Click, Crop & Detect: One-Click Offline Annotation for Human-in-the-Loop 3D Object Detection on Point Clouds

Nitin Kumar Saravana Kannan, Matthias Reuse, Martin Simon
Valeo Schalter und Sensoren GmbH
Kronach, Germany
{nitin.kannan, matthias.reuse, martin.simon}@valeo.com

Abstract

Recent cutting-edge methods for 3D object detection on point clouds are based on supervised learning methods. As these methods demand an extreme volume of data with the highest quality to train on, cost-effective annotation plays a crucial role in developing such perception algorithms, e.g., for autonomous vehicles or robots. Every inconsistency or error between the data captured by sensors and the subsequently generated labels might degrade the potential detection performance. Nevertheless, resources for annotation are usually very limited in terms of budget and time. We propose a straightforward yet highly effective technique called Click, Crop, and Detect (CCD) to address this issue. The core concept of CCD involves leveraging human input first to generate a prior rough localization of each object and employing 3D object detectors on a simplified cropped region of interest. We evaluate CCD across popular detectors such as PointPillars, CenterPoint, and TED on nuScenes and KITTI. Here, we show that only marginal changes to existing off-the-shelf detectors are required to make them compatible. Our method consistently outperforms state-of-the-art one-click detectors by 7.89% and 10.45% for cars and pedestrians, respectively, while being much more robust and precise on challenging, sparse inputs. This heavily increases label quality and efficiency when applied for semi-automated ground truth annotation.

1. Introduction

In recent years, the annotation of 3D objects in point clouds and its automation has received significant attention. High-quality 3D datasets are among the most essential sources for the ongoing development of environmental perception algorithms, e.g., in Autonomous Driving, as most of these methods are based on deep neural networks [26]. In other words, a vast amount of labeled data samples are needed to train and optimize such methods. Nonetheless, those samples must also be labeled accurately to avoid inconsistencies affecting the resulting model performance. In addition, the entire annotation process is very limited in terms of time and cost, while, in contrast, the latest dataset generations are on a completely different scale in terms of size [1, 45]. A handful of approaches are trying to reach full automation with only little or no supervision. Still, they are usually prone to errors in their labels, suffering from inaccuracies of the underlying algorithms [57]. Therefore, the majority of existing labels in this domain are partially or even completely manually generated by skilled human annotators. A lot of research has been done on making the job of the annotator easier, including automatic annotation, single click approaches, or research on UI aspects of the tool [20, 50, 51]. On the one hand, if the tool is completely manual annota-

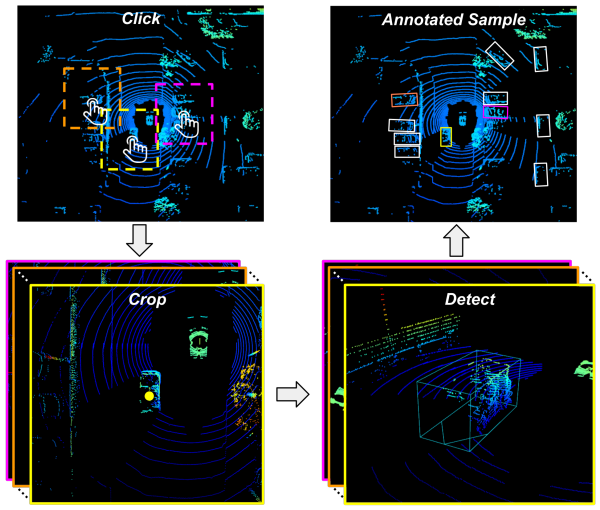

Figure 1. CCD Overview: The annotator clicks in the 3D point cloud on the center of an object he wishes to annotate. The point cloud is now cropped around the annotator's click and processed by the 3D object detector. This is repeated for each object in the scene, resulting in accurate 3D bounding boxes for the whole point cloud.

tion, with no automation, time is lost in the process of creating the labels. On the other hand, reliability is always an issue if the tool annotates automatically without any human in the loop. Therefore, a human annotator would have to check the annotations individually anyway and make necessary corrections. Moreover, [19] shows that the multiple tasks of the annotators slow them down and, at the same time, make them more prone to errors. Experiments done by [75] show that it took their best annotator 2 hours 30 minutes, 4418 clicks, and 1488 keystrokes to annotate 395 cars from a sequence in the nuScenes dataset [4]. Thus, we find an ideal trade-off would be a quick annotation tool that smartly interacts with a human in the loop.

To this end, we propose Click, Crop, and Detect (CCD), a semi-automated labeling pipeline aiming to alleviate the difficulty of ground truth annotation using both a human annotator and machine learning algorithms to provide accurate 3D ground truth boxes quickly and easily (see Figure 1). By initially transferring the task of localizing objects across a whole scene to a skilled human annotator, we significantly simplify the remaining task for the algorithm, reducing the overall manual effort in the best case to only one single click to annotate a full 3D object. At the same time, we show this design allows the reuse of existing 3D detectors with minimal effort while boosting their performance by 7.32% for cars and 38.4% for pedestrians. We focus on optimizing the performance of one-click detection throughout this paper, but our proposed method can easily be plugged into tools with even more sophisticated annotation features.

In summary, our main contributions are the following:

- We propose to customize 3D object detectors for one-click detection that are used for semi-automated annotation. We set a new state-of-the-art, showing the effectiveness of our design.
- We introduce a mathematical model to simulate human click behavior using existing annotations and practically compare them with actual clicks. It can be used to adapt 3D object detectors and remove the need to gather human clicks for existing datasets.
- We conduct a comprehensive analysis and demonstrate that individual difficulties in the 3D detection task are successfully transferred to humans, leading to significantly higher performance than baseline 3D detectors and existing one-click methods.
- We provide rich insights with additional experiments to study the effects of the number of points per object and further click simulation models.

2. Related Work

2.1. 3D Object Detection on Point Clouds

Current 3D object detectors are usually categorized according to the structure of how they process the point clouds, namely point-based and grid-based. In many cases, point-based methods [35, 40, 42, 43, 60, 61] originate from PointNet [33, 34] for feature extraction. In contrast, grid-based approaches first transform the point clouds into 3D voxels [6, 9, 27, 54, 56, 63, 65, 70–72], pillars [18, 52], bird's-eye view representations [44, 58], or range-images [7, 28, 46], heavily inspired by related architectures from vision. Hybrid approaches [3, 15, 41, 60, 73] are also utilized trying to leverage both advantages. More recently, transformer-based architectures also show great potential featuring the attention mechanism [8, 14, 25, 30, 39, 47, 64]. In addition, multi-frame detectors [5, 16, 62, 65, 74] can achieve a significantly increased performance by concatenating multiple point cloud inputs.

2.2. Ground Truth Annotation on Point Clouds

Research in ground truth annotation on point clouds aims to mitigate the expenses and can be broadly classified based on the level of human interaction. First, approaches with little or no human supervision, sometimes called off-board detection [10, 24, 36, 55, 57, 59], utilizing the temporal context with trajectory level refinement, offline tracking, and multi-frame detectors. Then, there are works which concentrate on easing the task of manual annotation by user-friendly interfaces [2, 20, 50, 75] and numerous assistance techniques exploring self-, weak-supervision or active learning [11, 19, 31, 32, 66–69].

Semi-automated methods automate certain parts of the annotation process but often require complete human attention. Here, the majority of work concentrates on single-click annotation employing different techniques. LATTE [50] uses a clustering-based method for one-click annotation by fitting a rectangle with the points around the click and predicting a 2D box in bird's-eye view space. SAnE [2] improves LATTE by using a denoising pointwise segmentation method to eliminate the need for sensitive ground plane removal. SUSTechPOINTS [20] places box prototypes around the recorded human click, followed by an Euclidean distance-based growing algorithm to cover all the points of the object and an auto-fitting algorithm to shrink the box to its actual size. Similarly, [19] proposes a three-stage processing based on handcrafted box templates using elements from PointNet [33]. Furthermore, [29] leverages YOLOv3 [37] for detection in bird's-eye view space.

Inspired by the abovementioned one-click methods, we propose replacing all handcrafted and multi-stage processing steps with just a single 3D detector sequentially applied to the region of interests defined by single clicks. In this way, a simple processing chain with fewer restrictions or assumptions is ensured while utilizing all the advantages of modern detectors. Plus, simplifying the inputs also leads to remarkable improvements in performance and is therefore better suited to automate labeling.

3. Click, Crop, and Detect (CCD)

The core idea of CCD is to significantly reduce the complexity of the input point clouds from scene level to object level through interaction with a human annotator. By having the human annotator click on the 3D point cloud, the model gets prior information on where to look for the object. Our method consists of three sequential steps: getting the prior of an object center by a human click or sampled from a random distribution (see subsection 3.1), cropping the region of interest centered to this prior (see subsection 3.2), and, finally, detecting the underlying object (see subsection 3.3). To this end, we aim for minimal modification to easily adapt existing detectors. An overview of the entire pipeline is shown in Figure 2.

3.1. Click

Given the height of the origin of the point cloud relative to the ground z_{pcl}, we aim to obtain pivot points $p = (x_p, y_p, -z_{pcl})$, $p \in \mathbb{R}^3$ close to the object centers. Obtaining the needed prior coordinates (x_p, y_p) can be divided into two paths: human-in-the-loop and simulation.

Human-in-the-loop. A human annotator is tasked to click on the perceived center of a target object in a bird's-eye view rendered on a screen. The resulting (u, v) coordinates are then back-projected into 3D using the orthographic projection matrix M by $p = M^{-1}[u, v, -z_{pcl}, 1]^T$, omitting the homogeneous dimension. To support this, annotation tools usually provide further features like image views or other automation features (see subsection 2.2). However, it is not necessary to click very precisely on the center as the following processing can handle some offsets. Note that errors from quantisation are also ignored.

Simulation. Obtaining clicks for every object across a whole dataset is a very tedious and costly job. However, for training and validation of the object detectors such clicks are necessary in a large variety to get the best results. Yet, inspired by [17] existing ground truth objects can be reused as an approximate value instead. This allows a simulation of human clicks. Given a ground truth box with center (c_x, c_y, c_z) and yaw rotation Ψ, we consider two variations to simulate a pivot point p close to that from human clicks. For the first variation, we draw

$$(x_p, y_p) \sim \mathcal{N}\big([c_x, c_y]^T, R(\Psi)\Sigma R(\Psi)^T\big) \tag{1}$$

from the normal distribution $\mathcal{N}$, where R describes a two-dimensional rotation, and

$$\Sigma = \begin{bmatrix} (\frac{a}{3})^2 & 0 \\ 0 & (\frac{b}{3})^2 \end{bmatrix}$$

with a and b referring to the maximum offsets to the box center in the two main axes respectively. To prevent too far away outliers, we omit p if

$$\frac{((x_p - c_x)\cos(\Psi) + (y_p - c_y)\sin(\Psi))^2}{a^2} + \frac{((x_p - c_x)\sin(\Psi) - (y_p - c_y)\cos(\Psi))^2}{b^2} > 1\,, \tag{2}$$

so if it is outside the 3σ confidence interval ellipse. Thus, the simulated points are oriented along the heading of the ground truth box Ψ.

As an alternative, we consider a rotation $R(\alpha)$ instead of $R(\Psi)$ with $\alpha = \arctan2(c_y, c_x)$. Thereby, the simulated points are oriented along the line of sight between the origin of the point cloud and the object center.

3.2. Crop

Based on a pivot point p, the region of interest Ω can be selected. All points from the input point cloud $\mathcal{P} \in \mathbb{R}^3$ within the volume around (x_p, y_p) are cropped, while all other points are discarded:

$$\mathcal{P}_\Omega = \{[x, y, z, i]^T \in \mathcal{P} \quad | \quad x \in [x_p - \frac{w}{2}, x_p + \frac{w}{2}], \quad y \in [y_p - \frac{h}{2}, y_p + \frac{h}{2}]\}\,, \tag{3}$$

where w is the width and h the height of the cropped region. Both can be defined per object class taking various aspects into account. For instance, they can be aligned with the architecture and configuration of the subsequent detector, i.e. the number of downsampling operations or the detection range, as well as potential augmentations during training to ensure full coverage of the target object (see subsection 4.1 for more details).

3.3. Detect

Finally, the cropped set of points P_Ω is sent to a 3D object detector to estimate the bounding box accurately. We utilize the well-established detectors PointPillars [18], CenterPoint [65], and TED [54] as baselines. Here, only minor changes in the configuration are needed to allow their usage even for the two-stage approach TED while all architectural details remain.

Preliminary. PointPillars (PP) [18] divides the point clouds into pillars, i.e. vertical columns along the z-axis. An encoder learns the features from the stacked vectors of these pillars. Lastly, the detection head [23] utilizes the extracted features to predict 3D bounding boxes.

CenterPoint (CP) [65] proposes representing 3D objects as points, i.e., first detect the object center and then estimate the size, orientation, and velocity through regression. Subsequently, these estimations are refined using additional point features associated with the object.

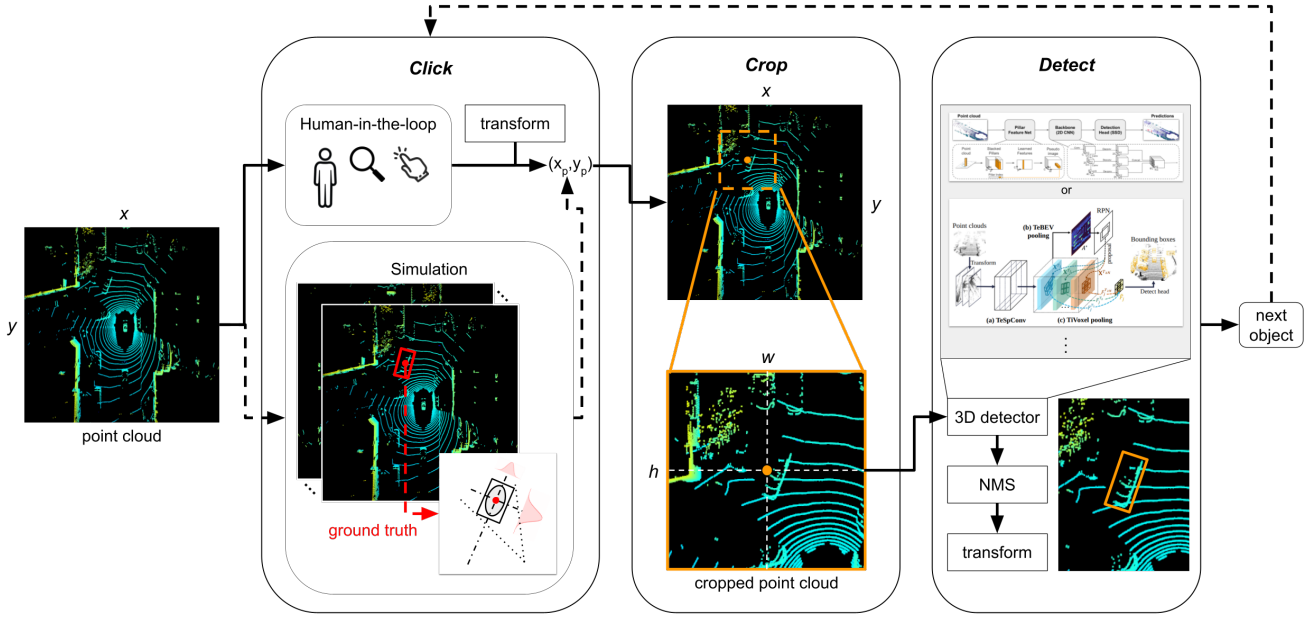

Figure 2. Overall architecture of CCD. First, (x_p, y_p) coordinates referencing an object center are generated either manually by an annotator or sampled from a distribution based on ground truth. Second, this point is used to crop the input point cloud. Finally, a 3D detector [18, 54] outputs a 3D bounding box, which is then transformed back into point cloud coordinates. The workflow is repeated for all objects.

TED [54] first applies a shared feature encoding on multiple transformed point clouds. Cross-grid attention is utilized to align and aggregate those features into a lightweight representation followed by a detection head.

Modifications to the one-click setting. Compared to the originals, the input voxel resolution is reduced due to the smaller detection range defined by w and h. At the same time, this allows to maintain smaller cell sizes dropping fewer points for more descriptive features. Furthermore, all multi-heads are removed except the one for the particular class. This is perfectly aligned with the annotation use case, as different models or even sets of weights can be used without any issues. Finally, we filter the predictions using only the one with the highest confidence. Only these small adjustments are necessary to significantly impact detection performance. This also ensures that emerging models can easily be integrated into our method.

4. Experiments

Numerous experiments are conducted to assess the performance of our models across various datasets and metrics. We also compare our performance quantitatively and qualitatively with existing one-click annotation methods. Additionally, ablation studies are conducted to gain further insights and investigate the performance changes with the number of points per object and different distributions and shapes utilized in the click simulation.

4.1. Experimental Setup

Dataset and Metrics. We use the two popular large-scale autonomous driving datasets nuScenes [4] and KITTI [13] with more than 1.4M and 80k 3D bounding boxes, respectively. All experiments are carried out on the official train and valid splits for nuScenes, and for KITTI, we follow recent work [54].

Following the datasets and competing one-click methods [2, 19, 50], two criteria are used to consider predictions as positive: the Intersection over Union (IOU) with varying thresholds of 0.25, 0.5, 0.7, and using the centroid distance with a threshold of 0.5 meters. We do not use any other benchmark-specific conditions unless otherwise mentioned. The Average Precision (AP) is utilized to evaluate the models for both criteria.

While the nuScenes Detection Score (NDS) [4] has demonstrated a strong correlation with driving performance in evaluating 3D object detection models [38], it is not fully applicable here. Among its six components, velocity error and attribute error are not relevant, as they rely on temporal information regarding objects and the ego vehicle.

Implementation details. Our baselines and models are re-implemented from the OpenPCDet [48] and the TED [53] code bases. We evaluate for two classes: cars and pedestrians. In the one-click setting, only one sweep of the nuScenes LiDAR point cloud is utilized, whereas baseline detectors leverage multiple sweeps. Separate models are

Model	Dataset	Car					Pedestrian				
		IOU >0.7		IOU >0.5		mIOU↑	IOU >0.5		IOU >0.25		mIOU↑
		AP↑ (%)	Recall↑	AP↑ (%)	Recall↑		AP↑ (%)	Recall↑	AP↑ (%)	Recall↑	
PP [18]	Full	62.60	0.71	86.40	0.91	0.76	25.20	0.51	52.30	0.79	0.46
CP [65]	Full	68.30	0.76	88.90	0.94	0.75	38.20	0.59	70.80	0.93	0.39
TED [54]	Full	74.80	**0.80**	88.80	0.94	0.78	54.00	0.75	67.90	0.91	0.52
One-click setting											
LATTE [50]*	*Partial*	-	-	(78.80)	(0.85)	**(0.83)**	-	-	-	-	-
SAnE [2]*	*Partial*	-	-	-	(0.81)	(0.74)	-	-	-	-	-
Leveraging [19]*	Full	-	-	88.33	-	0.70	-	-	88.73	-	0.47
CCD PP	Full	54.00	0.62	94.60	**0.96**	0.72	53.10	0.70	97.70	0.99	0.56
CCD CP	Full	48.60	0.62	89.00	0.92	0.71	50.80	0.68	**98.00**	**0.99**	0.53
CCD TED	Full	**78.10**	**0.80**	**95.30**	0.95	0.79	**70.50**	**0.78**	90.80	0.92	**0.62**

Table 1. Performance of the baseline 3D detectors, one-click models, and CCD on the KITTI valid dataset based on different IOU metrics. Results marked with * are from the respective papers. Values in parentheses denote results for BEV IOU metric, while others are 3D IOU.

adopted for each class, as the annotator can quickly recognize and select which object they will annotate. The model was also evaluated for the simultaneous handling of both classes. However, the results are less satisfactory than when the classes are trained separately. Also, employing different models offers flexibility in optimizing voxel sizes, point cloud ranges, and detectors tailored to achieve optimal performance for each object class. Further, object samples are randomly rotated across 360° for augmentation. An average car in the nuScenes dataset has a length and width of 4.65 and 1.95 meters, respectively. To have a realistic dimension of the ellipse centered at the car, we chose $a = 1$ and $b = 0.5$ meters. For pedestrians, we used a circle of radius $a = b = 0.3$ meters. For the nuScenes dataset, the detection range is constrained to ± 50 meters for cars and ± 40 meters for pedestrians, consistent with the evaluation parameters in nuScenes [4], while no such limits are used for KITTI.

4.2. Comparison with State-of-the-art One-Click Detectors

As explained in subsection 2.2, one-click annotation has been previously attempted using traditional clustering algorithms. Table 1 compares the performance of the baseline 3D object detectors and the one-click methods with CCD on the KITTI dataset. While LATTE and SAnE have only used selected portions of KITTI to validate their model, Leveraging and our CCD models use the entire valid subset of the KITTI dataset. Our models not only clearly outperform the traditional one-click methods but also exhibit notable performance improvements compared to when used as global detectors. This improvement can be attributed to the model's enhanced localization, facilitated by the prior information provided by the human click.

The challenge with using a traditional clustering algorithm arises when the points fail to provide sufficient information about the object's shape. This issue is particularly pronounced in LiDAR point clouds, characterized by sparsity and frequent occlusion, making clustering methods less effective. Our experience with these algorithms does not reflect the results claimed in their work or as shown in their examples. This can be well seen in Figure 3. All models were run on the same data samples, and their respective one-click methods were evaluated. It's evident that predictions from clustering-based algorithms are heavily influenced by the visible points, resulting in shapes much smaller than the ground truth. In contrast, CCD attempts to estimate the object even in case of partial occlusion, leveraging the knowledge it has gained from training data.

Another peculiarity with LiDAR point clouds is the prevalence of ground points. These points must be removed for the clustering algorithm to work as expected. While SAnE utilizes a denoising pointwise segmentation strategy and SUSTechPoints filters out points below 0.2 meters in the z-axis, the ground plane removal of LATTE is not publicly available. Nevertheless, after trying various methods, we found that a RANSAC-based algorithm [12] yields the best results for removing ground points before applying the clustering algorithm provided by LATTE. In contrast, our deep learning-based approach does not vary regarding ground point removal.

Practical difficulties with other methods. Three notable aspects are worth mentioning: firstly, LATTE provides only 2D bounding boxes, whereas other models provide a 3D bounding box. Secondly, our model's results are obtained using a simulated click with added noise. In contrast, the other models are simulated to be clicked precisely at the center of the ground truth box. Lastly, for the SUSTech-

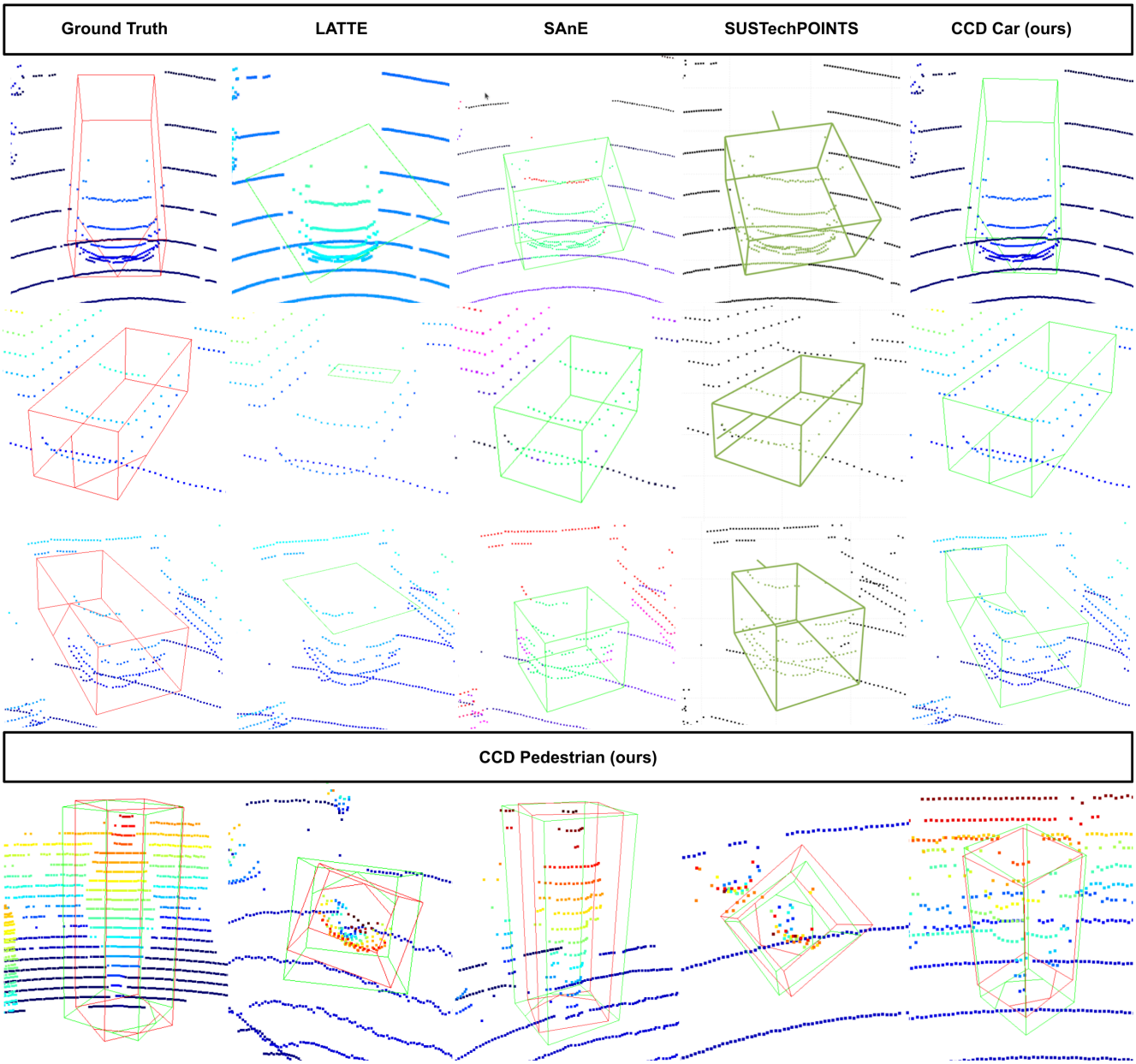

Figure 3. Qualitative comparison of the predictions on samples from nuScenes valid using LATTE [49, 50], SAnE [2, 22], SUSTechPOINTS [20, 21], and CCD. Note that LATTE only outputs a 2D box in bird's-eye view space. The last row shows predictions (green) from CCD and ground truth (red) of different samples from pedestrians.

POINTS model, the point cloud requires rotation to ensure the target object faces either upward or downward, as specified in their instructions [21]. Although they rely on a machine learning model to predict the yaw, their model's performance suffers significantly without this axis alignment. Our approach avoids this additional effort for the human annotator, as only a single click is required, albeit close to but not necessarily at the center.

4.3. Ablation Studies

Number of points per object. The performance of the detection model is significantly influenced by the number of points as shown in Figure 4. It can be observed that the average precision of the models improves at different IOU thresholds with a higher number of points per object.

Table 2 compares the average precision (AP), average translation error (ATE), average scale error (ASE), and average orientation error (AOE) of global detector models

Model	Car				Pedestrian			
	AP↑ (%)	ATE↓	ASE↓	AOE↓	AP↑ (%)	ATE↓	ASE↓	AOE↓
PP [18]	72.18	0.186	0.159	0.102	73.57	0.157	0.282	**0.393**
CCD PP	97.10	0.181	0.170	0.066	**100.00**	0.121	0.346	0.615
CCD PP 15	98.80	0.154	0.157	0.042	**100.00**	0.121	0.304	0.565
CCD PP 30	**99.50**	0.138	0.150	**0.035**	**100.00**	0.121	0.294	0.547
Improvement	+27.32	+0.048	+0.009	+0.067	+26.43	+0.036	-0.012	-0.154
CP [65]	71.87	0.189	0.154	0.222	78.40	0.173	0.277	0.441
CCD CP	95.90	0.180	0.169	0.088	**100.00**	0.087	0.294	0.675
CCD CP 15	98.20	0.148	0.154	0.062	99.90	**0.081**	0.280	0.840
CCD CP 30	99.40	**0.132**	**0.149**	0.051	99.80	**0.081**	**0.272**	0.813
Improvement	+27.53	+0.057	+0.005	+0.171	+21.50	+0.092	+0.005	-0.372

Table 2. Performance of baseline 3D detectors against our one-click models on the nuScenes valid dataset using the 0.5 meters centroid distance metric. CCD is tested on objects with a minimum number of 5, 15, and 30 points with the assumption of always having even more points when annotating a sequence of point clouds. Note that there is no publicly accessible model for TED on nuScenes.

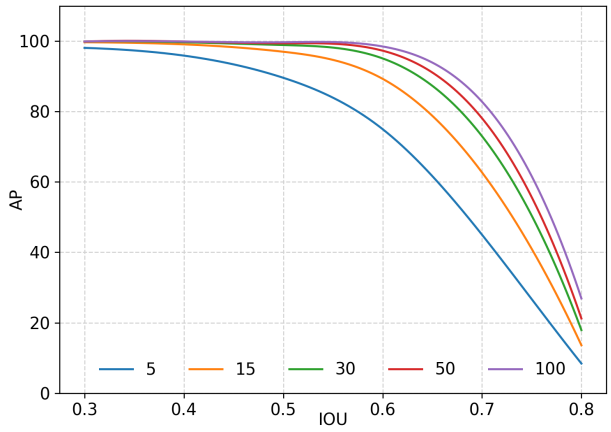

Figure 4. Average precision for different minimum number of points per object in the car class at various IOU thresholds.

with the same models used in CCD on the nuScenes valid dataset. It can be seen how the predicted bounding box tends to align more closely with the ground truth box as the various errors get closer to 0 when more points are present in the object. CCD demonstrates high AP due to two main factors: first, annotator clicks tend to be close to the center of objects, significantly enhancing localization accuracy. Second, false positives are minimized since only the most confident prediction is considered.

Simulated Click Variations. As explained in subsection 3.1, we simulate human clicks by drawing from a normal distribution around a ground truth object center. In addition to the angle variations already mentioned, other distributions and shapes for filtering are considered. Thus, we also experiment with a uniform distribution, a rectangle to filter outliers, and a clipping mechanism for the pivot point selection. We hypothesized that annotators are more inclined to click towards the part of the object closer to the ego than farther from it. Therefore, as another variation, we introduce a clipping parameter that prevents points from being generated beyond one standard deviation on the farther side of the object from the ego vehicle. To find the variation best matching human behavior, we collected 2000 clicks of cars and pedestrians from 10 different skilled annotators while annotating proprietary in-house data. Only objects with more than 30 points were considered following the previously shown findings.

Figure 5 shows the distributions of the actual human clicks for cars and the mathematical models used for simulation. Looking at the results of the human clicks, a uniform distribution and a rectangle shape for filtering can be quickly discarded. There is low correlation between human clicks and generated points for objects for a fixed α angle, and clipping does not improve point distribution alignment with human clicks. The model generating random points (using a Normal distribution) within an ellipse (major diameter: 2 meters, minor diameter: 1 meter) rotated along the yaw axis closely matches the distribution of the one from humans. Acknowledging the inherent approximation involved, we also present all the results of the experiments with different simulations in Table 3. The consistency in performance across different click simulation models, each with varying levels of randomness, highlights the robustness of the model in handling the diverse offsets encountered when different humans click on their perceived object centers. Conversely, the performance improvement when an annotator hypothetically clicks precisely at the object center for every object is also minimal.

Limitations. Although our method performs far better than other one-click methods, there are some limitations. First, experiments with cross-dataset validation are seen to

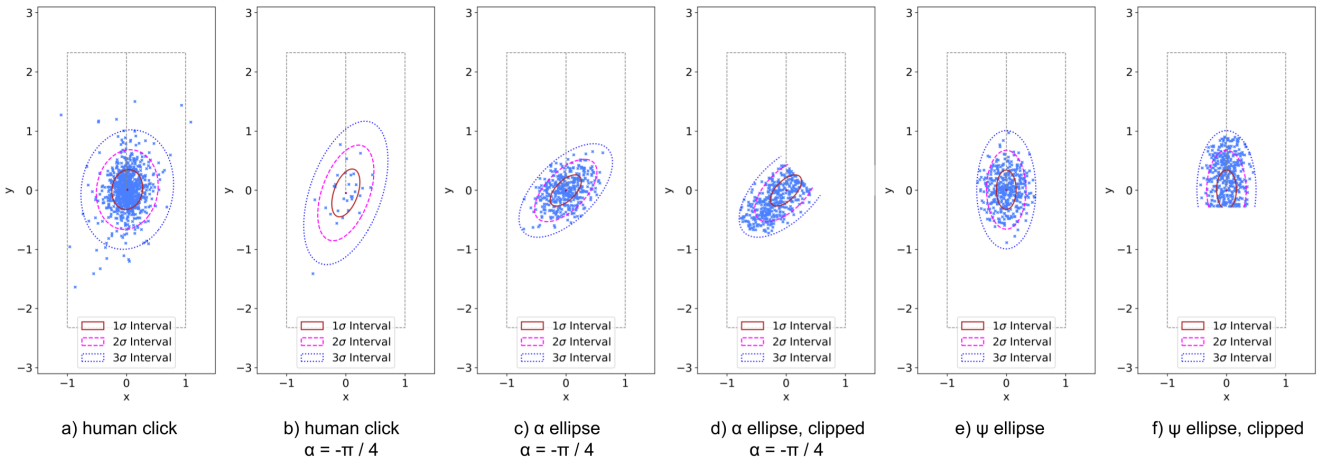

Figure 5. Comparison of collected human clicks (a, b) to various models of randomly generated points (c-f) for cars. The dotted rectangle depicts the average car dimensions in the dataset. Only objects with more than 30 points are considered for the human clicks. The human click distribution a) matches the best with e) a normal distribution rotated by Ψ and limited with the corresponding 3σ ellipse.

Train \ Valid	α Ellipse, $\mathcal{N}$		α Ellipse, Clipped $\mathcal{N}$		Ψ Ellipse, $\mathcal{N}$		Ψ Ellipse, Clipped $\mathcal{N}$		GT center	
	Recall↑	AP↑	Recall↑	AP↑	Recall↑	AP↑	Recall↑	AP↑	Recall↑	AP↑
α Ellipse, $\mathcal{N}$	97.0	96.4	96.5	95.7	97.0	96.4	96.7	96.0	97.7	97.0
α Ellipse, Clipped $\mathcal{N}$	97.3	96.7	97.2	96.6	97.4	96.8	97.3	96.6	98.1	97.5
Ψ Ellipse, $\mathcal{N}$	**97.8**	**97.3**	**97.4**	96.8	**97.8**	**97.4**	97.6	97.0	**98.2**	**97.7**
Ψ Ellipse, Clipped $\mathcal{N}$	97.3	96.7	97.1	96.4	97.4	96.9	97.4	96.8	98.1	97.6
Ψ Ellipse, $\mathcal{U}$	97.6	97.1	**97.4**	**96.9**	97.7	97.3	**97.7**	**97.2**	98.1	**97.7**
Ψ Box, $\mathcal{U}$	97.3	96.8	97.1	96.5	97.4	96.8	97.2	96.5	97.5	96.8
Ψ Box, $\mathcal{N}$	97.5	97.0	97.1	96.4	97.5	97.0	97.3	96.7	97.6	96.9
Ground Truth center	89.3	86.5	86.2	83.1	91.0	88.6	90.0	87.5	97.7	97.2

Table 3. Comparative performance matrix across different shapes and distributions used to simulate the click on nuScenes. Note that the IOU@0.5 metric is used to evaluate objects with at least 15 points. The AP and Recall values are expressed as percentages.

be poor. Even though we significantly simplify the inputs and detection range, the model does not seem to generalize well, e.g., when training completely on KITTI and validating the performance on the nuScenes dataset, or vice versa. Further efforts are needed to increase the domain adaptation capabilities.

While still more accurate than other one-click approaches, our method performs relatively poorly when fewer points exist. This can be attributed to the learned distribution of objects in CCD but missing information at the input level. Higher IOUs can be achieved when more points are available. Multiple sweeps can be used to mitigate this problem, and the time dimension can be exploited to generate objects with denser point clouds. However, this would also have to tackle the issue of dynamic objects, leaving a trail of points when multiple sweeps are stacked in time.

5. Conclusion

In this paper, we proposed CCD, a one-click annotation method with human-in-the-loop, to balance automation and the quality of the labeling results. With a single click towards an object center, a region of interest is cropped, and the object is detected leveraging state-of-the-art deep learning approaches. In extensive experiments, we outperformed all existing methods quantitatively and qualitatively and showed strengths, such as superior generalizability over hand-crafted methods and easy expandability with new detectors. Additionally, for training and validation, we introduced a mathematical model to simulate human clicks backed up by a study of human click behavior, which is valuable for future research. We see great potential for future work, e.g., by adopting the common practice of multi-frame detection to enhance the density of the point clouds.

References

[1] Mina Alibeigi, William Ljungbergh, Adam Tonderski, Georg Hess, Adam Lilja, Carl Lindström, Daria Motorniuk, Junsheng Fu, Jenny Widahl, and Petersson Zenseact. Zenseact open dataset: A large-scale and diverse multimodal dataset for autonomous driving. In *IEEE International Conference on Computer Vision (ICCV)*, pages 20178–20188, 2023. 1

[2] Hasan Asy Ari Arief, Mansur Arief, Guilin Zhang, Zuxin Liu, Manoj Bhat, Ulf Geir Indahl, Havard Tveite, and Ding Zhao. Sane: Smart annotation and evaluation tools for point cloud data. *IEEE Access*, 8:131848–131858, 2020. 2, 4, 5, 6

[3] Prarthana Bhattacharyya and Krzysztof Czarnecki. Deformable pv-rcnn: Improving 3d object detection with learned deformations. In *European Conference on Computer Vision Workshops (ECCV)*, pages 2–6, 2020. 2

[4] 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 *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020. 2, 4, 5

[5] 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 *European Conference on Computer Vision (ECCV)*, pages 680–697, 2022. 2

[6] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21674–21683, 2023. 2

[7] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Rangedet: In defense of range view for lidar-based 3d object detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2918–2927, 2021. 2

[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 *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8458–8469, 2022. 2

[9] Lue Fan, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Fully sparse 3d object detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[10] Lue Fan, Yuxue Yang, Yiming Mao, Feng Wang, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Once detected, never lost: Surpassing human performance in offline lidar based 3d object detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 19820–19829, 2023. 2

[11] Di Feng, Xiao Wei, Lars Rosenbaum, Atsuto Maki, and Klaus Dietmayer. Deep active learning for efficient training of a lidar 3d object detector. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 667–674, 2019. 2

[12] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 5

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. 4

[14] Tianrui Guan, Jun Wang, Shiyi Lan, Rohan Chandra, Zuxuan Wu, Larry Davis, and Dinesh Manocha. M3detr: Multi-representation, multi-scale, mutual-relation 3d object detection with transformers. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 772–782, 2022. 2

[15] Chenhang He, Hui Zeng, Jianqiang Huang, Xian sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11873–11882, 2020. 2

[16] 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. In *Conference on Artificial Intelligence (AAAI)*, pages 969–979, 2022. 2

[17] Hauke Kaulbersch. *Automotive Target Models for Point Cloud Sensors*. PhD thesis, Dissertation, Göttingen, Georg-August Universität, 2021, 2022. 3

[18] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12689–12697, 2019. 2, 3, 4, 5, 7

[19] Jungwook Lee, Sean Walsh, Ali Harakeh, and Steven L. Waslander. Leveraging pre-trained 3d object detection models for fast ground truth generation. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 2504–2510, 2018. 2, 4, 5

[20] E. Li, Shuaijun Wang, Chengyang Li, Dachuan Li, Xiangbin Wu, and Qi Hao. Sustech points: A portable 3d point cloud interactive annotation platform system. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1108–1115, 2020. 1, 2, 6

[21] E. Li, Shuaijun Wang, Chengyang Li, Dachuan Li, Xiangbin Wu, and Qi Hao. Sustechpoints: Point cloud 3d bounding box annotation tool for autonomous driving. https://github.com/naurril/SUSTechPOINTS, 2020. Accessed: 2024-03-21. 6

[22] Zi Li. smart-annotation-pointrcnn. https://github.com/ziliHarvey/smart-annotation-pointrcnn, 2020. Accessed: 2024-03-21. 6

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, pages 21–37, 2016. 3

[24] Tao Ma, Xuemeng Yang, Hongbin Zhou, Xin Li, Botian Shi, Junjie Liu, Yuchen Yang, Zhizheng Liu, Liang He, Yu Qiao, Yikang Li, and Hongsheng Li. Detzero: Rethinking offboard 3d object detection with long-term sequential point clouds. In *IEEE International Conference on Computer Vision (ICCV)*, pages 6736–6747, 2023. 2

[25] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3164–3173, 2021. 2

[26] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 3d object detection for autonomous driving: A comprehensive survey. *International Journal of Computer Vision*, 131:1909–1963, 2023. 1

[27] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015. 2

[28] 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 *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12677–12686, 2019. 2

[29] Trung Nguyen, Binh-Son Hua Duc, Thanh Nguyen, and Dinh Phung. Single-click 3d object annotation on lidar point clouds. In *Advances in Neural Information Processing Systems Workshops (NeurIPS)*, 2021. 2

[30] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3d object detection with pointformer. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7463–7472, 2021. 2

[31] Ziqi Pang, Zhichao Li, and Naiyan Wang. Model-free vehicle tracking and state estimation in point cloud sequences. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 8075–8082, 2021. 2

[32] Patrick Pfreundschuh, Hubertus Franciscus Cornelis Hendrikx, Victor Reijgwart, Renaud Dubé, Roland Siegwart, and Andrei Cramariuc. Dynamic object aware lidar slam based on automatic generation of training data. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 11641–11647, 2021. 2

[33] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017. 2

[34] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5099–5108, 2017. 2

[35] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *IEEE International Conference on Computer Vision (ICCV)*, pages 9277–9286, 2019. 2

[36] 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 *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6134–6144, 2021. 2

[37] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[38] Tim Schreier, Katrin Renz, Andreas Geiger, and Kashyap Chitta. On offline evaluation of 3d object detection for autonomous driving. In *IEEE International Conference on Computer Vision Workshops (ICCV)*, pages 4084–4089, 2023. 4

[39] Hualian Sheng, Sijia Cai, Yuan Liu, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, and Min-Jian Zhao. Improving 3d object detection with channel-wise transformer. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2743–2752, 2021. 2

[40] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–779, 2019. 2

[41] 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 *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10529–10538, 2020. 2

[42] 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. 2

[43] Weijing Shi and Ragunathan Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1711–1719, 2020. 2

[44] Martin Simon, Stefan Milz, Karl Amende, and Horst-Michael Gross. Complex-yolo: Real-time 3d object detection on point clouds. In *European Conference on Computer Vision Workshops (ECCV)*, pages 1–14, 2018. 2

[45] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurélien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020. 1

[46] Pei Sun, Weiyue Wang, Yuning Chai, Gamaleldin Elsayed, Alex Bewley, Xiao Zhang, Cristian Sminchisescu, Dragomir Anguelov, and Waymo Llc. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5725–5734, 2021. 2

[47] Pei Sun, Mingxing Tan, Weiyue Wang, Chenxi Liu, Fei Xia, Zhaoqi Leng, and Dragomir Anguelov. Swformer: Sparse window transformer for 3d object detection in point clouds. In *European Conference on Computer Vision (ECCV)*, pages 426–442, 2022. 2

[48] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. Accessed: 2024-03-21. 4

[49] Bernie Wang. Latte: Accelerating lidar point cloud annotation via sensor fusion, one-click annotation, and tracking. https://github.com/bernwang/latte, 2019. Accessed: 2024-03-21. 6

[50] Bernie Wang, Virginia Wu, Bichen Wu, and Kurt Keutzer. Latte: Accelerating lidar point cloud annotation via sensor fusion, one-click annotation, and tracking. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 265–272, 2019. 1, 2, 4, 5, 6

[51] Tai Wang, Conghui He, Zhe Wang, Jianping Shi, and Dahua Lin. Flava: Find, localize, adjust and verify to annotate lidar-based point clouds. In *ACM Symposium on User Interface Software and Technology*, 2020. 1

[52] Yue Wang, Alireza Fathi, Abhijit Kundu, David Ross, Caroline Pantofaru, Tom Funkhouser, and Justin Solomon. Pillar-based object detection for autonomous driving. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[53] Hai Wu. Transformation-equivariant 3d object detection for autonomous driving. https://github.com/hailanyi/TED, 2022. Accessed: 2024-03-21. 4

[54] Hai Wu, Chenglu Wen, Wei Li, Xin Li, Ruigang Yang, and Cheng Wang. Transformation-equivariant 3d object detection for autonomous driving. In *Conference on Artificial Intelligence (AAAI)*, pages 2795–2802, 2023. 2, 3, 4, 5

[55] 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 *European Conference on Computer Vision (ECCV)*, pages 193–209, 2022. 2

[56] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18:3337, 2018. 2

[57] Anqi Joyce Yang, Sergio Casas, Nikita Dvornik, Sean Segal, Yuwen Xiong, Jordan Sir Kwang Hu, Carter Fang, and Raquel Urtasun. Labelformer: Object trajectory refinement for offboard perception from lidar point clouds. In *Conference on Robot Learning (CoRL)*, pages 3364–3383, 2023. 1, 2

[58] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7652–7660, 2018. 2

[59] 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

[60] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1951–1960, 2019. 2

[61] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11040–11048, 2020. 2

[62] Zetong Yang, Yin Zhou, Zhifeng Chen, and Jiquan Ngiam. 3d-man: 3d multi-frame attention network for object detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1863–1872, 2021. 2

[63] Maosheng Ye, Shuangjie Xu, and Tongyi Cao. Hvnet: Hybrid voxel network for lidar based 3d object detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1631–1640, 2020. 2

[64] Mao Ye, Gregory P Meyer, Yuning Chai, and Qiang Liu. Efficient transformer-based 3d object detection with dynamic token halting. In *IEEE International Conference on Computer Vision (ICCV)*, pages 8438–8450, 2023. 2

[65] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11784–11793, 2021. 2, 3, 5, 7

[66] Yurong You, Katie Luo, Cheng Perng Phoo, Wei-Lun Chao, Wen Sun, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Learning to detect mobile objects from lidar scans without labels. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1130–1140, 2022. 2

[67] Sergey Zakharov, Wadim Kehl, Arjun Bhargava, and Adrien Gaidon. Autolabeling 3d objects with differentiable rendering of sdf shape priors. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12224–12233, 2020.

[68] Dingyuan Zhang, Dingkang Liang, Zhikang Zou, Jingyu Li, Xiaoqing Ye, Zhe Liu, Xiao Tan, and Xiang Bai. A simple vision transformer for weakly semi-supervised 3d object detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 8373–8383, 2023.

[69] Lunjun Zhang, Anqi Joyce Yang, Yuwen Xiong, Sergio Casas, Bin Yang, Mengye Ren, and Raquel Urtasun. Towards unsupervised object detection from lidar point clouds. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9317–9328, 2023. 2

[70] Wu Zheng, Weiliang Tang, Sijin Chen, Li Jiang, and Chi-Wing Fu. Cia-ssd: Confident iou-aware single-stage object detector from point cloud. In *Conference on Artificial Intelligence (AAAI)*, pages 3555–3562, 2021. 2

[71] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Se-ssd: Self-ensembling single-stage object detector from point cloud. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14494–14503, 2021.

[72] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4490–4499, 2018. 2

[73] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *Conference on Robot Learning (CoRL)*, pages 923–932, 2020. 2

[74] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. Centerformer: Center-based transformer for 3d object detection. In *European Conference on Computer Vision (ECCV)*, pages 496–513, 2022. 2

[75] Walter Zimmer, Akshay Rangesh, and Mohan Trivedi. 3d bat: A semi-automatic, web-based 3d annotation toolbox for

full-surround, multi-modal data streams. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1816–1821, 2019. 2