LiDAR MOT-DETR: A LiDAR-based Two-Stage Transformer for 3D Multiple Object Tracking

Martha Teiko Teye[1,2]
m.teye-hk@uni-wuppertal.de
Ori Maoz[2]
ori.maoz@aptiv.com
Matthias Rottmann[3]
matthias.rottmann@uos.de

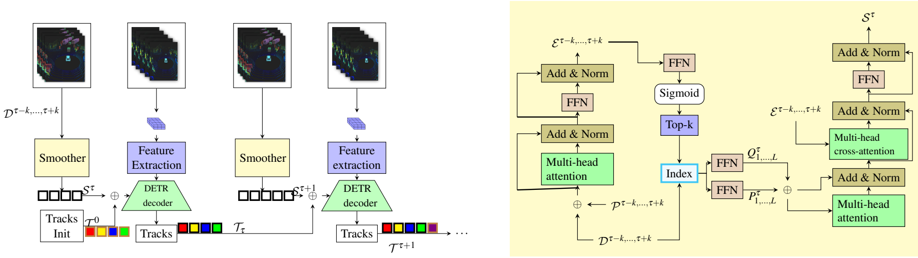
[1] School of Mathematics and Natural Sciences
University of Wuppertal, Germany
[2] Aptiv Services Deutschland GmbH
Wuppertal, Germany
[3] Department of Mathematics, Computer Science and Physics
Osnabrück University, Germany

Abstract

Multi-object tracking from LiDAR point clouds presents unique challenges due to the sparse and irregular nature of the data, compounded by the need for temporal coherence across frames. Traditional tracking systems often rely on hand-crafted features and motion models, which can struggle to maintain consistent object identities in crowded or fast-moving scenes. We present a lidar-based two-staged DETR inspired transformer; a smoother and tracker. The smoother stage refines lidar object detections, from any off-the-shelf detector, across a moving temporal window. The tracker stage uses a DETR-based attention block to maintain tracks across time by associating tracked objects with the refined detections using the point cloud as context. The model is trained on the nuScenes and KITTI datasets in both online and offline (forward peeking) modes demonstrating strong performance across metrics such as ID-switch and multiple object tracking accuracy (MOTA). The numerical results indicate that the online mode outperforms the lidar-only baseline and SOTA models on the nuScenes dataset, with an aMOTA of 0.724 and an aMOTP of 0.475, while the offline mode provides an additional 3 pp aMOTP.

1 Introduction

Multi-object tracking (MOT) involves the detection and association of multiple objects across frames in an image or video sequence. MOT is very useful in applications such as automated driving, traffic control [35], video surveillance [1] and animal behaviour studies [53]. The ability to accurately track multiple objects is a cornerstone of these systems, particularly in the context of self-driving vehicles and advanced robotics. LiDAR technology (Light Detection and Ranging) has become a key sensor in these domains due to its ability to provide precise 3D measurements of the environment [15, 34]. Numerous works have been conducted in 2D camera [3, 51, 55, 58], radar domains [27, 50] and some extensions to 3D [44, 54].



(a) Overall architecture (b) Smoother selection mechanism

Figure 1: LiDAR MOT-DETR works by first selecting relevant detections across a temporal window, k, using our Smoother mechanism. Relevant objects are then jointly tracked using detection queries, track queries and PC features in an iterative transformer track process. Brown background represents new tracks whiles existing tracks with black background.

However, MOT using LiDAR pointcloud (PC) data presents several challenges: inaccurate motion models [48], complex post-processing, life cycle management and re-identification, short-term track identification of objects and inconsistent ego-motion of the sensor platform [37, 43]. For example, in the field of autonomous driving, most object tracking approaches require enormous (and tiring) optimization and hyperparameter tuning effort [10, 55], with the resulting tracker not being robust to changes in the data [44].

Early (classical) tracking methods, which are still popular today, rely on explicit motion modelling as dynamical systems (for an introduction see e.g. [28]). These approaches, which includes Kalman Filter [45] and particle filter [20], suffer from imperfect appearance and motion models [18, 42] and are sensitive to the choice of hyper-parameters. They also strongly depend on ego-motion information (to compensate for induced motion in the sensors reference frame) that is often estimated separately and can be inefficient and error-prone [45].

Neural networks have become a powerful tool for MOT due to their ability to learn robust feature representations [9, 11, 21, 22]. Hybrid approaches that incorporate deep learning focused on modelling spatial and motion features inherently within the networks have improved the efficiency of object tracking [22, 39, 45], though their performance largely depends on the performance of the underlying object detectors [40]. Transformers [36], originally developed for natural language processing, have proven remarkably versatile across various domains, including computer vision and, more recently, pointcloud processing [13]. Their ability to model long-range dependencies and handle sequences of varying lengths makes them well-suited for the task of MOT in LiDAR data [37, 57].
Our main contributions are as follows:

- We introduce LiDAR MOT-DETR, a modular, detector-agnostic (i.e. works with any existing object detector), two-stage framework which can be tailored for both online and offline tracking.
- Both smoother and tracker architectures are based on the DETR transformer, tailored specifically to the needs of smoothing and tracking.
- Our tracker outperforms current state-of-the-art (SOTA) lidar models with an aMOTA of 0.724 and aMOTP of 0.475 on nuScenes validation set in an online setting. Even more significantly, it achieves an aMOTP of 0.445 on the nuScenes test set in offline mode.

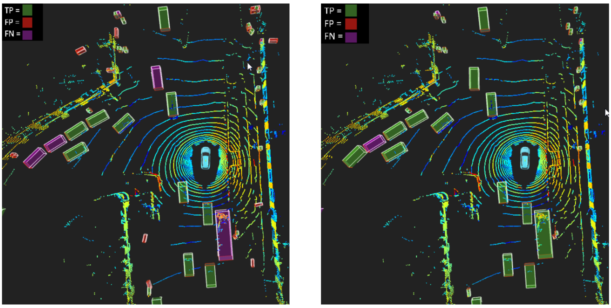
2 Related Work

LiDAR-based Multiple Object Tracking has seen significant developments in recent years, particularly with the incorporation of deep learning methods [13, 15, 16]:

Introduction of Multiple Object Tracking. Traditionally, MOT relied heavily on probabilistic models such as the Kalman Filter [28] and its variations, which use explicit motion modelling and heuristic track assignments such as greedy and Hungarian assignment [45]. Kalman Filters enabled efficient data association, allowing for the matching of detected objects across frames. However, they often struggled with occlusions, dynamic environments, and complex object interactions. Deep learning methods, particularly Convolutional Neural Networks (CNNs), have transformed MOT by enabling robust feature extraction and learning from large datasets [21, 33] in both 2D and 3D. Approaches like FairMOT [55] and Deep SORT [45] extended Kalman Filters by combining CNN-based object detection features with Recurrent Neural Networks (RNNs) for temporal modelling, achieving state-of-the-art performance in image-based MOT. However, these methods are ineffective with LiDAR data due to its sparse and irregular nature [48].

LiDAR-based Multiple Object Tracking. LiDAR sensors provide precise 3D point clouds that are invaluable for understanding the spatial arrangement of objects in an environment [15]. Early approaches to LiDAR-based MOT often relied on voxelization or point-wise feature extraction methods [2], followed by traditional tracking algorithms like the Extended Kalman Filter (EKF) [12, 45]. These methods were limited by their reliance on hand-crafted features and their inability to capture the complexity of object dynamics in 3D space. Recent advancements have seen the integration of deep learning models directly with LiDAR point clouds. Yin et al. [49] introduced CenterPoint, a voxel-based framework that detects objects and tracks using greedy closest-point matching in 3D space. CenterPoint demonstrates that leveraging spatial features directly from point clouds significantly improves tracking accuracy. Utilizing further features from these detection models have provided recent deep learning trackers with enriched depth and spatial information which has been a baseline for most deep learning detectors [30] and trackers [12, 33, 45].

Transformers in LiDAR Multiple Object Tracking. In MOT, transformers have been utilized to enhance temporal consistency in tracking [8, 11]. Wang et al. [41] presented a strong sensor-fusion transformer-based framework for egomotion-aware 3D object tracking, which directly models the motion of objects and the sensor platform. Ruppel et al. [32] developed a transformer-based tracker that builds on previous point-based detection systems [37]. The inclusion of transformers enhanced the model's ability to use rich query feature representations to track objects using attention-based mechanism [7, 39, 51, 54] which is used in camera and lidar domain. However, deep learning approaches often fall short with handling new-born trajectories [51], mangaging bounding box precision, complex post-processing methods [51], managing identity switches and the track life cycle [11]. In this paper, we focus on inherently tracking objects using transformers without the need for heuristics track life cycle management.



3DMOTFormer [11] (Ours)

Figure 2: Qualitative comparison between our method and another SOTA method on nuScenes validation set (scene_token: 01452fbfbf4543af8acdfd3e8a1ee806). Both methods use the same base object detector (CenterPoint [49]).

3 Methodology

We now introduce our novel two-stage DETR [6] transformer-based architecture, consisting of a smoother and a tracker as depicted in figure 1. The smoother takes the bounding boxes from the object detector across multiple frames (either past & future frames, or past only) and outputs cleaner and more consistent bounding boxes. The tracker takes the track predictions from the previous frames, the detected objects from the current frame and the raw point cloud, and outputs track predictions for the current frame.

Overview of Tracking and Smoothing. We start with a series of LiDAR point clouds and corresponding $\mathcal{D}^\tau = \{d_1^\tau, \dots, d_{n^\tau}^\tau\}$ sets of detected objects, from an existing (off-the-shelf) object detector, for each frame $\tau = 1, \dots, T$, with n^τ being the number of detections in frame τ. We assume that n^τ is bound by above by some $n \in \mathbb{N}$.

Each d_i^τ consists of a predicted class label, 3D bounding box center, extents and rotation angles. For $\tau \notin \{1, \dots, T\}$, let $D^\tau = \emptyset$. For some $k \in \mathbb{N}$, the smoother f processes $t' - t + 1$ frames $\mathcal{D}^{t,\dots,t'} = (\mathcal{D}^t, \dots, \mathcal{D}^{t'})$, providing smoothed detections for frame τ as output, i.e., $\mathcal{S}^\tau = f(\mathcal{D}^{t,\dots,t'})$. We set $t = \tau - k$ as well as $t' = \tau + k$, such that $\mathcal{D}^{\tau-k,\dots,\tau+k}$ contains $2k+1$ frames with the key frame τ right at the centre of the sequence. The corresponding loss function ℓ_f for learning the smoothing is discussed below.

The tracker g predicts the tracked objects in each frame based on the tracks of the previous frames, the detected objects from the current frame and the raw point cloud. It creates a set of track queries which includes the tracks from the previous frame $\mathcal{T}^{\tau-1}$ and newly-initialized tracks from any detection from $\mathcal{S}^\tau$ which could not be matched to the existing tracks. It outputs bounding boxes $\mathcal{T}^\tau = \{p_i, s_i, \theta_i, C_i, tid_i\}_{i=1,\dots,N^\tau}$ consisting of the center position, extent, yaw rotation, score and track id, respectively, with $N^\tau \leq \tau \cdot n$ denoting the total number of bounding boxes in frames $1, \dots, \tau$.

Discussion of Loss Functions. The smoother compares the smoothed detections $\mathcal{S}_i^\tau$ to the ground-truth $\mathcal{G}_j^\tau$, $i, j = 1, \dots, n$, while ignoring their ordering. It uses the same bipartite matching as proposed in DETR (DEtection TRansformers) [6]. Suppose we have n_s smoothed detection and n_g ground-truth boxes in frame τ. Let $B \subseteq \{(i,j) : i = 1, \dots, n_s,\ j = 1, \dots, n_g,\}$ be a bipartite matching such that for an elements B_ι and $B_{\iota'}$ with $\iota \neq \iota'$ with their two slots fulfilling $B_{\iota,\kappa} \neq B_{\iota',\kappa}$ for $\kappa = 1,2$. We express the loss functions that we use as:

$$\ell_f = \arg\min_B \sum_{(i,j)\in B} L_{match}(\mathcal{S}_i^\tau, \mathcal{G}_j^\tau) \ , \quad \text{where } L_{match} = W_{bbox}L_{bbox} + W_{cls}L_{cls} \tag{1}$$

L_{bbox} denotes the L1 loss which can also be expressed as $L_{bbox} = L_{box}(\mathcal{S}_i^\tau, \mathcal{G}_j^\tau)$ and L_{cls} denotes the focal loss which also represents the association loss [23]. The loss weights are denoted by W_{bbox} and W_{cls}, respectively. The tracker uses the same loss as in smoother, with an additional L_{giou} [31] generalized intersection over union loss:

$$\ell_g = \arg\min_B \sum_{(i,j)\in B} L'_{match}(\mathcal{T}_i^\tau, \mathcal{G}_j^\tau) \ , \quad \text{where } L'_{match} = W'_{bbox}L_{bbox} + W'_{cls}L_{cls} + L_{giou} \tag{2}$$

where the corresponding loss weights are denoted by W'_{bbox} and W'_{cls}, respectively. The generalised intersection over union loss helps in enhancing the precision of the tracker's 3D bounding box regression for better alignment.

Architectural Details of the Smoother. Noise from LiDAR detections often stems from environmental factors such as occlusions, reflective surfaces, and sensor artifacts. Therefore, it is essential to design pre-processing techniques that filter out unreliable detections while maintaining valuable object information. The smoother serves as a pre-processing step to fine-tune detections and make up for some missed predictions in-between frames.

Thus, all detections over a period k of a moving window of length are denoted by $\mathcal{D}^{\tau-k,\dots,\tau+k}$. The transformer encoder processes these input detections $\mathcal{D}^{\tau-k,\dots,\tau+k}$ for the input sequence $\mathcal{P}^{\tau-k,\dots,\tau+k}$, to represent a relationship of each detection to other detections in the sequence in the form of embeddings. We define an embedding function, $\mathcal{E}^{\tau-k,\dots,\tau+k} := \phi\left(\mathcal{D}^{\tau-k,\dots,\tau+k}\right)$ that maps the detections $\mathcal{D}^{\tau-k,\dots,\tau+k}$ to a d-dimensional embedding, and the embedding set for all detections with number N, within the time window.

The model encodes historical and future detections into a unified embedding space, containing $\mathcal{E}^{\tau-k,\dots,\tau+k}$, using a multi-head self-attention transformer encoder mechanism allowing it to learn long-term dependencies and patterns as shown in Figure 1 (b). The generated embeddings are used to create object queries $Q^\tau_{1,\dots,L}$ and its corresponding positional encoding $P^\tau_{1,\dots,L}$ using a Top-K selection filter mechanism used by [29], which selects the top scoring embeddings. We denote the number of tokens selected by Top-K as L. Each output query embedding $Q_i^\tau \in \mathbb{R}^d$, $i = 1,\dots,L$, is computed as a non-linear transformation:

$$Q_i^\tau = \sigma(W_2\text{ReLU}(W_1\mathcal{E}_i^t + b_1) + b_2), \tag{3}$$

where W_1 and W_2 are learnable transformation matrices, b_1 and b_2 are learnable bias terms, σ is a nonlinearity (ReLU). Finally, the set of query embeddings at frame τ is, $Q^\tau_{1,\dots,L} = (Q_1^\tau,\dots,Q_L^\tau) \in \mathbb{R}^{L\times d}$. The selected query embeddings is used by the DETR decoder with self-attention layers to fine-tune the bounding boxes, object classifications and association scores of the current frame improving the stability of detections. Finally, we apply a feed forward network to regress the objects $\mathcal{S}^\tau$ belonging to the current frame τ.

Architectural Details of the Tracker. Our proposed tracking method integrates a transformer-based architecture module for both online and offline multiple object tracking in the LiDAR domain. The core of our approach is a transformer architecture designed to accept tracked objects from the previous frame $\tau-1$, detected objects from the current frame τ and features of the point clouds, and output tracked objects in the current frame.

For each frame, we start by taking the smoothed detections $\mathcal{S}^\tau$ and comparing (matching) them to the tracks from the previous frame $\mathcal{T}^{\tau-1}$. Any detections without a match are used to initialize new tracks, resulting in a set of track queries $\mathcal{Q'}^\tau$ that include both tracks from the previous frame and newly-proposed tracks from the detections. The tracker accepts as input the track queries $\mathcal{Q'}^\tau$, the detected objects $\mathcal{S}^\tau$ and features $\mathcal{F}^{\tau-k+1,\dots,\tau} = (\mathcal{F}^{\tau-k+1},\dots,\mathcal{F}^\tau)$ extracted from the point clouds to generate outputs $\mathcal{T}^\tau$ which are tracked objects at time τ.

$$\mathcal{T}^\tau = g(\mathcal{S}^\tau, \mathcal{Q'}^\tau, \mathcal{F}^{\tau-k+1,\dots,\tau}) \tag{4}$$

At its core, the tracker is simply an attention block where the queries are the track queries $\mathcal{Q}^\tau$, the keys are the smoothed detections S^τ and the values are the point cloud features $\mathcal{F}^{\tau-k+1,\dots,\tau}$. To achieve this, we first need to apply an encoder to the track queries $\mathcal{Q'}^\tau_{feat} = encoder(\mathcal{Q'}^\tau)$ and the detected objects $\mathcal{S}^\tau_{feat} = encoder(\mathcal{S}^\tau)$:

$$\mathcal{T}^\tau_{feat} = \text{Attention}(\mathcal{Q'}^\tau_{feat}, \mathcal{S}^\tau_{feat}, \mathcal{F}^{\tau-k+1,\dots,\tau}) \tag{5}$$

the attention block essentially searches for detection features ($\mathcal{S}^\tau_{feat}$) which are similar to our track queries (features), with the point cloud features provided as context, and outputs them as the new track features. From these features we decode the tracked objects in the current frame as $\mathcal{T}^\tau = decoder(\mathcal{T}^\tau_{feat})$. The *encoder* maps $\mathcal{S}^\tau \rightarrow \mathcal{S}^\tau_{feat}$ by means of a multi-layer perceptron (MLP) and positional encoding for spatial awareness. The *decoder* also has an MLP which is applied to the output new track query: $\mathcal{T}^\tau = \sigma(W_b \mathcal{T}^\tau_{feat} + b_b)$, where W_b and b_b are learnable parameters and σ is the sigmoid function, ensuring normalized outputs.

After decoding the features back into tracks $\mathcal{T}^\tau$, we filter out tracks with a low predicted confidence. The high-confidence tracks comprise the output of the tracker, and are also fed forward as track queries for the next frame. The decoded object from the same query should be representing the same object across frames, thus forming a whole tracklet. To train the tracker, we need to assign one target ground-truth object for each query in each frame, and represent the assigned ground-truth object as the regression target for each object. Our mapping is defined as a one-to-one mapping where every predicted track is assumed to have only one corresponding ground-truth. In frames where the tracks exceed the ground-truth, we pad ground-truth with empty/null objects to ensure the one-to-one mapping is achieved. This streamlined design reduces computational overhead while improving the robustness and efficiency of LiDAR-based MOT. Utilizing track queries, we are able to better handle data associations thereby tracking long-term with less identity mismatches [57].

To obtain the point cloud features, we apply the VoxelNet [59] voxelization to a stack of point clouds in the range $\tau-k+1,\dots,\tau$. The weights of the feature encoder are also learned during training. These point cloud features provide context to the attention blocks so that an object which appears in multiple frames can be tracked across these frames based on the extracted LiDAR-based features.

Handling Occlusions and Ambiguities. We maintain a history buffer [51] for each tracked object, storing information from past frames. This buffer enables predictive tracking when objects temporarily disappear due to occlusion or sensor limitations. When new detections emerge in subsequent frames, their state is fused with historical data to maintain continuity. The fusion process prioritizes recent observations while incorporating historical states to

Method	aMOTA↑	aMOTP↓	MOTA↑	MOTP ↑	MT↑	ML↓	TP↑	FP↓	FN↓	IDS↓	FRAG↓
MotionTrack-L [52]	0.51	0.99	0.48	0.30	3723	1567				9705	
CenterPoint [49]	0.638	0.555	0.537	0.284	5584	1681	95877	18612	22928	760	529
SimpleTrack [28]	0.668	0.550	0.537	0.284	5584	1681	95877	18612	22928	575	529
UG3DMOT [17]	0.668	0.538	0.549	0.310	5468	1776	95704	19401	22955	906	653
ImmortalTrack [38]	0.677	0.599	0.572	0.285	5565	1669	97584	18012	21661	320	477
3DMOTFormer [11]	0.682	0.496	0.556	0.297	5466	1896	95790	18322	23357	438	529
NEBP [22]	0.683	0.624	0.584	0.300	5428	1993	97367	16773	21971	**227**	**299**
ShaSTA [33]	0.696	0.540	0.578	0.295	5596	1813	**97799**	**16746**	21293	473	356
VoxelNeXt [9]	0.710	0.511	0.600	0.308	5529	1728	97075	18348	21836	654	537
FocalFormer3D [8]	0.715	0.549	**0.601**	0.309	5615	**1550**	97535	16760	**21142**	888	810
Offline Track [24]	0.671	0.522	0.553	0.296	**5658**	1713	96617	16778	22378	570	592
LiDAR MOT-DETR (online)	0.724	0.475	0.594	0.310	5415	2063	97315	18446	21846	404	528
LiDAR MOT-DETR (offline)	**0.726**	**0.445**	0.592	**0.312**	5561	1892	97593	18695	21495	577	462

Table 1: Overall results of LiDAR-only methods on the nuScenes test set based on Nuscenes Leaderboard. Best value highlighted in bold and second place underlined.

prevent drift and abrupt positional changes. By aggregating information from multiple (10) LiDAR sweeps, the tracker can partially "see" occluded objects from alternate viewpoints. This ensures that tracking remains robust even in complex scenes. The model integrates both appearance (3D shape) and motion (position, and trajectory) information over time to differentiate objects that may look similar but behave differently.

4 Experiments

LiDAR MOTR-DETR uses detections from off-the shelf object detectors and hence the performance of our method is influenced by the underlying detector. Thus, for a fair comparison, we evaluate the performance with other tracking methods using the same object detector. We relied on two popular detectors used in previous papers [11], CenterPoint [49] and Focalformer3D-L [8]. We evaluate our results on both the nuScenes and Kitti public datasets. Further description of our dataset and data preprocessing steps can be found in the attached supplementary material. Although our smoother method primarily focused on an offline mode, the tracker is completely online as it uses point clouds from present and past frames only and the output of the smoother per key frame. To provide a fairer comparison with other online methods, we also trained an online variant of the smoother. All ablations were evaluated using the validation splits.

4.1 Training and Inference.

For better generalization and faster training, we pre-trained the smoother and tracker components on unannotated data, which was auto-ground-truthed using the CenterPoint model and a Kalman-based tracker. This unannotated data is from an private dataset collected with a different lidar sensor (Velodyne VLS-128). We later used the pre-trained checkpoints as a starting point when training our model on nuScenes [5] and KITTI [14]. For the CenterPoint and FocalFormer3D object detectors used in our results, we used the model checkpoints released by the authors [8, 49].
The smoother network is trained using detections with a temporal window of 15, ($k = 7$) for 24 epochs by AdamW optimizer. The initial learning rate (LR) is set to 5e-4. This tracker

Method	Detector	Modality	mAP↑	mATE↓	mASE↓	mAOE↓	mAAE↓	NDS↑
CenterPoint	CenterPoint	L	0.564	-	-	-		0.648
Ours (Online Smoother)	CenterPoint	L	0.637	0.255	**0.21.0**	0.364	0.138	0.695
Ours (Offline Smoother)	CenterPoint	L	0.671	0.253	0.214	0.342	0.135	0.711
FocalFormer3D [8]	FocalFormer	L	0.664	-	-	-	-	0.709
Ours (Online Smoother)	FocalFormer	L	0.672	**0.249**	0.245	**0.331**	**0.125**	**0.725**
Ours (Offline Smoother)	FocalFormer	L	**0.683**	0.251	0.242	0.337	0.133	0.717

Table 2: Smoother results on nuScenes validation set with temporal window of 15, when run on top of two different object detectors. L represents models trained using lidar pointclouds.

is trained on multi-sweep LiDAR frames (sweep size = 10), plus the output detections of the smoother, for 32 epochs using the AdamW optimizer, LR of 4e-4 and bs of 6. The loss weights in equation 2 are set as $W_{bbox} = 0.5$, $W_{cls} = 1.0$ with $\alpha = 0.5$ and $\gamma = 2$. For the tracker, we set $W'_{bbox} = 0.25$, $W'_{cls} = 2.0$ with $\alpha = 0.25$ and $\gamma = 2$.
Inference. During inference, our model processes raw inputs from multiple LiDAR sweeps, updates the track queries on a frame-by-frame basis, and outputs a complete set of object tracks. The relevant tracks are selected using a confidence-based pruning method $\mathcal{T}^\tau = \{t_i^\tau \mid C_i^\tau < \tau_c, t_i^\tau \in T^{\tau-1}\}$, where τ_c is the termination confidence threshold (0.2) and C is the track score. Inactive tracks are terminated when they exceed a predefined maximum of allowed frames. In this experiment, inactive tracks are terminated after 5 frames. This ensures that objects missing for several consecutive frames are eventually removed.

(a) offline

Total Frames	aMOTA↑	aMOTP↓	IDS↓	FRAG↓	mAP↑
8	0.682	0.559	442	506	0.632
11	0.705	0.556	425	500	0.657
13	0.726	0.538	419	480	0.670
15	**0.735**	**0.523**	**407**	**474**	**0.671**

(b) online

Total Frames	aMOTA↑	aMOTP↓	IDS↓	FRAG↓	mAP↑
8	0.677	0.557	457	502	0.629
11	0.698	0.542	428	498	0.635
13	0.713	0.532	424	497	0.634
15	**0.729**	**0.529**	**410**	**462**	**0.637**

Table 3: Comparison of tracker performance with different temporal window sizes of Smoother. Results are generated using CenterPoint detections, on nuScenes validation set in (a) offline and (b) online modes.

4.2 Primary Evaluation Metrics

MOT are commonly evaluated on two main metrics: Average Multiple Object Track Accuracy (aMOTA), which measures how well the tracker detects objects in the scene, and Multiple Object Track Precision (aMOTP) [4], which measures how well the tracker's bounding boxes fit the objects. On Kitti, the performance is evaluated using Higher Order Tracking Accuracy (HOTA) [25], both MOTA and MOTP [4]. Additionally, we compare the different methods by identity switches (IDS/IDSW) [4, 10] and track fragmentation (FRAG) [4], as well as a breakdown of how many of the objects were tracked across at least 80% of their lifetime (Mostly Tracked, MT) or at most 20% of their lifetime (Mostly Lost, ML) [10]. Any tracks which lie in between the MT and ML are considered Partially Tracked (PT) objects. Table 1 compares the performance of LiDAR MOT-DETR with baselines and state-of-the-art deep tracking methods on the nuScenes dataset. We observed that the aMOTP is significantly better than other methods, at 0.475 (compared to 0.549 for SOTA FocalFormer3D-L), while aMOTA remains slightly above SOTA. Thus, while LiDAR MOT-DETR's tracker accuracy is better than other methods (with an aMOTA of 0.724, compared to 0.715 of SOTA), its

Name	Sensor	HOTA	DetA	IDSW	MOTA
UG3DMOT [17]	L	0.808	0.782	13	0.866
MCTRACK [41]	L+C	0.839	-	3	0.64
BiTrack [19]	L+C	0.845	**0.819**	13	0.878
RobMOT [26]	L	0.863	-	1	0.915
CasTrack [47]	L	0.932	-	-	-
PC-TCNN [46]	L	0.944	-	3	0.886
LeGO [56]	L	**0.952**	-	1	0.90
Ours (Online)	L	0.852	0.817	12	0.913
Ours (Offline)	L	0.894	0.802	8	0.916

Table 4: Results on KITTI validation dataset.

Method	Detector	aMOTA↑	aMOTP↓	IDS↓	FRAG↓
CenterPoint [49]	Centerpoint	0.637	0.606	-	-
VoxelNeXt [9]	End-to-End	0.702	0.640	729	-
NEBP [22]	CenterPoint	0.708	-	172	271
3DMOTFormer [22]	CenterPoint	0.712	0.515	341	436
ShaSTA [33]	CenterPoint	0.728	-	-	-
Ours (w/o Smoother)	CenterPoint	0.681	0.579	435	501
Ours (Two-stage)	CenterPoint	<u>0.735</u>	0.523	407	474
FocalFormer3D-L [8]	FocalFormer	0.721	-	-	-
Ours (w/o Smoother)	FocalFormer	0.729	0.486	398	485
Ours (Two-stage)	FocalFormer	**0.752**	**0.479**	**332**	**454**

Table 5: Results on nuScenes validation set with and without smoother in comparison with other methods.

Detector	Pre-training	Smoother	aMOTA↑	aMOTP↓	IDS↓	FRAG↓
CenterPoint	✗	✗	0.672	0.581	456	521
CenterPoint	✗	✓	0.685	0.573	438	493
CenterPoint	✓	✗	0.681	0.579	435	501
CenterPoint	✓	✓	<u>0.735</u>	0.523	407	474
FocalFormer-L	✗	✗	0.728	0.541	419	489
FocalFormer-L	✗	✓	0.736	0.532	392	<u>468</u>
FocalFormer-L	✓	✗	0.729	0.486	398	485
FocalFormer-L	✓	✓	**0.752**	**0.479**	**332**	**454**

Table 6: Ablation study on the nuScenes validation set; (1) with and without pre-training; (2) with and without smoother.

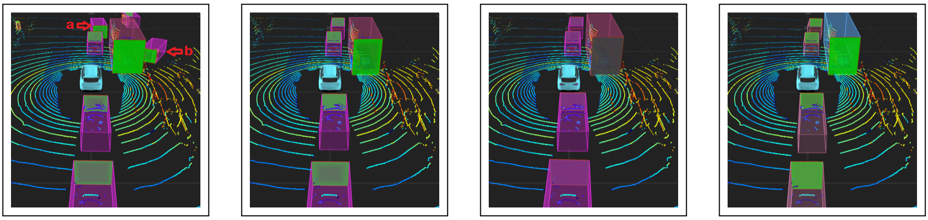
main contribution is in its refinement of the predicted bounding boxes as seen in figures 2 and 3. Although we do not outperform LiDAR SOTA [19, 26, 41, 46, 47, 56] on KITTI dataset, we see a very competitive performance in HOTA of 0.822 in table 4.

Smoother design choice – offline vs. online. We found that the offline (forward-peeking) variant of the smoother outperforms the online variant. It achieved an NDS score (nuScenes Detection Score – weighted sum of multiple detection metrics [43]) of 0.711, as opposed to 0.695 for the online variant (table 2). Both of these significantly outperform the 0.648 score of the CenterPoint detector itself. Use cases such as auto ground-truthing, which do not need to run in real-time, would benefit from this increased performance.

Length of smoother temporal window. We found that the performance of the tracker increases as we increase the size of the smoother's temporal window, in both the online and the offline (forward-peeking) variants, as observed in the aMOTA and aMOTP (table 3). When fixing the window size to 15 frames, the forward-peeking variant outperforms the online variant by 1.7 percent points (pp).

Effect of base object detector used. We evaluated our method on both FocalFormer3D [8], a SOTA object detector, and CenterPoint [49], an older object detector. We found that a large 8.4 pp aMOTA difference between the detectors themselves was reduced to only 1.7 pp post-tracking in (table 5) with our two stage method. Therefore, the gains provided by LiDAR MOT-DETR are higher when used with older/underperforming object detectors.

Effect of smoother model. Additionally, we evaluated the performance of the tracker with and without the underlying smoother model. Table 5 shows that the performance gain provided when applying the smoother to the CenterPoint detections (5.4 pp) is much larger than



(detections) (smoother) (tracker) (ground-truth)

Figure 3: Visual representation on nuScenes validation set. In (detections), we show arrow a, an object with wrong orientation and facing the wrong (opposite) direction from object detector. Arrow b shows false positive from detector. We recommend zooming in.

the gain when applying it to the FocalFormer3D detections (2.3 pp). Thus, much of the gain which LiDAR MOT-DETR gives to the older object detectors comes from the smoothing step, not necessarily the tracker. This highlights the importance of the smoother model in our approach, as it helps maintain internal consistency of bounding boxes. An example of this is illustrated in Figure 3, where we see how poorly-detected bounding boxes from the raw detector are corrected by the smoother and tracker.

Effect of pre-training. In table 6, we demonstrate the advantage and performance boost we get as a result of pre-training our model (excluding the detector) on a large-scale auto ground-truthed proprietary dataset. When training our model solely on nuScenes, we see a 1.3 pp improvement in aMOTA when incorporating the smoother instead of when not. However, we achieve an additional 5 pp in aMOTA when pre-training our model on the proprietary dataset. This shows, although the proprietary dataset rather consists of automatically generated ground-truth, our smoother and tracker clearly benefit from large volumes of data.

In summary, we have observed that our two-stage smoother and tracker exceeds state-of-the-art tracker performance on the nuScenes dataset. In particular, it significantly outperforms SOTA methods in the aMOTP metric, which reflects the accuracy of the bounding box. We believe that the smoother, which integrates detections across a window temporal window, is a major contributor for this. The same method can be used either in an online or forward-peeking variation; this allows offline use cases, such as auto ground-truthing, to take advantage of the increased performance in forward-peeking mode.

5 Conclusion

Neural networks have revolutionized the field of multi-object tracking, providing powerful tools for robust and accurate tracking in complex scenarios. We present a novel two-staged approach to multiple object tracking in the LiDAR domain, utilizing Deformable DETR-based architectures, which uses existing off-the-shelf object detectors. Our tracker outperforms current state-of-the-art LiDAR-based MOTs by 0.9 pp aMOTA and 10.5 pp aMOTP on nuScenes validation sets. without extensive manual parameter tuning or post-processing of the obtained tracks. This shows that the integration of transformers allows for the effective modelling of temporal dependencies. Our extensive evaluation demonstrates that this approach outperforms state-of-the-art methods in both accuracy and precision.

References

[1] Hyochang Ahn and Han-Jin Cho. Research of multi-object detection and tracking using machine learning based on knowledge for video surveillance system. *Personal and Ubiquitous Computing*, 26(2):385–394, 2022.

[2] Masoud S Bahraini, Ahmad B Rad, and Mohammad Bozorg. Slam in dynamic environments: A deep learning approach for moving object tracking using ml-ransac algorithm. *Sensors*, 19(17):3699, 2019.

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 941–951, 2019.

[4] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008: 1–10, 2008.

[5] 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 multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *CoRR*, abs/2005.12872, 2020.

[7] Xuesong Chen, Shaoshuai Shi, Chao Zhang, Benjin Zhu, Qiang Wang, Ka Chun Cheung, Simon See, and Hongsheng Li. Trajectoryformer: 3d object tracking transformer with predictive trajectory hypotheses, 2023.

[8] Yilun Chen, Zhiding Yu, Yukang Chen, Shiyi Lan, Anima Anandkumar, Jiaya Jia, and Jose M Alvarez. Focalformer3d: focusing on hard instance for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8394–8405, 2023.

[9] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21674–21683, June 2023.

[10] P Dendorfer. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020.

[11] Shuxiao Ding, Eike Rehder, Lukas Schneider, Marius Cordts, and Juergen Gall. 3dmotformer: Graph transformer for online 3d multi-object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9784–9794, October 2023.

[12] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, 25:8725–8737, 2023.

[13] Gopi Krishna Erabati and Helder Araujo. Li3detr: A lidar based 3d detection transformer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4250–4259, 2023.

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[15] Mahmudul Hasan, Junichi Hanawa, Riku Goto, Ryota Suzuki, Hisato Fukuda, Yoshinori Kuno, and Yoshinori Kobayashi. Lidar-based detection, tracking, and property estimation: A contemporary review. *Neurocomputing*, 506:393–405, 2022.

[16] Mahmudul Hasan, Junichi Hanawa, Riku Goto, Ryota Suzuki, Hisato Fukuda, Yoshinori Kuno, and Yoshinori Kobayashi. Lidar-based detection, tracking, and property estimation: A contemporary review. *Neurocomputing*, 506:393–405, 2022. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2022.07.087.

[17] Jiawei He, Chunyun Fu, Xiyang Wang, and Jianwen Wang. 3d multi-object tracking based on informatic divergence-guided data association. *Signal Processing*, 222: 109544, 2024.

[18] Peixuan Hu, Ruichen Cai, Witold Mlodzikowski, and Guan Huang. Monocular 3d object detection and tracking using class-aware motion estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 864–874, 2021.

[19] Kemiao Huang, Yinqi Chen, Meiying Zhang, and Qi Hao. Bitrack: Bidirectional offline 3d multi-object tracking using camera-lidar data, 2025. URL https://arxiv.org/abs/2406.18414.

[20] Hojoon Lee, Hyunsung Lee, Donghoon Shin, and Kyongsu Yi. Moving objects tracking based on geometric model-free approach with particle filter using automotive lidar. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):17863–17872, 2022.

[21] Xiaoyu Li, Dedong Liu, Lijun Zhao, Yitao Wu, Xian Wu, and Jinghan Gao. Fast-poly: A fast polyhedral framework for 3d multi-object tracking. *arXiv preprint arXiv:2403.13443*, 2024.

[22] Mingchao Liang and Florian Meyer. Neural enhanced belief propagation for data association in multiobject tracking. In *2022 25th International Conference on Information Fusion (FUSION)*, pages 1–7. IEEE, 2022.

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. doi: 10.1109/TPAMI.2018.2858826.

[24] Xianzhong Liu and Holger Caesar. Offline tracking with object permanence. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 1272–1279. IEEE, 2024.

[25] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixe, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision (IJCV)*, 2020.

[26] Mohamed Nagy, Naoufel Werghi, Bilal Hassan, Jorge Dias, and Majid Khonji. Robmot: Robust 3d multi-object tracking by observational noise and state estimation drift mitigation on lidar pointcloud. *arXiv preprint arXiv:2405.11536*, 2024.

[27] Zhijun Pan, Fangqiang Ding, Hantao Zhong, and Chris Xiaoxuan Lu. Ratrack: Moving object detection and tracking with 4d radar point cloud. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4480–4487. IEEE, 2024.

[28] Ziqi Pang, Zhichao Li, and Naiyan Wang. Simpletrack: Understanding and rethinking 3d multi-object tracking. In *European Conference on Computer Vision*, pages 680–696. Springer, 2022.

[29] Juliano Pinto, Georg Hess, William Ljungbergh, Yuxuan Xia, Henk Wymeersch, and Lennart Svensson. Can deep learning be applied to model-based multi-object tracking? *ArXiv*, abs/2202.07909, 2022.

[30] C. Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa T. Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3d object detection from point cloud sequences. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6130–6140, 2021.

[31] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression, 2019. URL https://arxiv.org/abs/1902.09630.

[32] Felicia Ruppel, Florian Faion, Claudius Gläser, and Klaus Dietmayer. Transformers for multi-object tracking on point clouds. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 852–859, 2022. doi: 10.1109/IV51971.2022.9827344.

[33] Tara Sadjadpour, Jie Li, Rares Ambrus, and Jeannette Bohg. Shasta: Modeling shape and spatio-temporal affinities for 3d multi-object tracking. *IEEE Robotics and Automation Letters*, 2023.

[34] Peize Sun, Yibing Cao, Yi Jiang, Rufeng Zhang, Ping Luo, Xiaogang Wang, and Wenyu Liu. Transtrack: Multiple-object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020.

[35] Wei Tian, Martin Lauer, and Long Chen. Online multi-object tracking using joint domain information in traffic scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):374–384, 2020. doi: 10.1109/TITS.2019.2892413.

[36] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[37] Hao Wang, Peiyun Sun, Thien Hoang, Ali Harakeh, and Steven L. Waslander. Pointtracknet: An end-to-end network for 3d object detection and tracking from point clouds. *arXiv preprint arXiv:2012.02353*, 2020.

[38] Qitai Wang, Yuntao Chen, Ziqi Pang, Naiyan Wang, and Zhaoxiang Zhang. Immortal tracker: Tracklet never dies. *ArXiv*, abs/2111.13672, 2021. URL https://api.semanticscholar.org/CorpusID:244709700.

[39] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection, 2023.

[40] Xiyang Wang, Chunyun Fu, Jiawei He, Mingguang Huang, Ting Meng, Siyu Zhang, Hangning Zhou, Ziyao Xu, and Chi Zhang. You only need two detectors to achieve multi-modal 3d multi-object tracking. *arXiv preprint arXiv:2304.08709*, 2023.

[41] Xiyang Wang, Shouzheng Qi, Jieyou Zhao, Hangning Zhou, Siyu Zhang, Guoan Wang, Kai Tu, Songlin Guo, Jianbo Zhao, Jian Li, and Mu Yang. Mctrack: A unified 3d multi-object tracking framework for autonomous driving. *ArXiv*, abs/2409.16149, 2024.

[42] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019.

[43] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10359–10366, 2020. doi: 10.1109/IROS45743.2020.9341164.

[44] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6499–6508, 2020.

[45] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. *ICIP*, pages 3645–3649, 2017.

[46] Hai Wu, Qing Li, Chenglu Wen, Xin Li, Xiaoliang Fan, and Cheng Wang. Tracklet proposal network for multi-object tracking on point clouds. In *IJCAI*, pages 1165–1171, 2021.

[47] Hai Wu, Jinhao Deng, Chenglu Wen, Xin Li, Cheng Wang, and Jonathan Li. Casa: A cascade attention network for 3-d object detection from lidar point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2022. doi: 10.1109/TGRS.2022.3203163.

[48] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11495–11504, 2020.

[49] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11784–11793, June 2021.

[50] Matthias Zeller, Daniel Casado Herraez, Jens Behley, Michael Heidingsfeld, and Cyrill Stachniss. Radar tracker: Moving instance tracking in sparse and noisy radar point clouds. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16170–16177. IEEE, 2024.

[51] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *European Conference on Computer Vision*, pages 659–675. Springer, 2022.

[52] Ce Zhang, Chengjie Zhang, Yiluan Guo, Lingji Chen, and Michael Happold. Motiontrack: End-to-end transformer-based multi-object tracking with lidar-camera fusion. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 151–160, 2023. URL https://api.semanticscholar.org/CorpusID:259286967.

[53] Libo Zhang, Junyuan Gao, Zhen Xiao, and Heng Fan. Animaltrack: A benchmark for multi-animal tracking in the wild. *International Journal of Computer Vision*, 131(2): 496–513, 2023.

[54] Tianyuan Zhang, Xuanyao Chen, Yue Wang, Yilun Wang, and Hang Zhao. Mutr3d: A multi-camera tracking framework via 3d-to-2d queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4537–4546, June 2022.

[55] Yifu Zhang, Chunyu Wang, Xinggang Wang, and Wenjun Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129(11):3069–3087, 2021.

[56] Zhenrong Zhang, Jianan Liu, Yuxuan Xia, Tao Huang, Qing-Long Han, and Hongbin Liu. Lego: Learning and graph-optimized modular tracker for online multi-object tracking with point clouds. *arXiv preprint arXiv:2308.09908*, 2023.

[57] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H.S. Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16259–16268, October 2021.

[58] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490. Springer, 2020.

[59] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *CoRR*, abs/1711.06396, 2017.