iDet3D: Towards Efficient Interactive Object Detection for LiDAR Point Clouds

Dongmin Choi[1,2*], **Wonwoo Cho**[1,2*], **Kangyeol Kim**[1,2], **Jaegul Choo**[1,2]
[1]Letsur Inc.
[2]Korea Advanced Institute of Science and Technology
{dmchoi, wcho, kangyeolk, jchoo}@kaist.ac.kr

Abstract

Accurately annotating multiple 3D objects in LiDAR scenes is laborious and challenging. While a few previous studies have attempted to leverage semi-automatic methods for cost-effective bounding box annotation, such methods have limitations in efficiently handling numerous multi-class objects. To effectively accelerate 3D annotation pipelines, we propose **iDet3D**, an efficient interactive 3D object detector. Supporting a user-friendly 2D interface, which can ease the cognitive burden of exploring 3D space to provide click interactions, iDet3D enables users to annotate the entire objects in each scene with minimal interactions. Taking the sparse nature of 3D point clouds into account, we design a negative click simulation (NCS) to improve accuracy by reducing false-positive predictions. In addition, iDet3D incorporates two click propagation techniques to take full advantage of user interactions: (1) dense click guidance (DCG) for keeping user-provided information throughout the network and (2) spatial click propagation (SCP) for detecting other instances of the same class based on the user-specified objects. Through our extensive experiments, we present that our method can construct precise annotations in a few clicks, which shows the practicality as an efficient annotation tool for 3D object detection.

Introduction

3D object detection is a long-standing research topic that has been actively studied in industrial fields such as autonomous driving (Geiger, Lenz, and Urtasun 2012; Mao et al. 2022) and robotics (Geiger et al. 2013; Wang and Posner 2015). Although LiDAR point clouds have been widely used to efficiently represent complex scenes in 3D applications, their sparse and orderless nature leads their annotation process to be costly and erroneous (Luo et al. 2023; Hu et al. 2022; Wu et al. 2021), which can be a bottleneck for developing robust 3D object detectors. For instance, even the renowned benchmark dataset, KITTI (Geiger, Lenz, and Urtasun 2012), contains several mislabeled objects (Li et al. 2020a).

To alleviate the complexity of LiDAR object annotation, this paper incorporates *user interactions* (*i.e.,* clicks) into 3D object detectors as in the literature of interactive segmentation (Jang and Kim 2019; Lin et al. 2020; Liu et al. 2022). However, due to the unique aspects of point cloud data, interactive annotation methods should consider the following requirements. (1) Distinguishing the points of foreground objects from the others is challenging in sparse point clouds (Guo et al. 2020); *mispredictions should be effectively handled.* (2) A point cloud scene often contains multiple 3D instances of different categories; *each user interaction should not be limited to a single object*. Thus, without properly taking account of the aspects, simply extending existing interactive approaches to 3D object detection may produce sub-optimal and insufficient results.

By addressing the aforementioned requirements, we propose an interactive 3D object detector called **iDet3D**, which includes three components: negative click simulation (NCS), dense click guidance (DCG), and correlation-based spatial click propagation (SCP). Based on our user-friendly annotation and click-encoding algorithm for point clouds, which allow users to easily click on a 2D visual interface without laboriously exploring 3D space, such components boost the efficiency and effectiveness of our proposed iDet3D.

Since iDet3D should be able to effectively remove false-positive predictions by applying a few negative clicks, it is important to simulate negative clicks at training time properly. Our NCS strategy aims to assign negative click candidates on background points that are prone to be mistakenly predicted as foreground instances. In addition, the two click propagation strategies, DCG and SCP, helps iDet3D to maintain and propagate user guidance throughout deep network layers and spatial point embeddings, respectively. To the best of our knowledge, iDet3D, which is supported by the proposed components, is the first interactive 3D framework capable of detecting numerous 3D multi-class objects.

In Figure 1, we demonstrate an interactive annotation example of our proposed iDet3D, giving a positive click on the pedestrian followed by a negative click to suppress the false positives. Our main contributions are as follows:

- We propose iDet3D, a novel interactive 3D object detector that detects multiple objects of different categories in LiDAR point clouds within a few user clicks.
- The NCS strategy makes iDet3D be capable of reducing false-positives by leveraging user-given negative clicks.
- We carefully design effective click propagation methods (DCG and SCP) to take full advantage of user-provided interactions throughout the network and a 3D scene.

[*]These authors contributed equally.

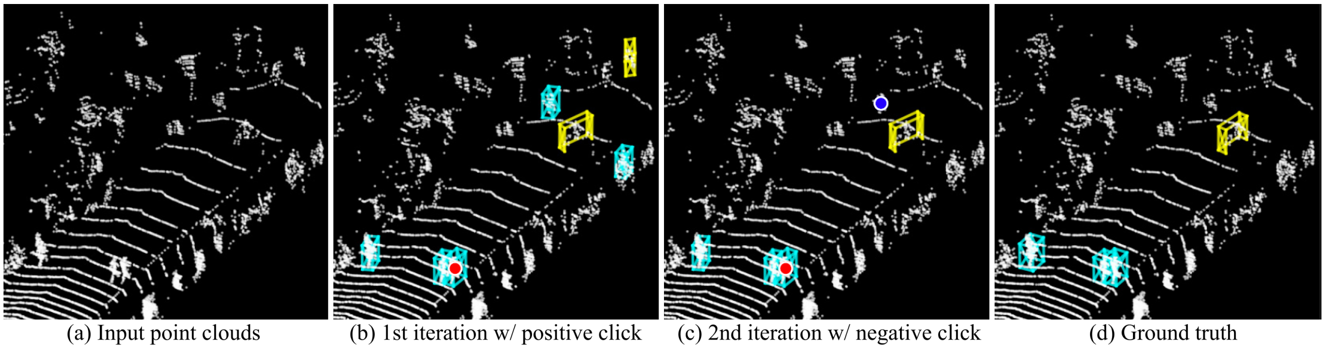

(a) Input point clouds (b) 1st iteration w/ positive click (c) 2nd iteration w/ negative click (d) Ground truth

Figure 1: An example of the iterative annotation process of iDet3D. (a) Given input point clouds. (b) Provided a positive click on a pedestrian (red circle), the proposed iDet3D detects multiple objects of various classes in the scene within a single click. (c) In the second iteration, the false-positive predictions can be filtered out at once by adding a single negative click (blue circle). (d) Ground truth. Within a few iterations, one can obtain high-quality annotation results.

Related Work

3D Object Detectors

Voxel-based. To use the mature convolutional neural networks (CNNs) in processing the unordered 3D point clouds, previous studies (He et al. 2020; Liu et al. 2020; Shi and Rajkumar 2020; Zheng et al. 2021) have proposed to transform sparse point clouds into dense voxel grids through spatial quantization (*i.e.,* voxelization). Although this enables CNN-based analysis of 3D scenes (Wang and Posner 2015) via 3D sparse convolution (Yan, Mao, and Li 2018) or pillars (Lang et al. 2019), voxel-based methods inevitably suffer from information loss caused by their inability to fully exploit the original structural information (Xu et al. 2021; Akhtar et al. 2020). In addition, their computational costs increase cubically with input resolution, while requiring additional time for input pre-processing and post-processing.

In interactive applications, it is intuitive that the response time (or latency) of an algorithm is particularly important for a better user experience. Since the aforementioned problems of voxel-based detectors may hinder flexible real-time interactions, we consider point-based backbones in this paper.

Point-based. Unlike voxel-based ones, point-based detectors directly take 3D point coordinates as input and then analyze per-point embeddings. After PointNet (Qi et al. 2017), the first backbone architecture that directly analyzes point clouds, was introduced, various point-based detectors (Shi and Rajkumar 2020; Li et al. 2020b; Yang et al. 2020; Zhang et al. 2022) have been developed. Compared to voxel-based models, point-based methods are generally more efficient in terms of memory (Guo et al. 2020).

Point-based detectors can be divided into *two-stage* and *single-stage* detectors. Although early studies have focused on two-stage approaches (Shi, Wang, and Li 2019; Qi et al. 2019) due to the limited detection accuracy of single-stage detectors, two-stage detectors can also suffer from low inference speed problems caused by their time-consuming layers. Thus, recent studies have focused on designing efficient but effective single-stage 3D detectors. 3DSSD (Yang et al. 2020) was proposed as the first single-stage point-based 3D object detector, which removes computational-heavy components for 3D proposals and overcomes the following performance drop through semantic feature-based farthest point sampling (FPS). Afterwards, a highly efficient 3D detector called IA-SSD (Zhang et al. 2022) was introduced for deployment in practice, which replaces time-consuming FPS with instance-aware downsampling MLP layers.

Since IA-SSD achieved competitive detection accuracy while maintaining high efficiency, we employ this architecture as the backbone of iDet3D. To present that the principle of iDet3D is applicable to other single-stage detectors, we also exploit 3DSSD backbone in our experiment.

Interactive Point Cloud Annotation

A few approaches have been proposed to incorporate interactive techniques into 3D point clouds. For point cloud segmentation tasks, scribble and click-based interactive refinement approaches have been investigated (Shen et al. 2020; Kontogianni et al. 2022). In 3D object detection, an interactive annotation framework based on LiDAR sensor fusion and one-click bounding box drawing was introduced (Wang et al. 2019). Afterwards, a more advanced annotation system that supports smart 3D bounding box initialization and automatic box fitting was developed (Li et al. 2020a).

While the previous methods can accelerate point cloud annotation compared to manual labeling processes, such interactive schemes are still limited to identifying an individual instance at once, *i.e.,* users can modify the annotation of only a single object for each interaction.

Method

Overview. Throughout this paper, we describe our proposed iDet3D based on the IA-SSD backbone (Zhang et al. 2022), a recently proposed 3D object detector. It is noteworthy that our principle can be easily applicable to other single-stage point-based detectors. iDet3D supports two types of user interactions: *class-specific* positive and *class-agnostic* negative clicks, which are designed to indicate the locations of foreground objects and background regions, respectively. Figure 2 shows the overall architecture of iDet3D.

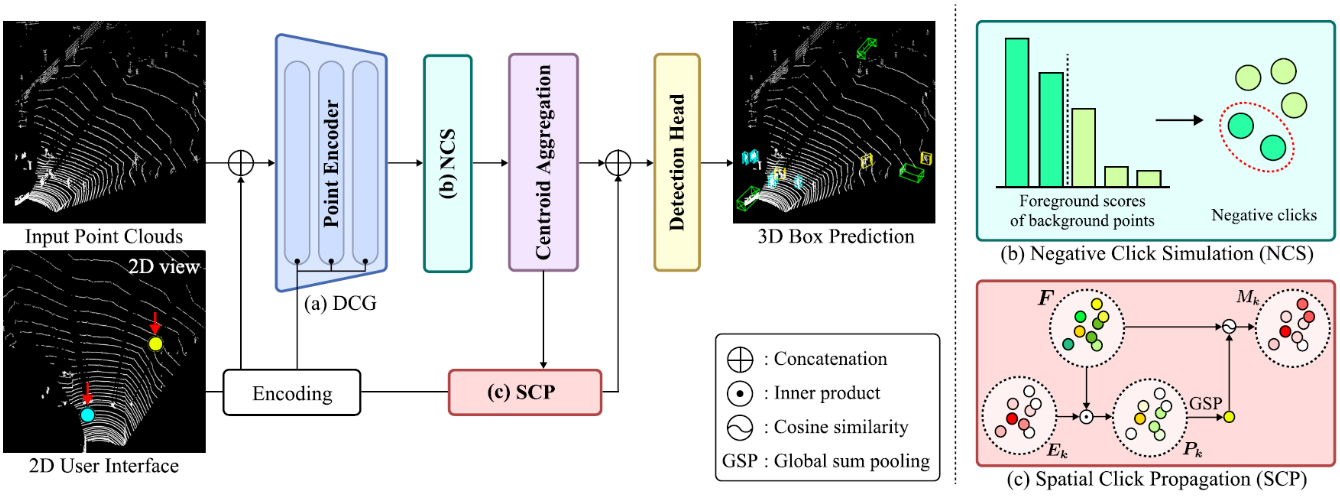


Figure 2: The training workflow of iDet3D. Given user clicks on target objects, the clicks are transformed into click encodings. (a) Dense click guidance (DCG) fuses the encodings into the backbone network architecture not only at the input side but also at the intermediate layers. (b) Negative click simulation (NCS) randomly simulates probable negative points by selecting challenging background points with high foreground scores. (c) The following spatial click propagation (SCP) module effectively propagates user clicks to detect other objects of the same class based on the similarity between feature embeddings.

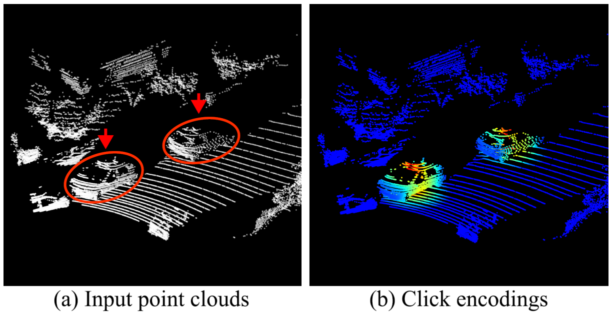

(a) Input point clouds (b) Click encodings

Figure 3: An example visual illustration of click encoding in our iDet3D. (a) Input point clouds and user clicks (red arrows). (b) The corresponding distance-encoded user interactions highlighted on the target objects.

Click Encoding

A straightforward approach to provide interaction to a given 3D scene is to directly click on the objects of interest (Kontogianni et al. 2022). However, the process of specifying the 3D coordinates of a small point in a vast 3D space imposes a significant cognitive burden on users.

Instead, we develop a user-friendly interface in 2D view, where users can provide simple 2D clicks on target objects. For a better understanding, we visualize the difference between 3D and our 2D interfaces. In the 3D interface, a slight shift of a cursor may lead to undesired movements of coordinates in another axis. However, our 2D annotation environment can mitigate such errors by eliminating the requirement of specifying z-axis locations.

Suppose that K class-specific positive clicks are provided on a scene to annotate foreground objects of total C categories. Then, the k-th click can be written as (p_k, c_k), where $p_k = (p_{k,x}, p_{k,y})$ denotes the 2D coordinate of the click, and $c_k \in \{1, \cdots, C\}$ is the corresponding class. Following the convention of deep interactive annotation methods (Xu et al. 2016), we transform user clicks into the corresponding distance heatmap to generate a proper input for point-based detectors. Given a 3D point cloud scene composed of N points $\{(x_i, y_i, z_i)\}_{i=1}^{N}$, we encode (p_k, c_k) into a *click-wise* encoding $E_k \in \mathbb{R}^N$, whose i-th element is

$$E_k[i] = \exp\left(\max\left\{\frac{\tau - d}{\tau}, 0\right\} \cdot \log 2\right) - 1. \quad (1)$$

In Eq. (1), $d = \sqrt{(p_{k,x} - x_i)^2 + (p_{k,y} - y_i)^2}$ represents the 2D Euclidian distance between p_k and (x_i, y_i), and τ is a hyperparameter to control the distance threshold. Note that E_k is designed to highlight p_k and its neighboring points within the $[0, 1]$ scale.

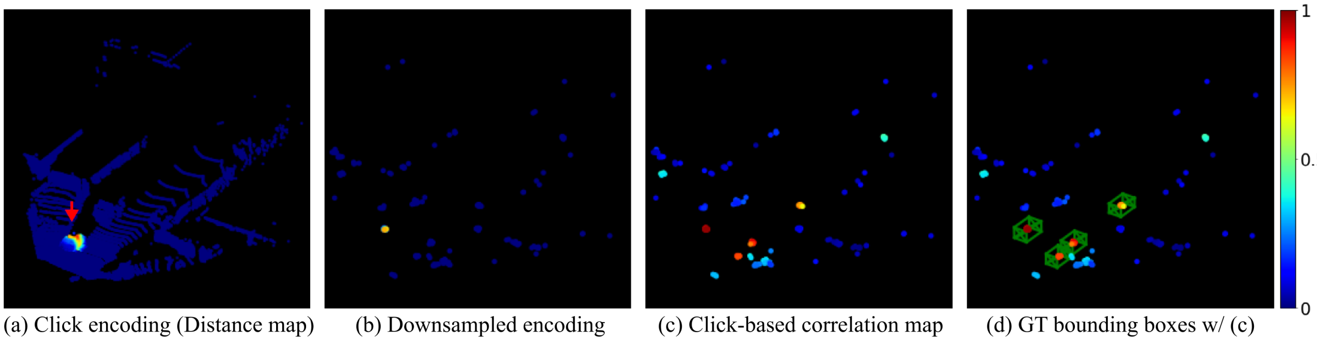
To effectively feed the encoded clicks $\{E_k\}_{k=1}^{K}$ to networks, we define a *class-wise* click encoding $U_c \in \mathbb{R}^N$ for class $c \in \{1, \cdots, C\}$ via element-wise max pooling, *i.e.,*

$$U_c[i] = \max\{E_k[i] | c_k = c, \forall k \in \{1, \cdots, K\}\}. \quad (2)$$

Once C encodings are generated, we concatenate $\{U_c\}_{c=1}^{C}$ to the corresponding input points. For better understanding, we visualize an example of U_c computed by two clicks of Car class in Figure 3. We define a *vanilla model* by a combination of this click encoding and the backbone encoder.

Negative Click Simulation

We observe that the vanilla model with only positive clicks fails to separate background point clouds from foreground ones, causing unexpected false-positive predictions. To mitigate a similar problem, previous studies on interactive segmentation (Xu et al. 2016; Sofiiuk, Petrov, and Konushin 2022) have made use of *negative clicks* to indicate the undesired region. In general, they randomly sample negative



(a) Click encoding (Distance map) (b) Downsampled encoding (c) Click-based correlation map (d) GT bounding boxes w/ (c)

Figure 4: An example of correlation map generation using spatial click propagation (SCP) module (visualized in the scale of [0,1]). (a) Click encoding E on a car object (red arrow) with respect to N input points. (b) Click encoding E' for N' downsampled points (after point reduction from N to N' with downsampling layers). (c) Click-wise correlation map M generated by the SCP and (d) visualization overlaid with the ground-truth 3D bounding boxes. Note that points of the same class with the click are highlighted in the output correlation map.

clicks based on the assumption that real users are likely to provide negative clicks to areas outside foreground regions but near object boundaries. However, because *false positives in 3D object detection can happen regardless of foreground object locations*, the simulation strategy of interactive segmentation may not derive reasonable negative clicks.

Instead, we propose negative click simulation (NCS) suitable for 3D object detection with the goal of sampling challenging background points that are likely to be inaccurately predicted as foreground. For this purpose, we take advantage of MLP-based scoring embedded in the down-sampling approaches of recent point-based detectors (Zhang et al. 2022; Chen et al. 2022). This method assigns high scores to potential foreground points and selects top-n points to be downsampled, which implies that several challenging background points can be ranked in top-n.

We expand the functionality of this layer to act as a negative click simulator by selecting *background points with high foreground scores* as negative clicks. After this simulation strategy, we sample top-K_n background points and encode them with the same manner of positive clicks, *i.e.*, click encoding becomes $(C+1)$ channels, where the additional single channel is for class-agnostic negative clicks.

User Click Propagation

In addition to the limitation of false-positive predictions, we discover that the vanilla model sometimes fails to detect the user-specified object. This finding implies two drawbacks of the model: (1) *user intention can be diluted through the forward pass of backbone layers* and (2) *user clicks are limited in affecting multiple objects*. To address these problems, we propose two click propagation methods, which are DCG to make iDet3D sustain user intention and SCP to empower a user click to influence other objects of the same category.

Dense click guidance (DCG). If the user click encoding $\{U_c\}_{c=1}^{C+1}$ are only fused to the input points, user intention in the click guidance can be diluted as the network layer goes deeper (Zhang et al. 2019; Ding et al. 2022; Hao et al. 2021). Furthermore, the point-based detectors with downsampling layers for computational efficiency may cause potential losses of foreground points or critical information for 3D scene understanding (Hu et al. 2021; Zhang et al. 2022). Thus, the following prediction head may be unable to effectively leverage user-provided hints.

To address this problem, we concatenate encoded clicks to input point clouds as well as the intermediate point embeddings of the encoder after each downsampling layer, as illustrated in Figure 2(b). This *dense guidance* strategy greatly helps iDet3D to utilize user guidance throughout the network without forgetting the user intention.

Spatial click propagation (SCP). Most 3D scenes contain not a single but multiple objects. For better efficiency, it is required to utilize user guidance to detect all instances including unspecified objects. However, the click encoding obtained by following Eq. (1) only highlights the single object specified by users. Therefore, it is limited to affecting other instances that have not been explicitly indicated.

Inspired by object counting algorithms (Arteta et al. 2014; Ranjan et al. 2021) and a multi-class 2D interactive detector (Lee et al. 2022), we add the SCP module to the output of the encoder to enhance the click efficiency (Figure 2). Let the output point embeddings of the encoder be $F \in \mathbb{R}^{N' \times D}$, where N' is the number of downsampled points and D indicates the dimension of the embedding. To be aligned with the downsampled points, we define $E'_k \in \mathbb{R}^{N'}$, a downsampled E_k for the k-th click, by recomputing Eq. (1) with respect to the encoder output features F. Then, we compute a click prototype vector $P_k \in \mathbb{R}^D$ by

$$P_k = \sum_{j}^{N'} \left(F[j,:] \cdot \frac{E'_k[j]}{\|E'_k\|_1} \right), \tag{3}$$

where $F[j,:] \in \mathbb{R}^D$ is the j-th point embedding and $\|E'_k\|_1$ is the L_1-norm of E'_k. In other words, each P_k is a weighted sum of point embeddings corresponding to the neighboring points of the click, which encodes the prototype representation of the object indicated by the k-th click.

	Method	N_{clks}	3D Car (IoU=0.7)			3D Ped. (IoU=0.5)			3D Cyc. (IoU=0.5)		
			Easy	*Mod.*	*Hard*	*Easy*	*Mod.*	*Hard*	*Easy*	*Mod.*	*Hard*
Voxel	VoxelNet (Zhou and Tuzel 2018)	-	81.97	65.46	62.85	57.86	53.42	48.87	67.17	47.65	45.11
	SECOND (IoU) (Yan, Mao, and Li 2018)	-	84.88	76.30	75.97	37.26	34.60	32.62	80.63	64.35	60.38
	PointPillars (Lang et al. 2019)	-	86.45	77.29	74.65	57.76	52.30	47.91	80.05	62.73	59.67
	Part-A^2 (Anchor) (Shi et al. 2020)	-	89.56	79.41	78.84	65.69	60.05	55.44	85.50	69.93	65.49
Point	3DSSD (Reproduced) (Yang et al. 2020)	-	89.12	83.94	78.47	60.65	56.05	52.19	84.75	69.14	64.58
	IA-SSD (Zhang et al. 2022)	-	89.47	79.57	78.45	62.38	58.91	51.46	86.65	71.24	66.11
	IA-SSD (Reproduced) (Zhang et al. 2022)	-	89.20	79.28	78.15	60.92	57.77	51.66	87.16	68.25	66.77
Interactive	Vanilla iDet3D (3DSSD backbone)	1	88.82	83.87	78.38	62.56	57.36	53.45	84.35	67.42	64.50
		3	88.85	83.93	78.42	62.72	58.29	53.81	84.85	68.65	64.86
		5	88.86	83.94	78.42	62.89	58.60	53.90	84.83	68.18	64.84
	Vanilla iDet3D (IA-SSD backbone)	1	88.70	79.28	78.34	58.71	55.47	50.98	86.47	70.40	65.49
		3	88.76	79.30	78.36	58.78	55.67	51.07	86.49	70.45	65.58
		5	88.76	79.31	78.37	58.76	55.72	51.07	86.51	70.44	65.56
	iDet3D (3DSSD backbone)	1	91.17	88.09	86.79	60.06	57.66	52.21	88.14	73.21	69.29
		3	96.26	88.91	88.51	62.26	58.79	55.08	88.88	75.73	72.66
		5	97.15	88.96	88.61	62.93	59.08	55.28	88.92	76.06	73.09
	iDet3D (IA-SSD backbone)	1	89.83	87.99	85.68	60.87	57.27	51.66	90.60	74.69	73.00
		3	97.21	89.74	89.47	66.13	62.37	57.40	96.46	83.31	77.91
		5	**98.55**	**90.37**	**90.27**	**70.07**	**65.74**	**60.57**	**98.43**	**88.00**	**80.19**

Table 1: Quantitative results of baselines and iDet3D on the KITTI *val* set. The best results (measured in the AP metric) among the non-interactive models are underlined, where the best ones of the interactive methods are highlighted in bold.

For P_k, we compute a *click-wise* correlation map $M_k \in \mathbb{R}^{N'}$ based on the cosine similarity followed by a global sum pooling, which can be represented as

$$M_k[j] = \frac{F[j,:] \odot P_k}{\|F[j,:]\|_2 \|P_k\|_2}, \tag{4}$$

where $F[j,:] \odot P_k$ is the inner product of $F[j,:]$ and P_k. As M_k is designed to highlight those points with high cosine similarity between their feature embeddings and P_k, it is able to spatially propagate a click on a single object to other unspecified ones belonging to the same class. To provide further insight, we illustrate an example in Figure 4.

To aggregate the given click-wise maps before incorporating them into the network, we compute a class-wise correlation map $S_c \in \mathbb{R}^{N'}$ for the class c by

$$S_c[j] = \max\{M_k[j] | c_k = c, \forall k \in \{1, \cdots, K\}\}. \tag{5}$$

Following both DCG and SCP, point embeddings (with class-wise click encodings of $C+1$ dimension and correlation maps of $C+1$ dimension) become $(D+2C+2)$-dimension vectors, and the detection head takes them as input to make final predictions. It is noteworthy that negative clicks in our system can also affect other unspecified background points since the SCP module generates a negative click-based correlation map.

Experiments

Experimental Settings

Evaluation protocols. We evaluate iDet3D with comparison to automatic (non-interactive) baselines including voxel-based (Zhou and Tuzel 2018; Yan, Mao, and Li 2018; Wang et al. 2020; Liu et al. 2020; Shi et al. 2020) and point-based 3D object detectors (Yang et al. 2020; Zhang et al. 2022).

Furthermore, the *vanilla* interactive model (without NCS, DCG, and SCP) serves as a simple baseline. Following the convention of evaluation protocol in interactive segmentation (Li, Chen, and Koltun 2018; Sofiiuk et al. 2020; Sofiiuk, Petrov, and Konushin 2022), we measure the performance by iteratively increasing the number of clicks.

At each interaction, we prioritize reducing false-negative predictions by adding a positive click. If all target objects in a given 3D scene are predicted, we then provide a negative click to suppress false-positive cases if they exist. Quantitative results are reported by averaging the scores of five randomized click sampling trials.

Datasets. KITTI benchmark (Geiger, Lenz, and Urtasun 2012) is a widely used 3D object detection dataset, which consists of 3,712 training and 3,769 validation samples with three object classes: Car, Pedestrian, and Cyclist. Following the official KITTI evaluation protocol, we measure the average precision (AP) metric with an intersection over union (IoU) threshold of 0.7 for Car and 0.5 for Pedestrian and Cyclist. For evaluation, we use KITTI *val* set instead of the *test* set because a user click simulation is not allowed in benchmark submission due to the unavailability of ground truth annotation. We also evaluate iDet3D on a more challenging nuScenes dataset (Caesar et al. 2020), which contains 1,000 scenes recorded in Boston and Singapore comprising 20,000 frames. In total, nuScenes includes approximately 1.4M objects with 10 object categories.

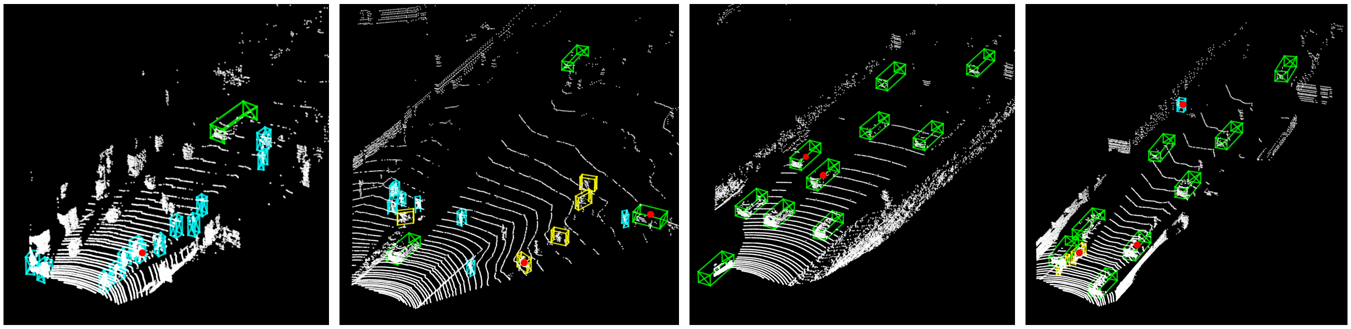


Figure 5: Qualitative results of iDet3D on the KITTI *val* set. Green boxes for Car, cyan for Pedestrian, and yellow for Cyclist with user clicks in red circles. Our model successfully detects multiple objects of different categories with a few user clicks.

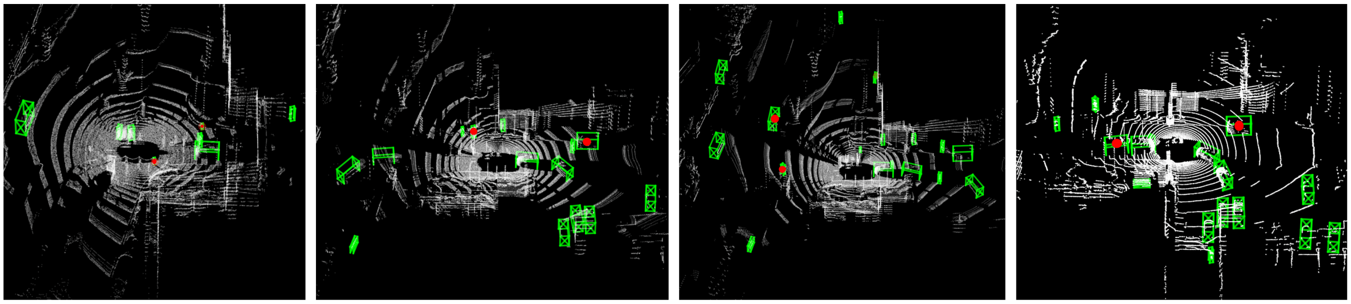


Figure 6: Qualitative results of iDet3D on the nuScenes validation set. The 3D bounding box predictions are colored with green, and the user clicks are represented as red circles.

Implementation details. During training and evaluation, we perform positive click simulation by randomly sampling 2D coordinates inside the 3D ground-truth boxes. The number of clicks K is determined as $\min{(N_u, N_o)}$, where N_u is sampled from $\{0, \cdots, 10\}$ uniformly at random and N_o refers to the number of existing objects in each scene. The distance threshold τ of user encodings is set to 2.0 in Eq. (1). For negative clicks, we set the maximum number K_n to 10.

For more detailed training configurations for each backbone architecture, readers are referred to the codebases of IA-SSD (Zhang et al. 2022) and SASA (Chen et al. 2022). We use 4 NVIDIA RTX A6000 GPUs for experiments.

Experimental Results

In the experiments, our main interest is two-fold: (1) demonstrating the effectiveness of the interactive approach compared to state-of-the-art automatic (non-interactive) detectors, *i.e.,* the performance can be significantly improved by using a few user clicks, and (2) validating the effectiveness of the proposed components of iDet3D (NCS, DCG, and SCP) by comparing iDet3D to the vanilla model.

KITTI. Table 1 shows quantitative comparison results between the baselines on the KITTI validation set. As shown in the table, iDet3D achieves superior or competitive results compared to non-interactive baselines, which implies that a few user clicks can be fulfilling sources for our model. It is noteworthy that even a single click can be effective in handling relatively challenging cases (*Moderate* and *Hard* cases of Car and Cyclist). Also, as the number of clicks increases, the detection accuracy gradually improves, indicating that additional user clicks are successfully incorporated. Several qualitative results are shown in Figure 5.

nuScenes. Next, we perform an evaluation on nuScenes, which has a larger number of object classes than KITTI. In comparison with our backbone IA-SSD (Zhang et al. 2022), Table 2 shows that iDet3D achieves superior detection performance by applying only a single click per frame. Although direct comparison between the detection accuracy of our implemented IA-SSD (and iDet3D) on nuSences and the reported best accuracy of other non-interactive baselines can be misleading, we confirm that iDet3D with five user clicks shows superior performance compared to the baselines in most of the categories and significantly outperforms them with respect to mAP. Especially, user clicks effectively work on challenging classes indicating that a few clicks can significantly aid our proposed model for 3D detection in LiDAR point clouds. We visualize qualitative results in Figure 6.

Additional Results

Ablation study. Our ablation study analyzes the effect of each component of iDet3D based on the KITTI *val*. As reported in Table 3, DCG leads to an overall improvement in detection accuracy. This finding suggests that retaining click information throughout the feature extraction process in the encoder is critical. Also, we also discover that NCS significantly enhances overall performance, which emphasizes the

	Method	N_{clks}	Car	Ped	Bus	Barrier	T.C.	Truck	Trailer	Moto	C.V.	Bicycle	mAP
Voxel-based	PointPillars (Lang et al. 2019)	-	70.5	59.9	34.4	33.2	29.6	25.0	20.0	16.7	4.5	1.6	29.5
	SECOND (Yan, Mao, and Li 2018)	-	75.53	59.86	29.04	32.21	22.49	21.88	12.96	16.89	0.36	0	27.12
Point-based	3DSSD (Yang et al. 2020)	-	81.20	70.17	61.41	47.94	31.06	47.15	30.45	35.96	12.64	8.63	42.66
	SASA (Chen et al. 2022)	-	76.8	69.1	66.2	53.6	29.9	45.0	36.5	39.6	16.1	16.9	45.0
	IA-SSD* (Zhang et al. 2022)	-	72.84	61.51	66.22	37.02	22.08	41.63	29.26	34.95	17.93	11.76	39.51
Interactive	Vanilla iDet3D	1	72.63	60.93	65.78	37.35	21.93	41.51	26.97	34.01	17.96	12.68	39.17
		3	72.64	61.16	65.77	37.61	22.17	41.57	26.96	34.68	17.93	12.93	39.34
		5	72.64	61.39	65.72	37.80	22.34	41.57	26.96	35.22	17.93	13.17	39.48
	iDet3D	1	73.41	62.99	66.97	39.59	22.96	43.31	26.74	39.97	21.41	19.03	41.64
		3	75.57	67.61	68.44	46.02	29.64	46.28	27.58	49.18	27.77	33.62	47.17
		5	**77.28**	**71.14**	**69.70**	**50.68**	**35.71**	**48.67**	**29.28**	**56.19**	**34.51**	**45.16**	**51.83**

Table 2: Quantitative results of the baselines and iDet3D (IA-SSD backbone) with the nuScenes dataset. The best results among the non-interactive models are underlined, where the best ones of the interactive methods are highlighted in bold. IA-SSD* is the reproduced version of (Zhang et al. 2022) by adapting the training configuration in the codebase of (Chen et al. 2022).

	DCG	NCS	SCP	Car (IoU=0.7)		Ped. (IoU=0.5)		Cyc. (IoU=0.5)	
				Easy.	*Mod.*	*Easy.*	*Mod.*	*Easy.*	*Mod.*
(1)	-	-	-	88.76	82.08	58.76	56.60	86.51	74.21
(2)	✓	-	-	90.84	88.86	60.39	58.89	89.00	76.24
(3)	✓	✓	-	92.11	90.30	69.94	65.15	**98.46**	84.60
(4)	✓	✓	✓	**98.55**	**90.37**	**70.07**	**65.74**	98.43	**88.00**

Table 3: Ablation study of iDet3D with five clicks. Adding DCG, NCS, and SCP consistently boosts detection accuracy.

necessity of negative clicks and the effectiveness of our simulation method. Lastly, a combination of all components enhances overall performance. It implies that the SCP module successfully propagates positive and negative clicks to the entire 3D scene, thus enhancing our click efficiency.

Another backbone. To show that the principle of iDet3D can be applicable to another backbone architecture, we also employ 3DSSD (Yang et al. 2020) as our backbone network. For NCS, we adapt centerness values computed in 3DSSD as foreground scores. The corresponding experimental results based on the KITTI *val.* set are presented in Table 1, where the results present that our proposed components can be incorporated into other single-stage point-based backbones.

Comparison with another annotation framework. Furthermore, we quantitatively compare our iDet3D to another semi-automatic annotation framework LATTE (Wang et al. 2019), where the system generates a single bounding box by receiving a click provided on each object (one-to-one). As shown in Table 4, our framework that detects multiple objects of different classes in a few clicks (many-to-many) outperforms LATTE in accuracy and click efficiency.

Furthermore, we conduct a user study with well-educated annotators. The study compares the efficiency between manual annotation, LATTE, and our proposed iDet3D in terms of the average number of clicks and time required to complete annotating each 3D scene as shown in Table 5.

Method	Precision (%)	Recall (%)	N_{clks} / instance
LATTE	78.8	84.8	1.29
iDet3D	82.7	85.9	0.23
	83.8	**88.0**	**0.69**

Table 4: Comparison between iDet3D and LATTE. The last column shows the average number of clicks per instance.

Method	Avg. # of clicks per scene	Avg. time per scene
Manual	45 clicks	$<$ 120 sec.
LATTE	19 clicks	$<$ 50 sec.
iDet3D	2.5 clicks	$>$ 10 sec.

Table 5: User study comparing the efficiency between manual annotation, LATTE, and our proposed iDet3D.

Concluding Remarks

In this paper, we propose iDet3D, the first interactive 3D object detector which is capable of detecting numerous multi-class objects within a few clicks. For effective and efficient 3D detection, we design NCS to filter out false positive predictions via negative clicks, and two click propagation modules (DCG and SCP) to empower user-provided guidance. Based on our extensive experiments showing the superiority of iDet3D in terms of detection accuracy and efficiency, we believe that iDet3D could be a promising option to accelerate data labeling pipelines for LiDAR point clouds.

Future works. In this work, iDet3D shows promising results by only analyzing a single frame. However, most LiDAR scenes are composed of multiple consecutive frames, containing complementary information between each other. We expect that iDet3D can be further improved in a multi-frame scenario by effectively handling point embeddings to be aligned between several sequential frames.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2022R1A2B5B02001913) and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)).

References

Akhtar, A.; Gao, W.; Zhang, X.; Li, L.; Li, Z.; and Liu, S. 2020. Point cloud geometry prediction across spatial scale using deep learning. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 70–73. IEEE.

Arteta, C.; Lempitsky, V.; Noble, J. A.; and Zisserman, A. 2014. Interactive object counting. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III 13*, 504–518. Springer.

Caesar, H.; Yin, S.; Gao, H.; Li, B.; Li, Y.; Chen, Y.; Zhu, S.-E.; Holzer, S.; Kitani, K.; and Koltun, V. 2020. nuScenes: A multimodal dataset for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*.

Chen, C.; Chen, Z.; Zhang, J.; and Tao, D. 2022. Sasa: Semantics-augmented set abstraction for point-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 221–229.

Ding, Z.; Wang, T.; Sun, Q.; and Chen, F. 2022. Rethinking click embedding for deep interactive image segmentation. *IEEE Transactions on Industrial Informatics*, 19(1): 261–273.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, 3354–3361. IEEE.

Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; and Bennamoun, M. 2020. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12): 4338–4364.

Hao, Y.; Liu, Y.; Wu, Z.; Han, L.; Chen, Y.; Chen, G.; Chu, L.; Tang, S.; Yu, Z.; Chen, Z.; et al. 2021. Edgeflow: Achieving practical interactive segmentation with edge-guided flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1551–1560.

He, C.; Zeng, H.; Huang, J.; Hua, X.-S.; and Zhang, L. 2020. Structure aware single-stage 3d object detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11873–11882.

Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; and Markham, A. 2021. Learning semantic segmentation of large-scale point clouds with random sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 8338–8354.

Hu, Z.; Bai, X.; Zhang, R.; Wang, X.; Sun, G.; Fu, H.; and Tai, C.-L. 2022. LiDAL: Inter-frame Uncertainty Based Active Learning for 3D LiDAR Semantic Segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, 248–265. Springer.

Jang, W.-D.; and Kim, C.-S. 2019. Interactive image segmentation via backpropagating refinement scheme. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5297–5306.

Kontogianni, T.; Celikkan, E.; Tang, S.; and Schindler, K. 2022. Interactive Object Segmentation in 3D Point Clouds. *arXiv preprint arXiv:2204.07183*.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12697–12705.

Lee, C.; Park, S.; Song, H.; Ryu, J.; Kim, S.; Kim, H.; Pereira, S.; and Yoo, D. 2022. Interactive Multi-Class Tiny-Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14136–14145.

Li, E.; Wang, S.; Li, C.; Li, D.; Wu, X.; and Hao, Q. 2020a. Sustech points: A portable 3d point cloud interactive annotation platform system. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, 1108–1115. IEEE.

Li, J.; Luo, S.; Zhu, Z.; Dai, H.; Krylov, A. S.; Ding, Y.; and Shao, L. 2020b. 3D IoU-Net: IoU guided 3D object detector for point clouds. *arXiv preprint arXiv:2004.04962*.

Li, Z.; Chen, Q.; and Koltun, V. 2018. Interactive image segmentation with latent diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 577–585.

Lin, Z.; Zhang, Z.; Chen, L.-Z.; Cheng, M.-M.; and Lu, S.-P. 2020. Interactive image segmentation with first click attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13339–13348.

Liu, C.; Gao, C.; Liu, F.; Liu, J.; Meng, D.; and Gao, X. 2022. SS3D: Sparsely-Supervised 3D Object Detection from Point Cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8428–8437.

Liu, Z.; Zhao, X.; Huang, T.; Hu, R.; Zhou, Y.; and Bai, X. 2020. Tanet: Robust 3d object detection from point clouds with triple attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 11677–11684.

Luo, Y.; Chen, Z.; Wang, Z.; Yu, X.; Huang, Z.; and Baktashmotlagh, M. 2023. Exploring Active 3D Object Detection from a Generalization Perspective. In *The Eleventh International Conference on Learning Representations*.

Mao, J.; Shi, S.; Wang, X.; and Li, H. 2022. 3D object detection for autonomous driving: a review and new outlooks. *arXiv preprint arXiv:2206.09474*.

Qi, C. R.; Litany, O.; He, K.; and Guibas, L. J. 2019. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, 9277–9286.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Ranjan, V.; Sharma, U.; Nguyen, T.; and Hoai, M. 2021. Learning to count everything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3394–3403.

Shen, T.; Gao, J.; Kar, A.; and Fidler, S. 2020. Interactive annotation of 3D object geometry using 2D scribbles. In *European Conference on Computer Vision*, 751–767. Springer.

Shi, S.; Wang, X.; and Li, H. 2019. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 770–779.

Shi, S.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. 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*, 43(8): 2647–2664.

Shi, W.; and Rajkumar, R. 2020. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1711–1719.

Sofiiuk, K.; Petrov, I.; Barinova, O.; and Konushin, A. 2020. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8623–8632.

Sofiiuk, K.; Petrov, I. A.; and Konushin, A. 2022. Reviving iterative training with mask guidance for interactive segmentation. In *2022 IEEE International Conference on Image Processing (ICIP)*, 3141–3145. IEEE.

Wang, B.; Wu, V.; Wu, B.; and Keutzer, K. 2019. LATTE: Accelerating LiDAR Point Cloud Annotation via Sensor Fusion, One-Click Annotation, and Tracking. *arXiv preprint arXiv:1904.09085*.

Wang, D. Z.; and Posner, I. 2015. Voting for voting in online point cloud object detection. In *Robotics: science and systems*, volume 1, 10–15. Rome, Italy.

Wang, Y.; Fathi, A.; Kundu, A.; Ross, D. A.; Pantofaru, C.; Funkhouser, T.; and Solomon, J. 2020. Pillar-based object detection for autonomous driving. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, 18–34. Springer.

Wu, T.-H.; Liu, Y.-C.; Huang, Y.-K.; Lee, H.-Y.; Su, H.-T.; Huang, P.-C.; and Hsu, W. H. 2021. Redal: Region-based and diversity-aware active learning for point cloud semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15510–15519.

Xu, J.; Zhang, R.; Dou, J.; Zhu, Y.; Sun, J.; and Pu, S. 2021. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16024–16033.

Xu, N.; Price, B.; Cohen, S.; Yang, J.; and Huang, T. S. 2016. Deep interactive object selection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 373–381.

Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.

Yang, Z.; Sun, Y.; Liu, S.; and Jia, J. 2020. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11040–11048.

Zhang, Y.; Gong, L.; Fan, L.; Ren, P.; Huang, Q.; Bao, H.; and Xu, W. 2019. A late fusion cnn for digital matting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7469–7478.

Zhang, Y.; Hu, Q.; Xu, G.; Ma, Y.; Wan, J.; and Guo, Y. 2022. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18953–18962.

Zheng, W.; Tang, W.; Chen, S.; Jiang, L.; and Fu, C.-W. 2021. Cia-ssd: Confident iou-aware single-stage object detector from point cloud. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 3555–3562.

Zhou, Y.; and Tuzel, O. 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4490–4499.