AS-Det: Active Sampling for Adaptive 3D Object Detection in Point Clouds

Ziheng Ding[1,2], Xiaze Zhang[1,2], Qi Jing[1,2], Ying Cheng[1,2,*], Rui Feng[1,2,*]

[1]School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
[2]Shanghai Collaborative Innovation Center of Intelligent Visual Computing
{zhding24, xzzhang22, jingq22}@m.fudan.edu.cn, {chengy18, fengrui}@fudan.edu.cn

Abstract

3D object detection in point clouds is critical in 3D computer vision, autonomous driving, and robotics. Existing point-based detectors, tailored to handle unstructured raw point clouds, often rely on simplistic sampling strategies to select a subset of points for local representation learning and detection. However, the diverse patterns exhibited by multiple types of point cloud data present a significant challenge to the universality of current detectors, particularly those captured by varied sensors (*e.g.*, LiDAR and 4D Imaging Radar). In response to this challenge, we introduce an adaptable point-based single-stage 3D detector, ***AS-Det***, engineered to excel on both LiDAR and 4D Radar point clouds. Specifically, we propose a novel *active sampling* strategy that actively mines object-related information to achieve efficient sampling and representation across different types of point clouds through end-to-end training. Additionally, we introduce a lightweight *multi-scale center feature aggregation* module to exploit multi-scale object context for precise and low-cost detection. By integrating the abovementioned modules, *AS-Det* achieves highly adaptive detection on various point clouds, encompassing different sensors and scales. Experimental results demonstrate the superior performance and adaptability of *AS-Det* on both LiDAR and 4D Radar point clouds.

Code — https://github.com/eat-slim/AS-Det

1 Introduction

Localizing and recognizing specific 3D objects is a significant issue in computer vision, autonomous driving, and robotics. Point clouds play a pivotal role in 3D detection tasks due to their exceptional ability to represent spatial information. Unlike images, point clouds with elevation information are typically generated by LiDAR or 4D Imaging Radar, which are discrete and unordered points in the 3D scene. LiDAR produces a large number of fixed-resolution points with coordinates and reflection intensity, whereas 4D Radar provides sparser points with additional attributes such as Doppler, Radar Cross-Section (RCS), and Signal-to-Noise Ratio (SNR). Such diversity of patterns poses a great challenge to the adaptability of current 3D detectors.

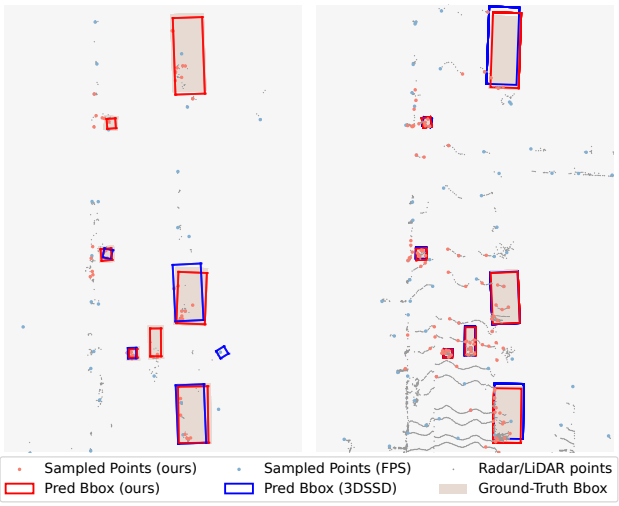

Figure 1: Qualitative comparison of *AS-Det* and the commonly used method. Ours can sample more useful points (red points) to achieve accurate detection on different modalities, *i.e.*, 4D Radar (left) and LiDAR (right).

Basic neural network models cannot process unstructured point sets directly. Some early voxel-based works organize points into regular representation, *e.g.*, multi-view projection images (Chen et al. 2017), followed by well-studied techniques for the images. Subsequent works introduce many voxelization methods, including dense and structured 3D grids (Zhou and Tuzel 2018; Yan, Mao, and Li 2018) and pillars (Lang et al. 2019). Although these methods achieve remarkable accuracy and efficiency in processing LiDAR point clouds (Chen et al. 2023; Li, Luo, and Yang 2023), they unavoidably lose raw information during voxelization. In contrast, detection in 4D Radar may rely more on raw information due to its sparse and low-structured characteristics. Besides, with the advent of various models (Qi et al. 2017a,b) specifically designed for learning from raw point clouds, several methods focus on generating 3D bounding boxes from unordered points. Hierarchical downsampling is an important module for efficiently learning from raw point clouds. Each downsampling layer selects a representative subset from the input points for subsequent

*Corresponding authors: Ying Cheng and Rui Feng.

set abstraction. Determining how to sample points from a large set of scene points is essential for accurate detection. Existing methods commonly rely on either (a) spatial distribution, such as Farthest Point Sampling (FPS) (Shi, Wang, and Li 2019), or (b) feature space (Yang et al. 2020; Zhang et al. 2022), to select foreground points. However, due to the lack of careful consideration of point distribution and contextual information, they usually struggle to adapt to a wide range of point cloud data, including various resolutions (sensors with different beams and numbers), FOVs (front view and 360-degree), and modalities (LiDAR and 4D Radar).

Motivated by these challenges, we propose a point-based single-stage 3D detector, *AS-Det*, capable of adaptive detection across multiple types of point clouds, especially for both LiDAR and 4D Radar point clouds as shown in Fig. 1. To achieve this, we focus on the point sampling strategy, which dominates the information selection in feature learning. We propose a novel downsampling strategy *active sampling*, which excavates instance-related contextual information to sample more valuable points and assign each instance a balanced chance of being sampled. Our *active sampling* transforms the point sampling into a learnable decision task, obtaining the decision results by sampling from the predicted probability distribution. Then, we construct a rational sampling objective to guide the network end-to-end in predicting a sampling distribution advantageous for highly adaptive detection. Furthermore, by introducing the Straight-Through (ST) technique (Bengio, Léonard, and Courville 2013), discrete selection no longer blocks gradient propagation, allowing the detection loss to optimize the predicting distribution directly and enabling the model to explore optimal sampling rules spontaneously. In addition, inspired by discussions on the issue of "center feature missing" in many works (Qi et al. 2019; Yang et al. 2020; Zhang et al. 2022; Fan et al. 2022), we introduce a *multi-scale center feature aggregation* module to regress bounding boxes precisely, which also assists in multilayer *active sampling* for better learning.

Our contributions can be summarized as follows:

1. We present a highly adaptive point-based single-stage 3D detector, *AS-Det*, consisting of *active sampling* and *multi-scale center feature aggregation*, capable of handling point clouds with diverse patterns.
2. We propose a novel *active sampling* strategy that efficiently mines instance-related information and achieves balanced sampling across instances.
3. Experiments conducted on various point clouds showcase that *AS-Det* outperforms other methods on both LiDAR and 4D Radar benchmarks, demonstrating the effectiveness and adaptability of our approach.

2 Related Work

Point-Based methods extract features and predict objects from raw point clouds. Among these methods, hierarchical encoders (Qi et al. 2017b; Qian et al. 2022; Thomas et al. 2019) are widely regarded as efficient structures for learning such unstructured point sets, which are integral to several point-based methods such as PointRCNN (Shi, Wang, and Li 2019). As range sensors only capture surfaces of objects, making it challenging for single-stage methods to regress the object centroid in a single step accurately, VoteNet (Qi et al. 2019) proposes deep Hough Voting to generate credible centroid proposals. Typically, point-based encoders consist of multiple extraction blocks. Each block first samples a subset of input points and then abstracts their vicinity. However, due to the uneven distribution of points between the background and instances, some sampling strategies are proposed to improve sampling efficiency. Early methods like Voxel and D-FPS aim to achieve a more balanced density but exhibit the worst task-oriented property, which is easy to overlook foregrounds. 3DSSD (Yang et al. 2020) believes that points in the background share similar features and thus proposes F-FPS based on feature distance to reduce such background points. However, F-FPS lacks explicit supervision for deep features, and such iterative methods are time-consuming. IA-SSD (Zhang et al. 2022) presents a more straight approach that selects points inside instances by semantic segmentation. Such Seg-based sampling suffers from imbalanced point distribution, leading to a preference for instances that contain more points. Only our *active sampling* exhibits strong task-oriented, balanced sampling across instances and low time complexity attributes.

Voxel-Based methods reorganize point clouds into regular representations, such as voxels (Zhou and Tuzel 2018; Yan, Mao, and Li 2018; Deng et al. 2021) or pillars (Lang et al. 2019; Li, Luo, and Yang 2023), enabling 3D detection akin to image detection. Some methods introduce additional point-based branches to utilize finer-grained raw information to obtain better representations (Shi et al. 2020; Miao et al. 2021; Yang et al. 2023). Although these methods exhibit better performance on LiDAR, their generalizability is limited due to the loss of raw information and the adverse effects of numerous empty voxels, particularly when handling sparser and attributes-richer 4D Radar points. Therefore, recent research (Fan et al. 2022; Chen et al. 2023) suggests that dense feature-based detection is unnecessary, tending to the integration of point-based techniques into voxel-based methods to address the issue of point sparsity.

4D Imaging Millimeter-Wave Radar (4D Radar) emerges as a promising range sensor due to its cost-effectiveness. There are two main differences between LiDAR and 4D Radar: (a) **Distribution:** Point clouds captured by 4D Radar are significantly sparser and often contain much more noise and artifacts owing to multipath propagation. (b) **Attribute:** While LiDAR collects 3D coordinates and reflection intensity, 4D Radar captures additional attributes, such as Doppler velocity, RCS, and SNR, providing valuable characteristics and motion attributes of instances. Due to these differences, most detectors designed for LiDAR struggle to perform satisfactorily in 4D Radar point clouds. Therefore, several specialized methods have been developed for 4D Radar. RCFusion (Zheng et al. 2023) introduces Radar PillarNet for voxelization of Radar point clouds, followed by a modified structure derived from BEV-Fusion (Liu et al. 2023). SMURF (Liu et al. 2024a) integrates a kernel density estimation branch into the PointPillar to mitigate the adverse influence of inherent noise and sparsity in radar point clouds.

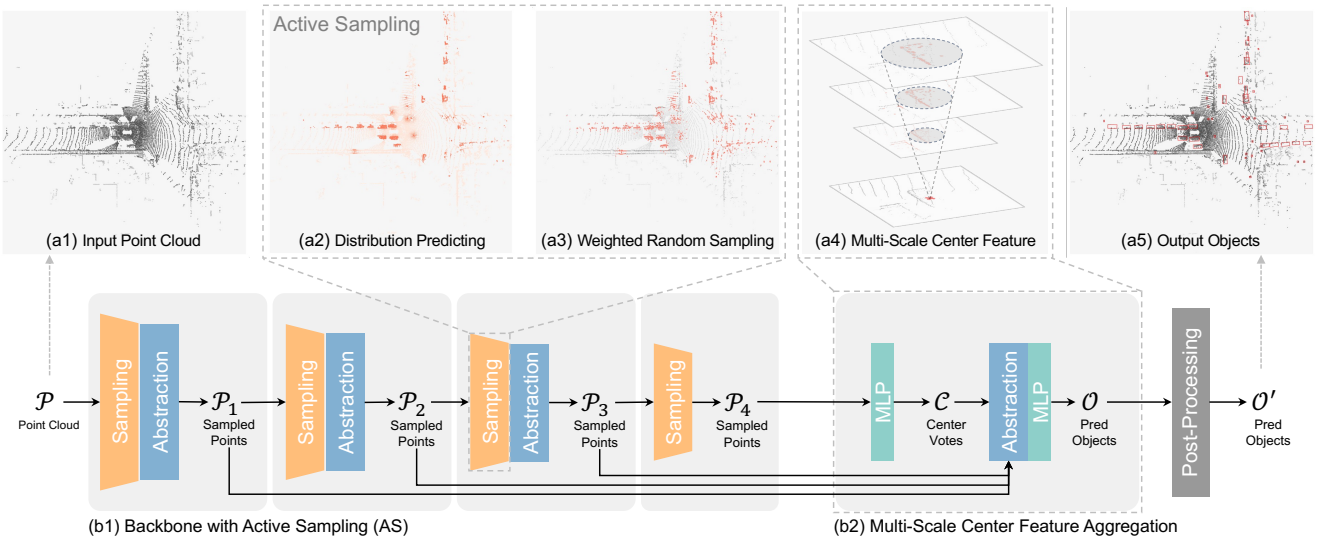

Figure 2: Model overview. The input point cloud (a1) is first passed through a hierarchical encoder (b1) to extract point-wise features, where *active sampling* is used to select representative points (a2,3). The center votes are fed into the *multi-scale center feature aggregation* module (b2) to predict precise objects (a4). Final predictions are obtained after post-processing (a5).

3 Method

3.1 Model Structure

Hierarchical encoders have the potential to learn diverse forms of point clouds, but existing downsampling methods limit their ability to represent specific foreground regions comprehensively. Our proposed *active sampling* (AS) can select instance-related points and be impartial to each instance, intending for better instance representation. As shown in Fig. 2, we propose a point-based single-stage detector, *AS-Det*, which utilizes *active sampling* and *multi-scale center feature aggregation*. The input point cloud is first processed through an encoder to extract point-wise local features, where *active sampling* (Section 3.2) selects a representative subset of instance-related points. Then, multi-scale contextual features are aggregated from object center votes for predicting precise 3D bounding boxes (Section 3.3).

3.2 Active Sampling

The AS strategy employs weighted random sampling to select a representative subset of input points, different from previous iterative or top-k-based methods. A point cloud with N points and C feature channels can be represented as $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_N\} \in \mathbb{R}^{N\times(3+C)}$, where each point $\mathbf{p}_i$ includes its feature $\mathbf{p}_i^{\text{feat}}$ and 3D coordinate $\mathbf{p}_i^{\text{coor}}$.

Pipeline. The *active sampling* strategy follows these steps:
(1) Value Prediction: Each point $\mathbf{p}_i$ is assigned a value v_i using its point-wise feature $\mathbf{p}_i^{\text{feat}}$:

$$v_i = \text{Decider}\left(\mathbf{p}_i^{\text{feat}}\right) = \text{Softplus}\left(\text{MLP}\left(\mathbf{p}_i^{\text{feat}}\right)\right), \quad (1)$$

where Decider is a trainable neural network consisting of MLPs and a Softplus (Dugas et al. 2000) activation function. The value v_i represents the importance of $\mathbf{p}_i$, and Softplus ensures positivity and continuity.

(2) Normalization: The outputs of the Decider are converted into a probability distribution using a normalization function $W = \{v_i / \sum_k v_k \mid i \in 1 \cdots N\}$.
(3) Sampling: Given the sampling number M, the distribution W is used as the weight for weighted random sampling. The representative subset $\mathcal{Q} \in \mathbb{R}^{M\times(3+C)}$ with M points is then obtained and is used for subsequent set abstraction.

Sampling Objective. To supervise the learning process of the aforementioned probability distribution W, we design a reasonable sampling objective tailored to the characteristics of detection in point clouds. It is widely recognized that points situated on instance surfaces yield crucial information regarding their category and size, while nearby points provide contextual scene information that enhances classification accuracy (Yang et al. 2020). Drawing from this prior knowledge, we construct the probability distribution $\hat{W}$ with two key properties: (a) considering feature space, points on or close to instances are assigned higher values to ensure the model prioritizes foreground features, and (b) considering spatial factors, the number of sampled points across different instances should be roughly equivalent to prevent the adverse effects of imbalanced point distributions.

To achieve the first property, we calculate the Euclidean distance between each input point $\mathbf{p}_i$ in $\mathcal{P}$ and the center of the instance in $\hat{\mathcal{O}} = \{\hat{\mathbf{o}}_1, \hat{\mathbf{o}}_2, \cdots, \hat{\mathbf{o}}_K\}$ as $d_{ij} = ||\mathbf{p}_i^{\text{coor}} - \hat{\mathbf{o}}_j^{\text{cent}}||_2$, where $\hat{\mathbf{o}}_j^{\text{cent}}$ is the center of the j-th instance. Next, the *pairing-score* for each pair $\hat{v}'_{ij}$ is calculated as:

$$\hat{v}'_{ij} = \begin{cases} 1, & \mathbf{p}_i \in \hat{\mathbf{o}}_j^{\text{bbox}} \\ \exp\left(-\lambda d_{ij}^2\right), & \mathbf{p}_i \notin \hat{\mathbf{o}}_j^{\text{bbox}} \end{cases}, \quad (2)$$

where $\hat{\mathbf{o}}_j^{\text{bbox}}$ is the bounding box of the j-th instance and symbols $\in/\notin$ indicate whether the point $\mathbf{p}_i$ is inside the box $\hat{\mathbf{o}}_j^{\text{bbox}}$. Then, the *foreground-score* $\hat{v}'_i = \max_j \hat{v}'_{ij}$ of

each point $\mathbf{p}_i$ indicates its relevance to the instance of interest. This method focuses on the foreground while also accounting for contextual background information by applying Gaussian kernels at the centers of instances.

To achieve the second property, we consider the inherent density imbalance resulting from the range sensors' measurement principles. For example, instances near the sensor typically exhibit higher point density, whereas distant instances are only represented by a few points. Therefore, each $\hat{v}_i'$ is scaled to obtain the final point-wise value $\hat{v}_i = \hat{v}_i'/n_i$, where n_i is the neighboring density of the point $\mathbf{p}_i$.

Finally, the normalization function is applied to obtain the objective probability distribution as follows:

$$\hat{W} = \left\{ \hat{w}_i := \frac{\hat{v}_i}{\sum_k \hat{v}_k} \mid i = 1, \ldots, N \right\} , \tag{3}$$

Here, the probability p_j of selecting a point inside a certain instance $\hat{\mathbf{o}}_j$ through a single weighted random sampling following $\hat{W}$ is calculated as:

$$p_j = \sum_i (\hat{w}_i \cdot I_{\{\mathbf{p}_i \in \hat{\mathbf{o}}_j^{\mathrm{bbox}}\}}) , \tag{4}$$

where $I_{\{\mathbf{p}_i \in \hat{\mathbf{o}}_j^{\mathrm{bbox}}\}}$ is an indicator function that denotes whether the point $\mathbf{p}_i$ lies on the instance $\hat{\mathbf{o}}_j$. Let N_j be the number of points inside the $\hat{\mathbf{o}}_j^{\mathrm{bbox}}$. When n_i remains a constant value, neglecting density imbalance, it can be inferred from Eq. (4) that p_j is positively correlated with N_j. This implies that if sampling weight only considers semantic categories, more sampled points will be distributed to clear and simple instances, while challenging instances with fewer points may be overlooked. Specifically, it can also be inferred that when $\mathbf{p}_i \in \hat{\mathbf{o}}_j^{\mathrm{bbox}}$ and n_i equals N_j for all points, the sampling probability remains consistent across all instances, which eliminates the imbalance.

Loss Function. KL divergence is a commonly used method for measuring the loss between distributions (Kingma and Welling 2013). However, our objective distribution $\hat{W}$ exhibits a significant distribution imbalance, *i.e.*, only a small number of foreground points with high objective probabilities while many background ones with extremely low probabilities, especially in high-density large-scale point cloud scenes with abundant background points. The gradient of the KL divergence for any item w_i in the model output probability distribution W can be calculated as:

$$\frac{\partial KL(\hat{W}||W)}{\partial w_i} = \frac{\partial\, \hat{w}_i \left(\ln \hat{w}_i - \ln w_i\right)}{\partial w_i} = -\frac{\hat{w}_i}{w_i} , \tag{5}$$

where the gradient depends on the ratio of the objective value $\hat{w}_i$ to the predicted value w_i. It could lead to the gradient being dominated by numerous background points, neglecting the crucial contribution of fewer foreground points. Hence, we propose the Focal-KL divergence:

$$\mathrm{FKL}\left(\hat{w}_i, w_i\right) = \sum_i \underbrace{\ln\left(\frac{\hat{w}_i - w_i}{\tau} \cdot I_{\{\hat{w}_i > w_i\}} + \mathrm{e}\right)}_{\mathrm{factor}} \cdot \underbrace{w_i \ln\left(\frac{w_i}{\hat{w}_i}\right)}_{\mathrm{kl-diverg.}} , \tag{6}$$

where τ is a scaling parameter and $I_{\{\hat{w}_i > w_i\}}$ is an indicator function that evaluates whether the predicted value w_i is less than the desired objective $\hat{w}_i$. Inspired by focal loss (Lin et al. 2017b), which prioritizes hard samples by utilizing the residual between the ground truth and the prediction, we introduce a similar modulating factor. In the large-scale point cloud mentioned above, the support set for W typically has a large cardinality, causing each w_i in W to tend towards small values. Consequently, the residual between $\hat{w}_i$ and w_i tends to be close to zero, leading to the original focal loss failing to produce distinguishable factors for each item. It is also noted that residuals from different items often have magnitude differences. To address these issues, we utilize τ to amplify the residuals significantly and then apply a logarithmic function to handle magnitude differences. This ensures clear differentiation between the factors of different items without introducing substantial discrepancies. Moreover, our factor primarily emphasizes lower predictions as they likely indicate false negatives of the foreground. FKL reverts to a regular KL divergence when the prediction exceeds the objective. We employ FKL as the sampling probability distribution loss $\mathcal{L}_{\mathrm{sample}}$.

Differentiable Sampling. It is not feasible to consider a fully spontaneous learnable sampling strategy in 3D detection. On the one hand, downsampling is a series of discrete selection operations. Sampling from the output of the model blocks the backpropagation of the gradient, resulting in an untrainable sampler. On the other hand, although there are some techniques to solve the non-differentiability problem, such as reparameterization, they perform poorly because of the large decision space and limited data. However, our objective sampling distribution contains sufficient inductive bias, which supports the introduction of the ST technique, enabling AS to incorporate spontaneous learnable sampling. ST approximates non-differentiable sampling with the identity function in the backpropagation, which has been proven to approximate the gradient with first-order accuracy (Liu et al. 2024b). Combining the exploratory nature of weighted random sampling, $\mathrm{Decider}$ with ST can be directly supervised by the detection objective, thereby spontaneously mining the optimal sampling rules end-to-end.

3.3 Multi-Scale Center Feature Aggregation

Mainstream sparse detectors utilize a center voting strategy to predict accurate boxes (Qi et al. 2019; Yang et al. 2020; Zhang et al. 2022; Fan et al. 2022). This strategy first proposes center coordinates $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_N\} \in \mathbb{R}^{N\times 3}$ of nearby instances from points, and then aggregates contextual features from these centers to predict instances $\mathcal{O} = \{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_N\}$. Based on this architecture, we introduce multi-scale features where the center votes aggregate the outputs of multiple layers of encoders. This approach has two benefits: (a) it shortens the range between each AS layer and the final instance prediction layer, which aids in better optimizing $\mathrm{Decider}$, and (b) it introduces more fine-grained information, similar to the effect of the FPN (Lin et al. 2017a). The multi-scale lightweight aggregation achieves better performance and maintains a similar inference speed compared to existing single-scale aggregation methods with wider and deeper networks. We utilize the smooth-l_1 func-

	Method	Reference	Car 3D AP$_{R40}$				Car BEV AP$_{R40}$			
			easy	moderate	hard	mean	easy	moderate	hard	mean
Voxel	SECOND (Yan, Mao, and Li 2018)	Sensors 2018	83.13	73.66	66.20	74.33	88.07	79.37	77.95	81.80
	SA-SSD (He et al. 2020)	CVPR 2020	88.75	79.79	74.16	80.90	95.03	91.03	85.96	90.67
	PVGNet (Miao et al. 2021)	CVPR 2021	89.94	81.81	77.09	82.95	94.36	91.26	**86.63**	90.75
	RDIoU (Sheng et al. 2022)	ECCV 2022	**90.65**	**82.30**	77.26	**83.40**	94.90	89.75	84.67	89.77
	VoxSet (He et al. 2022)	CVPR 2022	88.53	82.06	**77.46**	82.68	92.70	89.07	86.29	89.35
	PVT-SSD (Yang et al. 2023)	CVPR 2023	90.65	82.29	76.85	83.26	**95.23**	**91.63**	86.43	**91.10**
Point	PointRCNN (Shi, Wang, and Li 2019)	CVPR 2019	86.96	75.64	70.70	77.77	92.13	87.39	82.72	87.41
	Point-GNN (Shi and Rajkumar 2020)	CVPR 2020	88.33	79.47	72.29	80.03	93.11	89.17	83.90	88.73
	3DSSD (Yang et al. 2020)	CVPR 2020	88.36	79.57	74.55	80.83	92.66	89.02	85.86	89.18
	IA-SSD (Zhang et al. 2022)	CVPR 2022	88.87	80.32	75.10	81.43	93.14	89.48	84.42	89.01
	FARP-Net (Xie et al. 2023)	IEEE TMM 2023	88.36	81.53	**78.98**	82.96	91.20	88.45	86.01	88.55
	***AS-Det* (Ours)**	-	**90.55**	**82.41**	77.08	**83.35**	**95.22**	**91.33**	**86.25**	**90.93**

Table 1: Comparison among point-based and single-stage voxel-based methods on the KITTI *test* split set with AP$_{R40}$. The best results of point-based and voxel-based methods are shown in **bold**.

tion to measure residuals from points to their corresponding instance centers as center voting loss $\mathcal{L}_{\text{center}}$. Finally, we use simple MLPs to predict 3D bounding boxes with categories from center votes. For boxes, we regress their center coordinates, sizes, and rotation angles. For categories, we adopt an approach similar to 3DSSD (Yang et al. 2020), using the centerness as the objective. During inference, the output detection result is obtained after regular post-processing.

3.4 Overall Loss

The detection loss $\mathcal{L}_{\text{detection}}$ is the summary of box regression loss $\mathcal{L}_{\text{box}}$ and category centerness loss $\mathcal{L}_{\text{centerness}}$. During training, an instance prediction $\mathbf{o}_i$ generated from a center vote $\mathbf{c}_i$ is assigned to a ground truth $\hat{\mathbf{o}}_j$ iff $\mathbf{c}_i \in \hat{\mathbf{o}}_j^{\text{bbox}}$. We only calculate $\mathcal{L}_{\text{box}}$ for assigned predictions, which follows IoU criteria (Sheng et al. 2022). The $\mathcal{L}_{\text{centerness}}$ is a Binary Cross Entropy (BCE) loss applied to all category predictions. We use a multi-task loss to train all modules in our proposed *AS-Det* jointly:

$$\mathcal{L} = \mathcal{L}_{\text{sample}} + \mathcal{L}_{\text{center}} + \mathcal{L}_{\text{detection}} \,. \tag{7}$$

4 Experimental Analysis

4.1 Benchmarks

We conduct experiments on four autonomous driving datasets across different patterns of point clouds, including two LiDAR datasets and two 4D Radar datasets: **(a) KITTI** (Geiger, Lenz, and Urtasun 2012) is a widely used LiDAR benchmark and focuses on Car, Pedestrian, and Cyclist categories in the front view, which consists of 15k frames, with a single frame valid point of about 20k. **(b) nuScenes** (Caesar et al. 2019) is a 360-degree FOV LiDAR benchmark for larger scenes, including 40k annotated frames. Due to the introduction of temporal information, a single input is usually concatenated from 10 consecutive scans, totaling over 200k points. Meanwhile, it focuses on detecting 10 different forms of traffic elements, further increasing the detection complexity. **(c) VoD** (Palffy et al. 2022) is a pioneering benchmark of 4D Radar, providing 8,693 frames of annotated 4D Radar point clouds with Car, Pedestrian and Cyclist categories. Each point cloud frame has a 180-degree forward FOV, averaging nearly 2k points. **(d) Self-collected dataset** includes 85,919 frames of 4D Radar point clouds with the Car category and a 360-degree FOV collected by ourselves.

4.2 Results on LiDAR Benchmarks

KITTI. Comparison between our *AS-Det* and existing *state-of-the-art* (SoTA) point-based and single-stage voxel-based detectors on the KITTI *test* set is shown in Table 1. On the most compelling Car category of the KITTI dataset, our *AS-Det* broadly outperforms SoTA point-based methods across various types of metrics. We achieve 82.41% AP on the most important 3D *moderate* level, surpassing FARP-Net (Xie et al. 2023) and IA-SSD (Zhang et al. 2022) by $+0.88\%$ AP and $+2.09\%$ AP. Note that FARP-Net is a two-stage method that spends more time refining object proposals, and we achieve an average improvement of 0.39% and 2.38% over it for all difficulty levels of 3D and BEV. Meanwhile, *AS-Det* makes extensive improvements on other metrics to be on par with the SoTA voxel-based method PVT-SSD (Yang et al. 2023), advancing the point-based method to be comparable to voxel-based methods in the LiDAR point cloud. Overall, The experimental results demonstrate the contribution of our *AS-Det* to point-based methods.

nuScenes. NuScenes with larger-scale LiDAR point clouds has consistently posed a great challenge to point-based detectors. To highlight the contribution of our AS strategy in advancing point-based detectors for such challenging scenarios, we compare *AS-Det* with two other point-based detectors using different sampling strategies as shown in Table 2. We apply the best training strategy to all methods for a fair comparison. Our *AS-Det* achieves substantial improvement across multiple categories, outperforming others by more than 10% mAP and leading in most categories, including the important Car category. Our *AS-Det* also makes multi-fold improvements in difficult small objects such as

Method	mAP	NDS	Car	Truck	Bus	Trailer	C.V.	Ped	Mot	Byc	T.C.	Bar
3DSSD (Yang et al. 2020)	29.29	46.25	71.33	32.49	47.69	22.13	8.59	53.93	17.35	0.00	10.27	29.19
3DSSD*	35.49	53.77	72.40	37.13	56.05	**32.34**	**17.29**	57.73	25.75	4.35	10.91	41.00
IA-SSD (Zhang et al. 2022)	31.85	51.61	66.59	40.06	**63.07**	22.62	11.64	53.92	25.38	6.21	11.48	17.54
IA-SSD*	37.01	54.95	70.11	41.38	60.67	26.23	16.81	61.51	31.97	4.91	17.87	38.61
***AS-Det* (Ours)**	**51.92**	**61.80**	**75.58**	**43.28**	58.58	24.70	10.31	**81.66**	**56.44**	**43.51**	**63.89**	**61.24**

Table 2: Comparison on the nuScenes (LiDAR point cloud) *val* split set. The best results are shown in **bold**. * indicates the method is trained using the best practice strategy on nuScenes recommended by Zhu et al. (2019); Chen et al. (2023).

			Entire Annotated Area				Driving Corridor			
	Method	Reference	Car	Pedestrian	Cyclist	Mean	Car	Pedestrian	Cyclist	Mean
Voxel	PointPillars (Lang et al. 2019)	CVPR 2019	37.06	35.04	63.44	45.18	70.15	47.22	85.07	67.48
	CenterPoint (Yin, Zhou, and Krahenbuhl 2021)	CVPR 2021	32.74	38.00	65.51	45.42	62.01	48.18	84.98	65.06
	PillarNeXt (Li, Luo, and Yang 2023)	CVPR 2023	30.81	33.11	62.78	42.23	66.72	39.03	85.08	63.61
	RCFusion (Zheng et al. 2023)	IEEE TIM 2023	39.30	35.10	63.63	46.01	71.65	42.80	83.14	65.86
	LXL (Xiong et al. 2024)	IEEE TIV 2024	32.75	39.65	68.13	46.84	70.26	47.34	87.93	68.51
	SMURF (Liu et al. 2024a)	IEEE TIV 2024	42.31	39.09	71.50	50.97	71.74	50.54	86.87	69.72
Point	3DSSD (Yang et al. 2020)	CVPR 2020	37.98	36.78	69.82	48.19	71.14	46.67	86.19	68.00
	IA-SSD (Zhang et al. 2022)	CVPR 2022	34.18	44.09	74.19	50.82	69.54	52.78	86.07	69.46
	***AS-Det* (Ours)**	-	**42.46**	**49.02**	**78.03**	**56.50**	**77.45**	**59.41**	**96.32**	**77.73**

Table 3: Comparison on the VoD (4D Radar point cloud) *val* split set with AP_{R40}. The best results are shown in **bold**.

Method	Entire Annotated Area	Driving Corridor
3DSSD (Yang et al. 2020)	26.68	57.11
IA-SSD (Zhang et al. 2022)	27.11	51.23
***AS-Det* (Ours)**	**33.47**	**57.99**

Table 4: Comparison on self-collected data (4D Radar point cloud) for Car with AP_{R40}. The best results are **bolded**.

bicycles, motorcycles, and traffic cones, demonstrating its ability to extract details from large-scale scenes.

4.3 Results on 4D Radar Benchmarks

VoD. We compare *AS-Det* with SoTA methods designed specifically for 4D Radar (RCFusion (Zheng et al. 2023), LXL (Xiong et al. 2024), SMURF (Liu et al. 2024a)) and migrated from LiDAR on the VoD benchmark. *AS-Det* significantly outperforms other methods as shown in Table 3. We achieve improvements in all metrics ($+5.53\%$ mAP in "Entire Annotated Area" and $+8.01\%$ mAP in "Driving Corridor") compared to the SoTA method SMURF. Our method also remarkably surpasses others on small objects, delivering around 10% improvement for pedestrians and cyclists. These enhancements demonstrate the better exploitation of fine-grained features by our *AS-Det*. Voxel-based methods no longer dominate 4D Radar point clouds since our single-stage point-based *AS-Det* obviously widens the gap compared to all voxel-based methods. These results illustrate that our method is more adaptable to different range sensors.

Self-collected Dataset. To verify the adaptability of our method further, we collected 4D Radar point clouds with a 360-degree FOV ourselves. We employ four 4D Radars placed in different vehicle orientations and align data with the same coordinate system. Similar to the previous setup, we reproduce two point-based detectors with different sampling strategies and refer to the evaluation metric of VoD as shown in Table 4. our *AS-Det* outperforms IA-SSD with Seg-based sampling and 3DSSD with FS sampling in both "Entire Annotated Area" and "Driving Corridor". We find that many false objects are caused by artifacts in 4D Radar, and using the Seg-based strategy supervised by BCE results in excessive penalties for predicting false positives due to artifacts. In contrast, KL-alike divergence and weighted random sampling encourage the exploration of more sampling distributions and have a higher tolerance for false positives, making it more adaptable to the negative impact of artifacts.

4.4 Ablation Studies

Different Sampling Strategies. We evaluate the effectiveness and adaptability of our AS with commonly used sampling strategies in both LiDAR and 4D Radar point clouds as shown in Table 5. The model with four D-FPS serves as the baseline. Since all sampling strategies (except D-FPS) rely on point-wise deep features, we replace the last two or three D-FPS with other sampling strategies to observe performance changes. All configs are trained with three-class.

For LiDAR point clouds, D-FPS usually performs the worst due to the lack of special treatment for the foreground. By focusing on feature space, FS achieves appreciable improvement compared to D-FPS, while Seg-based improves even further. However, our AS widely achieves better perfor-

Encoder Layer Stack				Speed			LiDAR (KITTI)				4D Radar (VoD)			
#1	#2	#3	#4	#2	#3	#4	Car	Pedestrian	Cyclist	Mean	Car	Pedestrian	Cyclist	Mean
D	D	D	D	9.57 ms	3.74 ms	0.87 ms	82.69	61.45	64.36	69.50	71.61	52.69	86.69	70.33
D	D	F	F	9.57 ms	4.31 ms	2.13 ms	82.86	59.73	71.59	71.39	71.49	52.24	86.68	70.14
D	D	S	S	9.57 ms	0.28 ms	0.28 ms	83.73	61.87	70.60	72.06	66.83	49.12	79.06	65.01
D	D	**A**	**A**	9.57 ms	0.26 ms	0.26 ms	**84.69**	**64.55**	**72.24**	**73.83**	**75.71**	**59.66**	**95.53**	**76.97**
D	F	F	F	19.04 ms	4.31 ms	2.13 ms	83.15	60.62	70.10	71.29	-	-	-	-
D	S	S	S	0.28 ms	0.28 ms	0.28 ms	73.83	53.70	62.05	63.19	-	-	-	-
D	**A**	**A**	**A**	0.26 ms	0.26 ms	0.26 ms	**84.79**	**65.40**	**75.57**	**75.25**	-	-	-	-

Table 5: Comparison w.r.t. different sampling strategies on the LiDAR and 4D Radar dataset with AP_{R40}. Speed represents the sampling runtime per frame of the corresponding layer. **D**: D-FPS, conventional FPS based on Euclidean distance, **F**: Fusion Sampling (FS) proposed by Yang et al. (2020) that fuses D-FPS and feature-based FPS, **S**: Seg-based, a segmentation-based strategy proposed by Zhang et al. (2022), and **A**: *active sampling* (ours).

	KITTI	nuScenes	
$\mathcal{L}_{\text{sample}}$	mAP	mAP	NDS
KL	74.10	48.03	59.09
FKL	**75.25** (+1.15)	**51.92** (+3.89)	**61.80** (+2.71)

Table 6: Effect of FKL in point clouds of different scales.

Head	mAP	Speed(batch=1)	Speed(batch=16)
Vanilla	62.65	47 ms	7.2 ms
VoteHead	72.26	50 ms	8.8 ms
Ours (S=2)	73.06	50 ms	8.2 ms
Ours (S=3)	**75.25**	52 ms	8.9 ms

Table 7: Comparison w.r.t. different heads on the KITTI *val* split set with AP_{R40}. Speed represents the inference runtime per frame of *AS-Det* combined with the corresponding head.

mance by balanced foreground sampling and consideration of contextual information. In addition, when more D-FPS are replaced, the performance of Seg-based significantly degrades, delivering the worst AP. The reason is that input points of the earlier layer are more density-imbalanced. Easy objects with more points usually possess the highest foreground scores, dominating the sampled subset, resulting in the points on other objects not being selected. Our AS addresses it by utilizing weighted random sampling and considering point density, yielding the best results.

For 4D Radar point clouds, we remove the first layer and only experiment with replacing the last two layers. The issue discussed above that the sampled points are concentrated on a few easy objects, also reappears on Seg-based. However, AS still notably outperforms other strategies, demonstrating higher AP for all categories, showcasing its adaptability and robustness across LiDAR and 4D Radar data.

In terms of time consumption, our AS consumes less than 1ms, which is obviously faster than FPS-based methods and ensures efficiency under any sampling scale. The overhead of AS with linear complexity in the detector is negligible.

Effect of FKL divergence. The performance of *AS-Det* applying KL and FKL in point clouds of different scales is shown in Table 6. For KITTI, FKL provides an improvement of 1.15% mAP compared to KL. For nuScenes, which involves much larger-scale point clouds with more than $4\times$ KITTI points, FKL is crucial in addressing the strong imbalance, bringing a 3.89% mAP and 2.71 NDS gain.

Center Feature Aggregation. The effect of using different detection heads on the performance of *AS-Det* is shown in Table 7. Vanilla directly predicts objects from the output of the backbone and shows the worst performance. We construct VoteHead according to 3DSSD and IA-SSD, which applies deep and wide aggregation networks on the single-scale feature and effectively improves mAP. Our center feature aggregation module utilizes a lightweight network to aggregate center features at two scales (S=2), improving AP and accelerating speed over VoteHead. By introducing more fine-grained features, our head with three scales aggregation (S=3) performs best and adds a limited burden. We observe that most of the time consumption occurs by *ball query* with poor parallelism. Through batch processing with a batch size of 16, our speed is only slightly down ($+0.1$ms), but it gets around a 3% mAP increase compared to VoteHead.

5 Conclusion

In this paper, we propose a highly adaptive point-based single-stage detector *AS-Det*. By introducing a novel *active sampling* strategy, we improve the adaptability of the detector to various point clouds. The proposed *multi-scale center feature aggregation* module can also efficiently utilize multi-scale features to obtain fine-grained bounding boxes. Experimental results indicate that our *AS-Det* achieves *state-of-the-art* performance on multiple point cloud data, including LiDAR and 4D Radar with different scales and FOVs.

Limitations. Although our *AS-Det* advances the performance of point-based detectors on large-scale point clouds, this scenario still remains highly challenging for point-based methods and is currently dominated by voxel-based methods. We attribute this to underlying reasons, including model architecture and representation capability. In future work, we will focus on these limitations and explore solutions.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 62172101), the Science and Technology Commission of Shanghai Municipality (No. 23511103102), and the Postdoctoral Fellowship Program of CPSF (No. GZC20230483).

References

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2019. nuScenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*.

Chen, X.; Ma, H.; Wan, J.; Li, B.; and Xia, T. 2017. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 1907–1915.

Chen, Y.; Liu, J.; Zhang, X.; Qi, X.; and Jia, J. 2023. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21674–21683.

Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; and Li, H. 2021. Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection. In *AAAI*.

Dugas, C.; Bengio, Y.; Bélisle, F.; Nadeau, C.; and Garcia, R. 2000. Incorporating second-order functional knowledge for better option pricing. *Advances in neural information processing systems*, 13.

Fan, L.; Wang, F.; Wang, N.; and Zhang, Z.-X. 2022. Fully sparse 3d object detection. *Advances in Neural Information Processing Systems*, 35: 351–363.

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

He, C.; Li, R.; Li, S.; and Zhang, L. 2022. Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8417–8427.

He, C.; Zeng, H.; Huang, J.; Hua, X.-S.; and Zhang, L. 2020. Structure aware single-stage 3D object detection from point cloud. In *CVPR*, 11873–11882.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

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 *CVPR*, 12697–12705.

Li, J.; Luo, C.; and Yang, X. 2023. PillarNeXt: Rethinking network designs for 3D object detection in LiDAR point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17567–17576.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017a. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017b. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Liu, J.; Zhao, Q.; Xiong, W.; Huang, T.; Han, Q.-L.; and Zhu, B. 2024a. SMURF: Spatial Multi-Representation Fusion for 3D Object Detection With 4D Imaging Radar. *IEEE Transactions on Intelligent Vehicles*, 9(1): 799–812.

Liu, L.; Dong, C.; Liu, X.; Yu, B.; and Gao, J. 2024b. Bridging discrete and backpropagation: Straight-through and beyond. *Advances in Neural Information Processing Systems*, 36.

Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D. L.; and Han, S. 2023. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE international conference on robotics and automation (ICRA)*, 2774–2781. IEEE.

Miao, Z.; Chen, J.; Pan, H.; Zhang, R.; Liu, K.; Hao, P.; Zhu, J.; Wang, Y.; and Zhan, X. 2021. Pvgnet: A bottom-up one-stage 3d object detector with integrated multi-level features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3279–3288.

Palffy, A.; Pool, E.; Baratam, S.; Kooij, J. F.; and Gavrila, D. M. 2022. Multi-class road user detection with 3+ 1D radar in the View-of-Delft dataset. *IEEE Robotics and Automation Letters*, 7(2): 4961–4968.

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. 2017a. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 5099–5108.

Qian, G.; Li, Y.; Peng, H.; Mai, J.; Hammoud, H.; Elhoseiny, M.; and Ghanem, B. 2022. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 35: 23192–23204.

Sheng, H.; Cai, S.; Zhao, N.; Deng, B.; Huang, J.; Hua, X.-S.; Zhao, M.-J.; and Lee, G. H. 2022. Rethinking IoU-based optimization for single-stage 3D object detection. In *European Conference on Computer Vision*, 544–561. Springer.

Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. Pv-rcnn: Point-voxel feature set abstraction for 3D object detection. In *CVPR*, 10529–10538.

Shi, S.; Wang, X.; and Li, H. 2019. Pointrcnn: 3D object proposal generation and detection from point cloud. In *CVPR*, 770–779.

Shi, W.; and Rajkumar, R. 2020. Point-gnn: Graph neural network for 3D object detection in a point cloud. In *CVPR*, 1711–1719.

Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of*

the IEEE/CVF international conference on computer vision, 6411–6420.

Xie, T.; Wang, L.; Wang, K.; Li, R.; Zhang, X.; Zhang, H.; Yang, L.; Liu, H.; and Li, J. 2023. FARP-Net: Local-global feature aggregation and relation-aware proposals for 3D object detection. *IEEE Transactions on Multimedia*, 26: 1027–1040.

Xiong, W.; Liu, J.; Huang, T.; Han, Q.-L.; Xia, Y.; and Zhu, B. 2024. LXL: LiDAR Excluded Lean 3D Object Detection With 4D Imaging Radar and Camera Fusion. *IEEE Transactions on Intelligent Vehicles*, 9(1): 79–92.

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

Yang, H.; Wang, W.; Chen, M.; Lin, B.; He, T.; Chen, H.; He, X.; and Ouyang, W. 2023. Pvt-ssd: Single-stage 3d object detector with point-voxel transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13476–13487.

Yang, Z.; Sun, Y.; Liu, S.; and Jia, J. 2020. 3dssd: Point-based 3D single stage object detector. In *CVPR*, 11040–11048.

Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021. Center-based 3d object detection and tracking. In *CVPR*, 11784–11793.

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, L.; Li, S.; Tan, B.; Yang, L.; Chen, S.; Huang, L.; Bai, J.; Zhu, X.; and Ma, Z. 2023. RCFusion: Fusing 4D Radar and Camera with Bird's-Eye View Features for 3D Object Detection. *IEEE Transactions on Instrumentation and Measurement*.

Zhou, Y.; and Tuzel, O. 2018. Voxelnet: End-to-end learning for point cloud based 3D object detection. In *CVPR*, 4490–4499.

Zhu, B.; Jiang, Z.; Zhou, X.; Li, Z.; and Yu, G. 2019. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*.