institute of Automation, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences (UCAS) [3]School of Future Technology, UCAS
[4]Hunan University [5]Centre for Artificial Intelligence and Robotics (HKISI_CAS)
[6]TuSimple [7]State Key Laboratory of Multimodal Artificial Intelligence Systems
{fanlue2019, yangyuxue2023, zhaoxiang.zhang}@ia.ac.cn mym0729@hnu.edu.cn
{feng.wff, chenyuntao08, winsty}@gmail.com

Abstract

*This paper aims for high-performance offline LiDAR-based 3D object detection. We first observe that experienced human annotators annotate objects from a track-centric perspective. They first label objects in a track with clear shapes, and then leverage the temporal coherence to infer the annotations of obscure objects. Drawing inspiration from this, we propose a high-performance offline detector in a track-centric perspective instead of the conventional object-centric perspective. Our method features a bidirectional tracking module and a track-centric learning module. Such design allows our detector to infer and refine a complete track once the object is detected at a certain moment. We refer this characteristic to "on**C**e dete**T**ed, neve**R** **L**ost" and name the proposed system **CTRL**. Extensive experiments demonstrate the remarkable performance of our method, surpassing the human-level annotating accuracy and previous state-of-the-art methods in the highly competitive Waymo Open Dataset leaderboard without model ensemble. The code is available at https://github.com/tusen-ai/SST.*

1. Introduction

As a fundamental task in autonomous driving, 3D object detection has made great progress in recent years, in both LiDAR-based detectors [44, 17, 49, 31] and image-based detectors [40, 15, 20, 21]. Such success is primarily attributed to the data-driven paradigm, which requires a massive amount of labeled data. As a result, there is growing interest in developing a high-performance offline detector for auto-labeling.

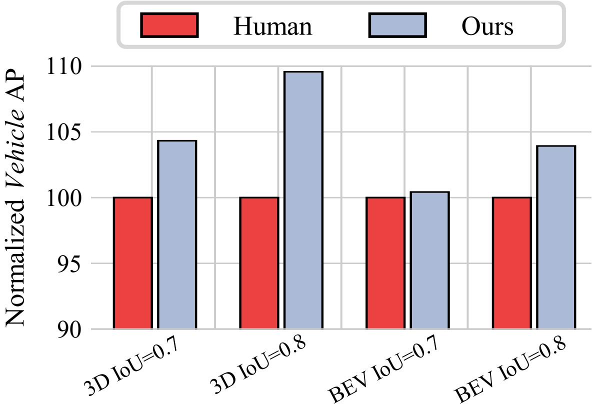

Figure 1: Our method consistently outperforms human performance in Waymo Open Dataset. Human performance is reported by 3DAL [30]. We normalize the AP of human performance to 100 for better visualization.

To achieve a high-performance offline detector, we first analyze the behavior of human annotators in a standard sequence labeling process. We find that humans annotate objects from a track-centric perspective, meaning that they utilize the temporal motion cues of objects to achieve precise labeling. In particular, annotators first label easy samples in a sequence, and then use temporal cues to propagate these high-confidence labels to other time steps, when the objects may contain very few points and are hard to be accurately labeled individually. In this process, once an object is detected in a certain moment, human annotators continuously track that object and predict its movements over time, even if there are some periods when the object is temporarily invisible. An important fact is exploited here: *a detected object will not disappear unless it moves outside the perception range.* Inspired by human labeling behavior, we develop a track-centric offline detector that aims to achieve two objectives: 1) Accurate labeling of well-observed tracks. 2) For tracks containing only a few high-quality frames, our detector propagates the predictions in high-quality frames to low-quality frames. We refer to this key property of the detector as "*onCe detecTed, neveR Lost*" and accordingly name our method **CTRL**.

In order to implement our vision, we first utilize a base detector to obtain detection results. Then we propose a bidirectional tracking module. During the tracking process, a simple motion model is adopted to fill in missing frames and bidirectionally extend the track, greatly extending the life cycle of the track. The bidirectional tracking offers the detector an opportunity to catch obscure objects which might be totally ignored by the base detector. However, such a heuristic frame filling or extension cannot obtain precise poses of objects. Therefore, we develop a track-centric learning module for refinement. In the design of this module, we follow a track-centric principle: tracks, rather than objects, should be regarded as first-class citizens in the workflow. This module takes the all points and all proposals in the entire track as input, and refines all their poses simultaneously. In addition, after the refinement, we could optionally optimize the track coherence via a temporal coherence optimization module if needed. Particularly, we manually specify a bounding box containing a clear object shape in a track, and then align other object shapes with the specified one by point cloud registration[1, 7]. We summarize our contributions as follows.

- Based on the behavior of human annotators, we propose an offline detection system CTRL, following the philosophy of "track-centric" and "once detected, never lost". CTRL boosts the performance of auto-labeling.
- Single-model CTRL outperforms the previous state-of-the-art offline detector and all the online detectors. It is worth emphasizing that among millions of vehicles, only 0.48% of them would be completely missed by CTRL.
- We carefully relabel some diverged cases between our predictions and official ground truths. Our results demonstrate that our method even surpasses the ground-truth accuracy provided by Waymo human annotators in those cases.
- We keep our methodology simple and clean, greatly simplifying the workflow and reducing the resource requirements of existing offline frameworks.

2. Related Work

LiDAR-based 3D Object Detection LiDAR-based 3D object detectors usually adopt three representation modes, namely point-based [28, 29, 27, 33, 47], voxel-based [44, 17, 31, 8, 9, 10, 5], and range-image-based [25, 11, 2, 37]. Recently, combining multi-frame point clouds becomes a prevalent approach since it can provide a more comprehensive representation than a single-frame point cloud. Multi-frame detectors [51, 49, 14, 32, 48] have demonstrated that the concatenation of multi-frame point clouds can significantly outperform the single-frame setting. MPPNet [4] incorporates multi-frame feature encoding and interaction modules, which results in higher performance. INT [43] builds a temporal feature bank for temporal information aggregation. FSD++ [12] leverages temporal information to generate residual points, reducing the temporal redundancy of multi-frame detection.

LiDAR-based 3D Multi-object Tracking Due to the accurate distance measurements that LiDAR provides, most state-of-the-art 3D Multiple Object Tracking (MOT) algorithms adopt a "tracking-by-detection" paradigm [41]. The MOT in 3D space is usually easier than that in the 2D image space, since the motion of objects in 3D is much easier to predict which is beneficial for data association. These algorithms proposed various methods to enhance data association [26, 42, 49], motion propagation [6], and object life cycle management [26, 39]. Specifically, Immortal Tracker [39] demonstrates that even a simple motion model with (almost) forever track preservation could significantly reduce premature track termination. Their findings coincide with our observations of human annotators. Recently, some of the latest methods incorporate learning-based algorithms to improve association with features from point clouds [35, 23]. For example, SimTrack [23] demonstrates an end-to-end trainable model for joint detection and tracking from raw point clouds by feature alignment.

3D Object Automated Labeling In recent years, the annotation costs for training data-hungry models have increased significantly. Accurate auto-labeling can significantly reduce annotation time and cost [3]. For 3D object detection, 3DAL [30] used point cloud sequence data, which achieves significant gains compared to state-of-the-art onboard detectors and offboard baselines, and even matches human performance. Auto4D [45] proposed an automatic annotation pipeline for generating accurate object trajectories in 3D space from LiDAR point clouds, achieving a 25% reduction in human annotation workload. Numerous studies have proposed different settings to assist human annotators and mitigate the expenses associated with annotation [50, 24, 13, 18].

3. Once Detected, Never Lost

3.1. Base Detector

We opt for the emerging FSD [9] as the base detector because of its proven effectiveness and ease of use. Some

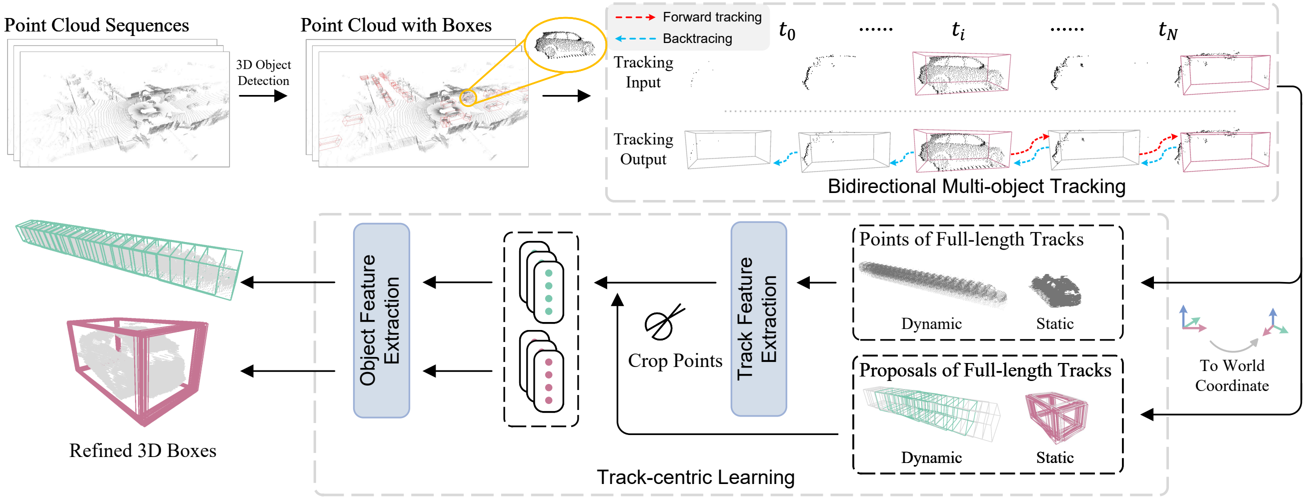

Figure 2: **Overall architecture of CTRL.** (1) A base detector is utilized to generate basic detection results. (2) In the bidirectional tracking module, a forward tracking process is applied first to fill missing boxes and extend the track to the future, indicated by the red arrow. Then we backtrace to the start frame and extend the track to the past, indicated by the blue arrow. (3) The bidirectional extended tracks are sent into the track-centric learning module for refinement.

modifications are made to make FSD better suit our needs. (1) As our system is designed for offline use, in addition to the traditional multi-frame point cloud concatenation, we also incorporate future frames, which leads to a considerable performance improvement. (2) In order to utilize longer temporal information without too much computational overhead, we employed a frame-skipping strategy. In practice, we sample 9 frames of data in total, i.e. $[t-8, t-6, \cdots, t, \cdots, t+6, t+8]$. (3) To prevent overfitting, we employ the frame dropout strategy, in which half of the frames in the selected frames are randomly dropped with a probability of 20%.

3.2. Bidirectional Multi-object Tracking

Since the motion of objects in 3D space is much easier to predict than that in the 2D images, one simple idea to achieve "never lost" is to extend the track based on the motion model. Recently, Immortal Tracker [39] is designed to fit this principle, which predicts a pseudo-box by the motion model when a tracklet matches no observation in a time step. Thus, the tracklets will never die unless the objects are out-of-range or the sequence ends. With its "immortal" feature, it is able to associate broken tracks due to occlusion or missing detection in long term.

The Immortal Tracker operates in a forward-only manner, which enables it to fill in missing boxes once an object has been initially detected. However, the objects cannot appear from nothing. They should also exist before being detected. Thus, we **backtrace** the tracks to the beginning of the tracks, and use the backward motion model to extend tracks into the past. We illustrate this process in Figure 2.

Here, it is necessary to point out that the positions and confidences of these added boxes are not accurate, leading to many false positives. To address this issue, we propose a solution in the following §4.

4. Track-centric Learning

In this section, we propose a track-centric learning module to improve prediction quality. This module takes full-length tracks as input which will be elaborated in §4.1. Its network structure consists of two major parts. The first part utilizes the whole track for track-level feature extraction. The second part extracts box-level features for proposal refinement. We present these two parts in §4.2. In §4.3, we propose a novel track-centric label assignment. We also propose a couple of post-processing techniques to further improve the prediction quality in §4.5.

4.1. Organizing Track Input

Multiple-In-Multiple-Out. In the previous object-centric designs [30, 4], the model takes *multiple* point clouds from nearby time steps to refine a *single* proposal in every forward pass. For brevity, we name this as *Multiple-In-Single-Out* (MISO). Instead, in our track-centric design, the network takes a full-length track as input and simultaneously refines all proposals of the track in a single forward pass. Correspondingly, we name this way as *Multiple-In-Multiple-Out* (MIMO). Specifically, in MIMO, we first expand proposals in a track by 2 meters in all three dimensions. Then we crop the point clouds in every time step by the expanded proposals. The cropped points are transformed into the pose of the first frame of the track and concatenated together as network input. MIMO enables two unique features of our methods.

- **Effective Training.** With MIMO, our approach simultaneously applies supervision on every object in different time step in a single forward pass. Furthermore, the track features can be shared by all the objects in the track. Such dense supervision and feature sharing make the training very effective.
- **Computational Efficiency.** In MISO, point clouds from multiple time steps are paired with a single proposal for input. Thus, the computational overhead grows along with the length of input point clouds. However, in MIMO design, all the point clouds in a track are concatenated and shared by all proposals. This characteristic allows for high efficiency and almost unlimited input length.

Motion-state agnostic. The previous work [30] partitions all tracks by their motion states, and designs two different pipelines for the dynamic and static tracks, respectively. However, we find it unnecessary and even harmful. On one hand, such a partition reduces the diversity of training data, thus hindering generalization. On the other hand, some categories may have unstable motion states, such as pedestrians. Therefore, we don't distinguish the dynamic and static objects and treat them in a unified way throughout the whole pipeline, greatly simplifying the workflow.

Timestamp encoding. To distinguish the points from different time steps, we attach timestamp encoding to the input points. Although there are many options, we opt for the simplest one that uses their absolute frame ID in sequence as the time encoding. Specifically, points from i-th frame will be attached with $0.01 \times i$ as its timestamp encoding.

4.2. Sparse Feature Extraction

Here we introduce the network structure for feature extraction and prediction. The network consists of a Sparse UNet[34] backbone to extract features of the track points and a PointNet-like head to predict object parameters.

Track features. Since we use full-length tracks as input, the spatial span of these tracks might be considerably large. Thus, in practice, we empirically adopt an input space in the size of $[-256m, 256m] \times [-256m, 256m]$, which could cover the spatial spans of most tracks in WOD [36]. To handle such a large space, we turn to the emerging fully sparse architecture. In particular, we first voxelize the point clouds of tracks. Then a Sparse UNet with aggressive downsampling is leveraged to extract the voxel features, which has sufficient receptive fields. Eventually, the voxel features are mapped back to their containing points by interpolation. The obtained point features are utilized for object feature extraction in the following.

Object features. After the SparseUNet, we use proposals to crop the point features, where the proposals are expanded by a certain margin (e.g., 0.5 meters) to ensure the integrity of the object. Then an efficient PointNet is utilized to extract the proposal features, which contains several MLPs and max-pooling layers. Note that a proposal in the time step t will also crop the points from other time steps to get more additional information. We append a special flag on the points in the current frame, making the model easier to recognize the current object shape, avoiding being confused by points from other time steps.

4.3. Label Assignment

Previous object-centric detectors [30, 31, 34, 9] match objects and ground-truth boxes with a certain metric (e.g., IoU) to assign labels. Due to a lack of temporal constraints, proposals might be mistakenly assigned as negative. In our method, global track information could be leveraged as strong priors for more robust label assignment. Therefore, we develop a track-centric label assignment.

Track IoU. We first define Track IoU (TIoU) to measure distances between tracks, formulated in Equation 1.

$$\mathrm{TIoU}(T_a^{S_a}, T_b^{S_b}) = \frac{\sum_{i=1}^{|S_a \cap S_b|} \mathrm{IoU}(B_a^i, B_b^i)}{|S_a \cup S_b|}, \qquad (1)$$

where S_a and S_b are the indices of time step of track T_a and T_b, respectively, and B^i is the i-th box in the overlapped part of the two tracks. As can be seen from Equation 1, TIoU is actually the average IoU of each box pair divided by the length of the union of two tracks. Using the TIoU, we assign the tracks in a two-round manner as follows.

Assignment. In the first round, we match the predicted tracks with ground-truth tracks. A ground-truth track is assigned to a predicted track only if their TIoU is larger than a predefined threshold. Afterward, a predicted track might have multiple ground-truth track candidates. Predicted tracks with no candidates are regarded as negative. In this round, the track-level matching offers better robustness and accuracy than the object-level matching. In the second round, for a matched track pair, we assign the GT boxes to the proposals frame by frame. Figure 3 illustrates the process. At a certain time step, if there are multiple GT boxes from different candidate GT tracks, we simply choose one GT box from the GT track with a higher TIoU. This second round helps alleviate the potential assignment ambiguity of the wrong associated predicted track.

4.4. Detection Head and Losses

Classification. In the classification, we use a quality-aware loss as in [34, 31, 9, 19]. We first calculate the IoU between each proposal and its corresponding GT. Eventually, using the IoU, we obtain a soft classification target

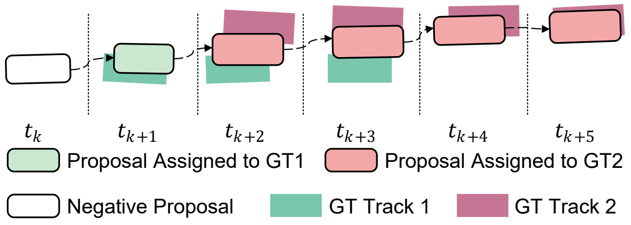

Figure 3: **The second-round assignment**. At t_{k+2}, the proposal has higher IoU with GT1, but it is assigned to GT2 due to the predicted track having higher TIoU with GT track 2.

q, formulated as $q = \min(1, \max(0, 2\text{IoU} - 0.5))$. For each proposal, we aggregate its contained point features extracted by PointNet into a proposal embedding by max pooling. Then an MLP is adopted to predict the soft label q from the embedding. The final classification loss $\mathcal{L}_{cls}$ is the cross entropy between predicted logits and the soft labels.

Regression. The regression target for a positive sample is the box residual to its corresponding ground-truth box, and is the same as [34, 9]. Then the regression loss is denoted as $\mathcal{L}_{reg} = L1(\Delta_{res}, \widehat{\Delta_{res}})$, where Δ_{res} is the ground-truth residual and $\widehat{\Delta_{res}}$ is the predicted residual. Similar to the classification branch, the box residual is also predicted from an aggregated proposal embedding.

The total loss is defined as:

$$\mathcal{L} = \mathcal{L}_{cls} + \alpha \mathcal{L}_{reg}, \tag{2}$$

where α is set to 2. More details can be found in our supplementary materials.

4.5. Post-processing

Remove Empty Predictions. In a typical labeling process, empty bounding boxes will not be annotated even if there indeed exists an object. Our method is capable to predict such boxes using temporal context. To match the GT annotation process, we also remove all the empty predictions after the track-centric learning module.

Track TTA. In our framework, test-time augmentations (TTA) [16] are very simple to implement. The output tracks in different augmentations are naturally aligned according to the time steps. Thus we conduct the simple weighted average on the predictions at each time step by scores. For headings, we adopt the medians instead of the weighted average in case of the "heading flip".

Track coherence optimization (TCO). We can further improve the temporal coherence of tracks by object shape registration. In particular, we specify a box containing a clear object shape, and then align other object shapes to the specified one. We adopt the multi-way registration [7] via pose graph optimization for the alignment. The pose graph has two key elements: nodes and edges. A node is the point cloud P_i associated with a pose matrix M_i which transforms P_i into the specified frame point cloud. For each edge, the transformation matrix $T_{i,j}$ that aligns the source points in P_i to the target points in P_j. We use Point-to-point ICP [1] to estimate the all mentioned transformations. Then the alignment is achieved by optimizing $\mathcal{M} = \{M_i\}$ via:

$$\min_{\mathcal{M}} \sum_{i,j} \sum_{(p,q)\in K_{ij}} \|M_i p - M_j q\|_2^2. \tag{3}$$

In Equation 3, K_{ij} is the set of matched point pairs between $T_{i,j}P_i$ and P_j. Afterwards, we utilize the inverse of the optimized transformations to transform the box centers and heading angles in the canonical box coordinate system. In this way, we align the poses of boxes in a track to the specified frame. More details are in our supplementary materials.

5. Experiments

5.1. Setup

Dataset. Following pioneering work 3DAL [30], we use the large-scale challenging Waymo Open Dataset (WOD) [36] as the testbed. WOD adopts the strict 3D IoU as the matching metric in evaluation, which makes it a very suitable benchmark for our method since one of our objectives is to estimate accurate object poses.

Implementation Details. For the base detector, the training and inference schemes are consistent with the original FSD [9], except that we use more input frames. For the tracker, we adopt the default settings in Immortal Tracker [39]. In our track-centric learning module, the sparse UNet backbone has the same hyper-parameters as the one in FSD. As for data augmentation, we adopt the commonly used global random flip, rotation, and scaling. We also add size and center jittering to the proposals for better robustness. All models are trained with eight RTX 3090 GPUs with batch-size 128, which means there are 16 full-length tracks on each GPU. Due to the limited space, we present more details in the supplementary materials.

5.2. Main Results

Comparison with state-of-the-art detectors. We first compare the CTRL with the state-of-the-art online detectors on the validation set, and the results are shown in Table 1, where CTRL significantly outperforms all existing methods. For the results in the test split, we achieve 82.52 L1 mAPH, surpassing the first-place method by 1.2 mAPH. It is also noteworthy that the *Vehicle/Pedestrian* L1 AP of our method exceeds **90%** with strict IoU-based evaluation metric. We present the detailed results in supplementary materials.

Methods	#. frames	mAP/mAPH L2	*Vehicle* 3D AP/APH L1	*Vehicle* 3D AP/APH L2	*Pedestrian* 3D AP/APH L1	*Pedestrian* 3D AP/APH L2	*Cyclist* 3D AP/APH L1	*Cyclist* 3D AP/APH L2
PV-RCNN++(center) [32]	1	71.7/69.5	79.3/ 78.8	70.6/70.2	81.3/76.3	73.2/68.0	73.7/72.7	71.2/70.2
Graph-RCNN [46]	1	73.2/70.9	80.8/80.3	72.6/72.1	82.4/76.6	74.4/69.0	75.3/74.2	72.5/71.5
FSD [9]	1	72.9/ 70.8	79.2/78.8	70.5/70.1	82.6/ 77.3	73.9/ 69.1	77.1/76.0	74.4/73.3
FlatFormer [22]	3	73.5/72.0	79.7/79.2	71.4/71.0	82.0/76.1	74.5/71.3	77.2/76.1	74.7/73.7
CenterFormer [51]	8	75.1/73.7	78.8/78.3	74.3/73.8	82.1/79.3	77.8/75.0	75.2/74.4	73.2/72.3
INT [43]	10	-/73.6	-/-	-/73.3	-/-	-/71.9	-/-	-/75.6
MPPNet [4]	16	75.6/74.9	82.7/82.3	75.4/75.0	84.7/82.3	77.4/75.1	77.3/76.7	75.1/74.5
FSD++ [12]	7	76.8/75.5	81.4/80.9	73.3/72.9	85.1/82.2	78.2/75.4	81.2/80.3	78.9/ 78.1
DSVT [38]	4	77.5/76.2	82.1/81.6	74.5/74.1	86.0/83.2	79.1/76.4	81.1/80.3	78.8/78.0
CTRL w/o. any TTA	all	83.6/82.1	88.0/87.3	81.7/81.0	88.9/86.1	83.2/80.4	87.7/86.7	85.8/84.8
CTRL w/. Track TTA	all	83.9/82.3	88.5/87.5	82.3/81.3	89.1/86.3	83.3/80.5	87.9/86.9	86.0/85.0

Table 1: Comparison with the state-of-the-art detectors in Waymo Open Dataset validation split.

Tracking results. Table 2 showcases our tracking performance and also outperforms previous arts. Due to the limited space, we report the detailed results in supplementary materials.

Comparison with the previous offline method. 3DAL [30] is the most representative work in offline LiDAR-based 3D object detection. Table 3 shows the comparison results. CTRL consistently outperforms 3DAL in all metrics by large margins. CTRL achieves even better results in strict IoU threshold, which indicates CTRL can better refine the object poses in fine-grained.

	Vehicle MOTA↑	*Vehicle* MOTP↓	*Pedestrian* MOTA↑	*Pedestrian* MOTP↓
AB3DMOT* [41]	55.7	16.8	52.2	31.0
CenterPoint* [49]	55.1	16.9	54.9	31.4
SimpleTrack* [26]	56.1	16.8	57.8	31.3
CenterPoint++ [49]	56.1	-	57.4	-
Immortal Tracker [39]	56.4	-	58.2	-
CTRL (Ours)	71.7	15.0	70.5	29.3

Table 2: Tracking results on WOD validation split (L2). *:from [26].

5.3. Performance Inspection

In this section, we thoroughly investigate our method for a comprehensive understanding in detail.

Overall Analysis We first propose several targeted metrics in addition to Average Precision to better characterize the performance. The three metrics we proposed are the ratio of *Totally missed False Negatives* (T-FNs), the ratio of *High-confidence False Positives* (H-FPs), and the ratio of *High-precision True Positives* (H-TPs). Here, a T-FN is defined as the ground-truth box which does not overlap with any predictions. To avoid being dominated by too many low-confidence predictions, we set the maximum number of predictions in each frame as 200. H-FPs refers to false positives with confidence scores higher than a certain threshold s_t. We use the score s_t at 50% recall to calibrate the different confidence distributions between models. H-TP stands for the predicted box having high IoUs (e.g., 0.9) with the ground truth boxes. Table 4 presents the statistics. Three conclusions can be drawn from it.

IoU Threshold	Method	Vehicle 3D	Vehicle BEV	Pedestrian 3D	Pedestrian BEV
Normal	3DAL [30]	84.5	93.3	82.9	86.3
	Ours	88.5	95.5	89.1	92.3
	Improvement	+4.0	+2.2	+6.2	+6.0
Strict	3DAL [30]	57.8	84.9	63.7	75.6
	Ours	64.9	87.9	73.3	82.4
	Improvement	+7.1	+3.0	+9.6	+6.8

Table 3: **Comparison with state-of-the-art offline detectors on WOD validation split**. The normal/strict thresholds for vehicles are 0.7/0.8, and are 0.5/0.6 for pedestrians.

- The bidirectional tracking significantly reduces the T-FNs in all classes. In 1.03M vehicle objects, only **5.1k (0.48%)** of them are missed by CTRL.
- Although bidirectional tracking increases the number of boxes, it does not introduce extra high-confidence FP (H-FP). We owe it to the track learning module which gives proper confidence to proposals and effectively identifies negative samples.
- The refinement in the track-centric learning greatly boosts the number of high-precision predictions (H-TPs), especially for vehicles which get improved from **46.4%** to **56.6%**.

	Vehicle (∼1.0M GTs)				*Pedestrian* (∼0.45M GTs)			
	L2 AP/APH	T-FN↓	H-FP↓	H-TP↑	L2 AP/APH	T-FN↓	H-FP↓	H-TP↑
Offline FSD	74.9/74.4	21.4k (2.05%)	13.1k (1.25%)	488k (46.4%)	79.5/77.0	10.1k(2.21%)	6.9k (1.52%)	272k (59.2%)
Refine w/. †	80.7/80.1	21.1k (2.00%)	9.2k (0.87%)	591k (56.3%)	81.0/78.3	10.0k (2.20%)	5.6k (1.21%)	280k (61.0%)
Refine w/. forward ext ‡	81.4/80.7	9.1k (0.87%)	8.1k (0.77%)	594k (56.5%)	82.8/80.1	3.7k (0.82%)	5.4k (1.18%)	287k (62.7%)
Refine w/. bidirectional ext ¶	81.7/81.0	5.1k (0.48%)	8.1k (0.77%)	595k (56.6%)	83.2/80.4	2.4k (0.53%)	5.4k (1.18%)	289k (63.1%)

Table 4: **Performance inspection of CTRL**. TTA is not adopted. †: the proposal refinement in track-centric learning module with default tracking results from Immortal Tracker. ‡: Refinement with tracks extended in the forward direction. ¶: Refinement with tracks extended bidirectionally. T-FN: totally missed false negative. H-FP: high-confidence false positive. H-TP: high-precision true positive. The IoU threshold of H-TP is **0.9** BEV IoU for vehicles and **0.7** BEV IoU for pedestrians.

Failure Cases. We then take a closer look at those objects still missed by CTRL. We find the major source of missed objects are those objects with very few points. Statistically, 50% totally missed GTs contain less than **5 points** and 90% of them contain less than **24 points**. Another important characteristic of missed objects is that they are likely to belong to **short-life tracks**. Among all the GT tracks containing more than 10% missed objects, 70% of them have life cycles shorter than **2.5 seconds**. Due to the limited space, we demonstrate more failure cases in the supplementary materials.

5.4. Human Labeling Study

Since the annotation of accurate 3D boxes is much harder than that in 2D, human annotations usually contain errors and are inconsistent with each other. In this section, we try to compare our method with human annotations from several aspects.

Normal cases. Previous art 3DAL [30] reports the human labeling accuracy on five sequences in the WOD validation split. We evaluate our method on the same data, and the results are shown in Table 6. Although 3DAL shows strong performance, it is still worse than human performance in the main metrics. In contrast, CTRL consistently outperforms human performance in all the metrics by large margins. In the strict IoU thresh (0.8), CTRL significantly surpasses human by **5.79 3D AP**.

Hard cases. We further compare our method with human performance on the hard cases. To this end, we ask experienced annotators to label 1000 "hard vehicles" in the WOD validation split carefully. These relabeled GTs are randomly sampled from GTs which has 3D IoU with our predictions lower than 0.7 (official evaluation threshold).[1] Since the box confidence given by annotators may not be accurate and have a considerable impact on the quantitative evaluation, we only calculate the IoU distribution to evaluate the results. Note that we opt for BEV IoU instead of 3D IoU because we observed that official WOD annotations have the incorrect vertical position occasionally.

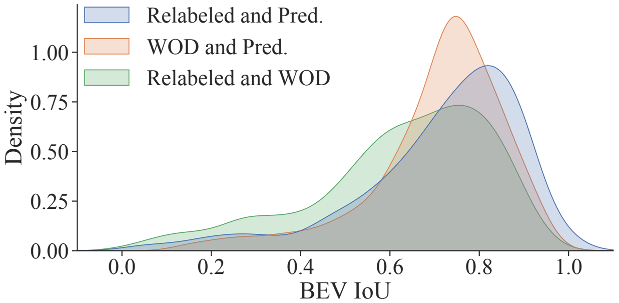

Figure 4: **IoU distribution between each pair of the three kinds of boxes (hard cases only)**. "Relabeled" stands for boxes relabeled by us. "WOD" stands for official WOD annotations. "Pred." stands for the predictions of CTRL. Taking "WOD and Pred." as an example, it means the distribution of IoUs between WOD official annotations and our predictions. The curves are smoothed for clear visualization.

After relabeling, we have three kinds of boxes: official WOD GTs, relabeled boxes, and our predictions. Then we calculate the IoU distribution between each pair of them, shown in Figure 4, where we draw the following two conclusions:

- **Even human annotations are not consistent for hard cases.** The relabeled boxes and WOD annotations have the smallest overall IoU as the green curve indicates, which means these so-called GT annotations may contain noise, especially for hard cases.
- **CTRL has higher quality.** Comparing the blue curve with the green curve, our predictions have better IoU than WOD annotations on these carefully relabeled data, which indicates our method might actually have better performance than the WOD official annotations.

5.5. Ablation Studies

Input track length. In our default setting, the input track length is not limited. Here we set an upper bound for input track length to see how it affects the performance. In Table 5a, $[0, N]$ means that all tracks longer than N will

[1] We present the detailed labeling protocol and examples of hard cases in supplementary materials.

Lengths	Veh.	Ped.
$[0, 50)$	77.1	77.6
$[0, 100)$	78.6	78.0
$[\mathbf{0}, \mathbf{200})$	80.1	78.3

(a) **Track length**. The maximum track length in WOD is 200.

Strategy	Veh.	Ped.
Object-centric	79.6	78.1
Track-centric	80.1	78.3

(b) **Track-centric assignment vs. object-centric assignment**.

Strategy	Veh.	Ped.
Separated	79.8	78.3
Mixed	80.1	78.3

(c) **Separating motion states**.

Mode	Veh.	Ped.	FPS
MISO	79.5	77.7	1.1
MISO 4×	80.0	78.3	1.1
MIMO	80.1	78.3	33.6

(d) **MIMO vs. MISO**. 4× means 4× longer training schedule.

Table 5: **Ablations of track-centric learning**. We report level 2 APH for vehicle and pedestrian. For a clean ablation, here we do not adopt track TTA and bidirectional tracking in the baseline setting. The baseline experiments are marked in **bold**.

	Vehicle 3D AP		*Vehicle* BEV AP	
	IoU=0.7	IoU=0.8	IoU=0.7	IoU=0.8
Human	86.45	60.49	93.86	86.27
3DAL	85.37	56.93	92.80	87.55
Ours	**90.19**	**66.28**	**94.26**	**89.66**

Table 6: **Comparison with human performance**. Evaluation is conducted on relabeled data provided by 3DAL [30].

be broken into several parts. For example, if N is 50, then a 140-frame track will be broken into three tracks with lengths of 50/50/40, respectively. The results in Table 5a suggest that the input length is crucial and the unbounded input length offers CTRL a significant boost.

Track-centric assignment. We compare the proposed track-centric assignment and conventional object-centric assignment in Table 5b. For the object-centric assignment, we follow the commonly-used setting and hyper-parameters in previous work [34, 31, 9], where 3D IoU is adopted as the box matching metric. Specifically, the positive IoU threshold is 0.45 for the vehicle class and 0.35 for the pedestrian class. The results in Table 5b show that track-centric assignment is more suitable in our pipeline since the track-level matching offers more robustness.

Motion agnostic. In our default setting, we do not separate static and dynamic tracks. Here we follow the previous 3DAL [30] to deal with the two kinds of tracks separately. For vehicle class, we regard a track with an average velocity higher than 1.0 m/s as dynamic. For pedestrian class, tracks faster than 0.2 m/s are viewed as dynamic. We train them separately and merge the results together. Table 5c shows that separating them leads to no gains.

MIMO vs. MISO. We then proceed to compare the proposed MIMO with conventional MISO. For adapting MIMO to MISO, we choose a single proposal to supervise in every forward pass. Table 5d shows that MISO needs a much longer training schedule to achieve similar performance to MIMO and is much slower during inference.

Track coherence optimization (TCO). When given one frame high quality human annotation, we could utilize our proposed TCO to align other predictions in the same track to this box. Table 7 shows the results. We demonstrate consistent improvements on all metrics and all quasi-rigid classes. The improvement is especially significant when evaluated at stricter IoU. We also present a visualization of object shapes after TCO in supplementary materials to qualitatively demonstrate the effectiveness.

	Vehicle L1 3D AP		*Cyclist* L1 3D AP	
	IoU=0.7	IoU=0.8	IoU=0.5	IoU=0.6
CTRL	88.5	64.9	87.7	72.7
+TCO	90.2(+**1.7**)	76.5(+**11.6**)	88.9(+**1.2**)	82.7(+**10.0**)

Table 7: The gain from track coherence optimization on WOD validation split. One ground-truth box is given to a track for shape registration.

5.6. Performance Roadmap

As a summary, Table 8 shows how the performance is improved step by step from a state-of-the-art online detector to our best offline setting.

Techniques	L2 3D AP/APH			
	Mean	Vehicle	Pedestrian	Cyclist
Single-frame FSD [9]	72.9/70.8	70.5/70.1	73.9/69.1	74.4/73.3
Multi-frame FSD [12]	76.8/75.5	73.3/72.9	78.2/75.4	78.9/78.1
Offline FSD*	78.8/77.5	74.9/74.4	79.5/77.0	82.0/81.1
+Refine†	81.7/80.3	80.7/80.1	81.0/78.3	83.3/82.5
+Refine w/. Bi‡	83.6/82.1	81.7/81.0	83.2/80.4	85.8/84.8
+Track TTA	83.9/82.3	82.3/81.3	83.3/80.5	86.0/85.0

Table 8: **Performance roadmap from the state-of-the-art online detector to our best offline detector.** $*$: our base detector. †: refinement with default non-extended tracking results. ‡: refinement with bidirectionally extended tracking results.

6. Conclusion

This paper proposed CTRL, a high-performance offline 3D object detection system. CTRL is motivated by human labeling behavior that usually first labels the easy samples in a track, and then propagates the labels to other harder samples. Equipped with bidirectional tracking and track-centric learning, CTRL achieves strong detection performance, surpassing all existing models and human labeling accuracy. Notably, only 0.48% objects are entirely lost by CTRL . We adopt simple design choices in CTRL , making it a good baseline for offline LiDAR-based auto-labeling. We will try to incorporate multi-modality input to further improve the performance in future work.

7. Acknowledgements

This work was supported in part by the Major Project for New Generation of AI (No.2018AAA0100400), the National Natural Science Foundation of China (No. 61836014, No. U21B2042, No. 62072457, No. 62006231), and in part by TuSimple Collaborative Research Project.

References

[1] Paul J Besl and Neil D McKay. Method for Registration of 3-D Shapes. In *Sensor fusion IV: control paradigms and data structures*. Spie, 1992. 2, 5

[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] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating Object Instances with a Polygon-RNN. In *CVPR*, 2017. 2

[4] Xuesong Chen, Shaoshuai Shi, Benjin Zhu, Ka Chun Cheung, Hang Xu, and Hongsheng Li. MPPNet: Multi-Frame Feature Intertwining with Proxy Points for 3D Temporal Object Detection. In *ECCV*. Springer, 2022. 2, 3, 6

[5] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking. In *CVPR*, 2023. 2

[6] Hsu-kuang Chiu, Jie Li, Rareş Ambruş, and Jeannette Bohg. Probabilistic 3D Multi-Modal, Multi-Object Tracking for Autonomous Driving. In *ICRA*. IEEE, 2021. 2

[7] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust Reconstruction of Indoor Scenes. In *CVPR*, 2015. 2, 5

[8] 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 *CVPR*, 2022. 2

[9] Lue Fan, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Fully Sparse 3D Object Detection. In *NeurIPS*, 2022. 2, 4, 5, 6, 8

[10] Lue Fan, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. FSD V2: Improving Fully Sparse 3D Object Detection with Virtual Voxels. *arXiv preprint arXiv:2308.03755*, 2023. 2

[11] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and ZhaoXiang Zhang. RangeDet: In Defense of Range View for LiDAR-Based 3D Object Detection. In *ICCV*, 2021. 2

[12] Lue Fan, Yuxue Yang, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Super Sparse 3D Object Detection. *arXiv preprint arXiv:2301.02562*, 2023. 2, 6, 8

[13] Di Feng, Xiao Wei, Lars Rosenbaum, Atsuto Maki, and Klaus Dietmayer. Deep Active Learning for Efficient Training of a LiDAR 3D Object Detector. In *IV*. IEEE, 2019. 2

[14] Yihan Hu, Zhuangzhuang Ding, Runzhou Ge, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. AFDetV2: Rethinking the Necessity of the Second Stage for Object Detection from Point Clouds. *arXiv preprint arXiv:2112.09205*, 2021. 2

[15] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. BEVDet: High-performance Multi-camera 3D Object Detection in Bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 1

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, 2017. 5

[17] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast Encoders for Object Detection from Point Clouds. In *CVPR*, 2019. 1, 2

[18] Jungwook Lee, Sean Walsh, Ali Harakeh, and Steven L Waslander. Leveraging Pre-Trained 3D Object Detection Models for Fast Ground Truth Generation. In *ITSC*. IEEE, 2018. 2

[19] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. *arXiv preprint arXiv:2006.04388*, 2020. 4

[20] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. BEVFormer: Learning Bird's-eye-view Representation from Multi-camera Images via Spatiotemporal Transformers. In *ECCV*. Springer, 2022. 1

[21] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. PETR: Position Embedding Transformation for Multi-view 3D Object Detection. In *ECCV*. Springer, 2022. 1

[22] Zhijian Liu, Xinyu Yang, Haotian Tang, Shang Yang, and Song Han. FlatFormer: Flattened Window Attention for Efficient Point Cloud Transformer. *arXiv preprint arXiv:2301.08739*, 2023. 6

[23] Chenxu Luo, Xiaodong Yang, and Alan Yuille. Exploring Simple 3D Multi-Object Tracking for Autonomous Driving. In *ICCV*, 2021. 2

[24] Qinghao Meng, Wenguan Wang, Tianfei Zhou, Jianbing Shen, Luc Van Gool, and Dengxin Dai. Weakly Supervised 3D Object Detection from Lidar Point Cloud. In *ECCV*. Springer, 2020. 2

[25] 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 *CVPR*, 2019. 2

[26] Ziqi Pang, Zhichao Li, and Naiyan Wang. Simpletrack: Understanding and Rethinking 3d Multi-object Tracking. In *ECCVW*. Springer, 2023. 2, 6

[27] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep Hough Voting for 3D Object Detection in Point Clouds. In *ICCV*, 2019. 2

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*, 2017. 2

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NeurIPS*, 2017. 2

[30] 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 *CVPR*, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[31] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In *CVPR*, 2020. 1, 2, 4, 8

[32] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection. *arXiv preprint arXiv:2102.00463*, 2021. 2, 6

[33] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In *CVPR*, 2019. 2

[34] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 4, 5, 8

[35] Colton Stearns, Davis Rempe, Jie Li, Rareş Ambruş, Sergey Zakharov, Vitor Guizilini, Yanchao Yang, and Leonidas J Guibas. SpOT: Spatiotemporal Modeling for 3D Object Tracking. In *ECCV*. Springer, 2022. 2

[36] 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 *CVPR*, 2020. 4, 5

[37] 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 *CVPR*, 2021. 2

[38] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. DSVT: Dynamic Sparse Voxel Transformer with Rotated Sets. *arXiv preprint arXiv:2301.06051*, 2023. 6

[39] Qitai Wang, Yuntao Chen, Ziqi Pang, Naiyan Wang, and Zhaoxiang Zhang. Immortal Tracker: Tracklet Never Dies. *arXiv preprint arXiv:2111.13672*, 2021. 2, 3, 5, 6

[40] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. DETR3D: 3D Object Detection from Multi-view Images via 3D-to-2D Queries. In *CoRL*. PMLR, 2022. 1

[41] Xinshuo Weng and Kris Kitani. A Baseline for 3D Multi-object Tracking. *arXiv preprint arXiv:1907.03961*, 2019. 2, 6

[42] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M Kitani. GNN3DMOT: Graph Neural Network for 3D Multi-object Tracking With 2D-3D Multi-feature Learning. In *CVPR*, 2020. 2

[43] Jianyun Xu, Zhenwei Miao, Da Zhang, Hongyu Pan, Kaixuan Liu, Peihan Hao, Jun Zhu, Zhengyang Sun, Hongmin Li, and Xin Zhan. INT: Towards Infinite-frames 3D Detection with An Efficient Framework. In *ECCV*. Springer, 2022. 2, 6

[44] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 18(10), 2018. 1, 2

[45] 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

[46] Honghui Yang, Zili Liu, Xiaopei Wu, Wenxiao Wang, Wei Qian, Xiaofei He, and Deng Cai. Graph R-CNN: Towards Accurate 3D Object Detection with Semantic-Decorated Local Graph. In *ECCV*. Springer, 2022. 6

[47] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3DSSD: Point-based 3D Single Stage Object Detector. In *CVPR*, 2020. 2

[48] Zetong Yang, Yin Zhou, Zhifeng Chen, and Jiquan Ngiam. 3D-MAN: 3D Multi-Frame Attention Network for Object Detection. In *CVPR*, 2021. 2

[49] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3D Object Detection and Tracking. *arXiv preprint arXiv:2006.11275*, 2020. 1, 2, 6

[50] Sergey Zakharov, Wadim Kehl, Arjun Bhargava, and Adrien Gaidon. Autolabeling 3D Objects with Differentiable Rendering of SDF Shape Priors. In *CVPR*, pages 12224–12233, 2020. 2

[51] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. CenterFormer: Center-based Transformer for 3D Object Detection. In *ECCV*. Springer, 2022. 2, 6