A Simple Vision Transformer for Weakly Semi-supervised 3D Object Detection

Dingyuan Zhang*[1], Dingkang Liang*[1], Zhikang Zou*[2], Jingyu Li[1], Xiaoqing Ye[2]
Zhe Liu[1], Xiao Tan[2], Xiang Bai†[1]

[1]Huazhong University of Science and Technology, {dyzhang233, dkliang, xbai}@hust.edu.cn
[2] Baidu Inc., China

Abstract

Advanced 3D object detection methods usually rely on large-scale, elaborately labeled datasets to achieve good performance. However, labeling the bounding boxes for the 3D objects is difficult and expensive. Although semi-supervised (SS3D) and weakly-supervised 3D object detection (WS3D) methods can effectively reduce the annotation cost, they suffer from two limitations: 1) their performance is far inferior to the fully-supervised counterparts; 2) they are difficult to adapt to different detectors or scenes (e.g, indoor or outdoor). In this paper, we study weakly semi-supervised 3D object detection (WSS3D) with point annotations, where the dataset comprises a small number of fully labeled and massive weakly labeled data with a single point annotated for each 3D object. To fully exploit the point annotations, we employ the plain and non-hierarchical vision transformer to form a point-to-box converter, termed ViT-WSS3D. By modeling global interactions between LiDAR points and corresponding weak labels, our ViT-WSS3D can generate high-quality pseudo-bounding boxes, which are then used to train any 3D detectors without exhaustive tuning. Extensive experiments on indoor and outdoor datasets (SUN RGBD and KITTI) show the effectiveness of our method. In particular, when only using 10% fully labeled and the rest as point labeled data, our ViT-WSS3D can enable most detectors to achieve similar performance with the oracle model using 100% fully labeled data.

1. Introduction

3D object detection is one of the fundamental tasks in computer vision and has a wide range of real-world applications, such as self-driving and navigation. It aims to regress the 3D bounding boxes and corresponding category labels of objects for a given scene. Due to the inherent limitation of LiDAR sensors, point clouds are usually disordered and sparse, making 3D object detection a challenging task.

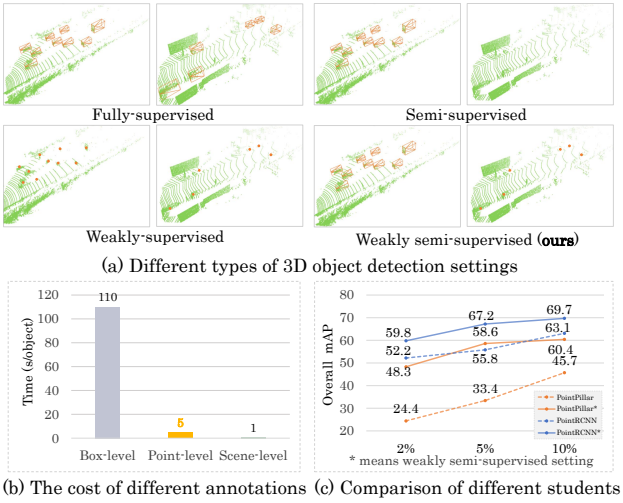


Figure 1. (a) Different types of supervision for 3D object detection. (b) Various annotation formats. The cost of point-level annotations is significantly lower than the box-level annotation. (c) The comparative performance of using the fully-supervised and our weakly semi-supervised settings.

To accurately locate the objects, elaborately annotated large-scale data is inevitable for the existing methods, while labeling 3D bounding boxes is tedious and time-consuming. Recently, some methods [16, 35, 27, 1] have been proposed to reduce the expensive cost of labeling. There are two typical settings: **s**emi-**s**upervised **3**D object **d**etection (SS3D) [35, 47], where only a small amount of precisely annotated scenes are available; **w**eakly-**s**upervised **3**D object **d**etection (WS3D) [27, 37], where coarse annotations (*e.g.*, labeling a point for an object) are used to train the 3D detector instead of precisely annotated 3D bounding boxes.

Although SS3D and WS3D methods can effectively reduce annotation costs, they still have obvious limitations. On the one hand, their performance is still far inferior

*Equal contribution. †Corresponding author.
Work done when Dingkang Liang was an intern at Baidu.

to their fully-supervised counterparts. In specific, the SS3D methods [35, 47] usually transfer knowledge from labeled data to unlabeled data in a teacher-student framework. However, knowledge transfer may be ineffective when the domain gap between labeled and unlabeled data is vast (*e.g.*, labeled and unlabeled data belong to sunny and rain, respectively). For the WS3D methods [27, 37], the supervision information provided by the weak annotation is hard to reflect characters (*e.g.*, the geometric structure) of 3D objects, leading to poor performance. On the other hand, current SS3D and WS3D methods are usually designed for specific frameworks or scenes (*e.g.*, indoor or outdoor), which are hard to transform into other frameworks or scenes. For example, a representative semi-supervised method 3DIoUMatch [35], initially designed for PVRCNN [29], can bring a 4.6% improvement compared with the supervised counterpart under 2% full label setting. However, we empirically find that it can not work well in PointRCNN [31], with only 1.5% improvement achieved.

Considering these issues, training 3D object detectors with considerably low annotation cost by a general paradigm while achieving comparable performance with the fully-supervised counterpart is worth exploring. To achieve this goal, a cheap yet effective annotation format is needed. Among various weak formats (*e.g.*, point-level [37], scene-level [27]), point-level annotation is simple to annotate, convenient to store and use, and localization-aware, which provides a stronger prior of object location. According to the method in [37], a box annotation takes 110 seconds[1], while a point annotation only takes 5 seconds, as shown in Fig. 1 (b).

Nevertheless, only adopting point-level annotations is not enough. A natural way to achieve a good trade-off between detection performance and annotation costs is to combine a small number of fully annotated data, where we treat such a setting as the weakly semi-supervised paradigm. Recently, some methods [3, 44, 8] have demonstrated the potential of weakly semi-supervised paradigm in 2D object detection. These methods help students obtain favorable results and save tremendous resource consumption. Regarding 3D object detection, there is no doubt that replacing 3D bounding boxes with point labels in 3D point clouds is necessary since annotating 3D objects is more time-consuming and labor-intensive than 2D objects. Whereas, how to adopt weakly semi-supervised learning with points to 3D scenes, especially point cloud, has not been explored yet.

In this paper, we aim to explore the **w**eakly **s**emi-**s**upervised **3**D object **d**etection (WSS3D) with points, as shown in Fig. 1 (a). To fully utilize the limited box-level annotations and abundant points, we propose a simple yet effective WSS3D pipeline: 1) Train a point-to-box converter with a small number of fully-labeled data. 2) The trained converter transforms massive point annotations into pseudo-bounding boxes. 3) Finally, train any 3D object detector with fully-labeled and pseudo-labeled scenes in a fully-supervised setting.

The core of such a pipeline is to build a robust point-to-box converter. Recently, vision transformers [4, 38] have shown great potential in feature interaction. Inspired by YOLOS [7] that directly encode the image token as a sequence for object detection, we propose a simple vision transformer-based converter for WSS3D, termed ViT-WSS3D. Specifically, the ViT-WSS3D adopts the plain and non-hierarchical ViT [5] to extract features from point clouds and point annotations. Despite the simple designs, ViT-WSS3D can generate high-quality pseudo boxes through point annotations.

The benefits of ViT-WSS3D are from three aspects: 1) Thanks to vision transformers' strong feature representation ability, our ViT-WSS3D can be extremely simple, which enjoys a plain and non-hierarchical encoder structure without specific domain knowledge for design. 2) The simple and compact ViT-style architecture makes it easy to scale up the model and take advantage of pre-trained technologies (*e.g.*, MAE [11]) proposed in advances of 2D vision. 3) Our method is out-of-the-box, which can be adapted to any 3D object detector without exhaustive tuning and modification.

To demonstrate the effectiveness of our method, we conduct extensive experiments on outdoor KITTI [9] and indoor SUN RGB-D [32] datasets. In particular, with only 10% fully-annotated scenes on both datasets, our ViT-WSS3D can help existing detectors perform closely compared to the 100% fully-supervised counterparts.

2. Related work

2.1. Fully-supervised 3D object detection

Existing 3D object detectors can be roughly categorized into three branches by feature representation: voxel/pillar-based [49, 39, 15, 41, 14, 10], point-based [31, 25, 40, 45, 6], and hybrid-style [29, 30].

For voxel and pillar-based methods, VoxelNet [49] divides a point cloud into equally spaced 3D voxels and uses convolution to extract features. Due to the high expense of 3D convolution, SECOND [39] and PointPillars [15] introduce the sparse convolution and pillars representation, respectively, to increase the speed. CenterPoint [41] flattens representation into an overhead map view and uses an image-based keypoint detector. To better use the voxel feature, the density-aware RoI grid pooling [14] and voxel-based set attention [10] are proposed. For point-based methods, PointRCNN [31] generates proposals via segmentation before refining, and VoteNet [25] handles the sparse nature

[1]Note that some modern softwares [50] may improve labeling processing, but it also accelerates point labeling processing, and the cost gap between full and weak labels still remains.

of point cloud with deep hough voting. To reduce the information loss brought by downsampling, various sampling methods [40, 45] are introduced, and some detectors [6] get rid of downsampling. For hybrid-style methods, PV-RCNN [29, 30] series integrates 3D CNN and point-based set abstraction to learn more discriminative features.

Although these methods have achieved remarkable performance, their success is built on large-scale, elaborately labeled datasets, which is tedious and time-consuming to fulfill such requirements.

2.2. Semi/Weakly-supervised 3D object detection

Two branches of methods have been proposed to reduce the heavy burden of labeling: semi-supervised [47, 35, 16, 13] and weakly-supervised [22, 27, 26, 37] methods.

Semi-supervised methods usually leverage the teacher-student learning framework. Specifically, SESS [47] designs a thorough perturbation scheme and consistency losses to enhance the consistency between predicted proposals. 3DIoUMatch [35] introduces a 3D IoU-based filtering mechanism to filter noisy pseudo-labels. DDS3D [16] proposes a dynamic threshold strategy used to choose high-quality pseudo-labels. Different from the traditional semi-supervised setting, Liu *et al*. [19] propose the first work to explore the sparsely annotated strategy for the 3D object detection task, which only needs to annotate some instances for each scene.

Weakly-supervised methods attempt to recover the loss of information brought by weak labels via various means. Specifically, Qin *et al*. [26] presents a cross-modal knowledge distillation strategy to help students predict results. Ren *et al*. [27] proposes self and cross-task consistency losses with no access to spatial labels at training time. Xu *et al*. [37] makes use of synthetic 3D shapes to complement and refine the real labels. Meng *et al*. [22] propose generating cylindrical object proposals under weak supervision and refining them using a few well-labeled instances.

Although these methods reduce the heavy burden of labeling, their performance is inferior to their fully-supervised counterparts, and they are usually designed for specific frameworks or scenes (e.g., outdoor and indoor). Unlike them, our method can generate more precise pseudo labels to guide students efficiently without making assumptions about students and scenes, which is easy to migrate.

2.3. Vision transformer

The transformer [34] dominates many computer vision tasks [2, 48, 38, 20, 17], attributed to its strong feature extraction ability. 3D point clouds are unordered data and sets, which makes it feasible to utilize transformer [34] to process point clouds. Recently, many transformer-based networks have been proposed for 3D object classification [46], point cloud pre-training [42, 24] and 3D object detection [23, 21, 36, 43]. Different from them, we adopt a plain vision transformer to address the weakly semi-supervised 3D object detection.

3. Our method

3.1. Preliminaries

Problem definition. In this work, we explore weakly semi-supervised 3D object detection, where the datasets consist of a small set of fully-labeled Lidar scenes $\Phi_f = \{(I^i, \phi_f^i)\}_{i=1}^{N_f}$ and massive weak annotated (*i.e*., point annotation) Lidar scenes $\Phi_p = \{(I^i, \phi_p^i)\}_{i=1}^{N_p}$. Specifically, N_f and N_p represent the number of fully-labeled scenes and point-labeled scenes. I^i indicates point clouds of the fully-labeled or point-labeled scenes. The annotations ϕ_f^i of fully-labeled scenes mean the 3D bounding boxes (center, dimension, and orientation) and corresponding class labels, and the ϕ_p^i represents the point annotation with the class label (*i.e*., $[p_i^x, p_i^y, p_i^z, c_i]$) of point-labeled scenes. Note that for point-labeled Lidar scenes, due to the original datasets not providing point-level annotations, we add random disturbances R to the gravities of the 3D bounding boxes to construct ϕ_p.

3.2. Overall framework

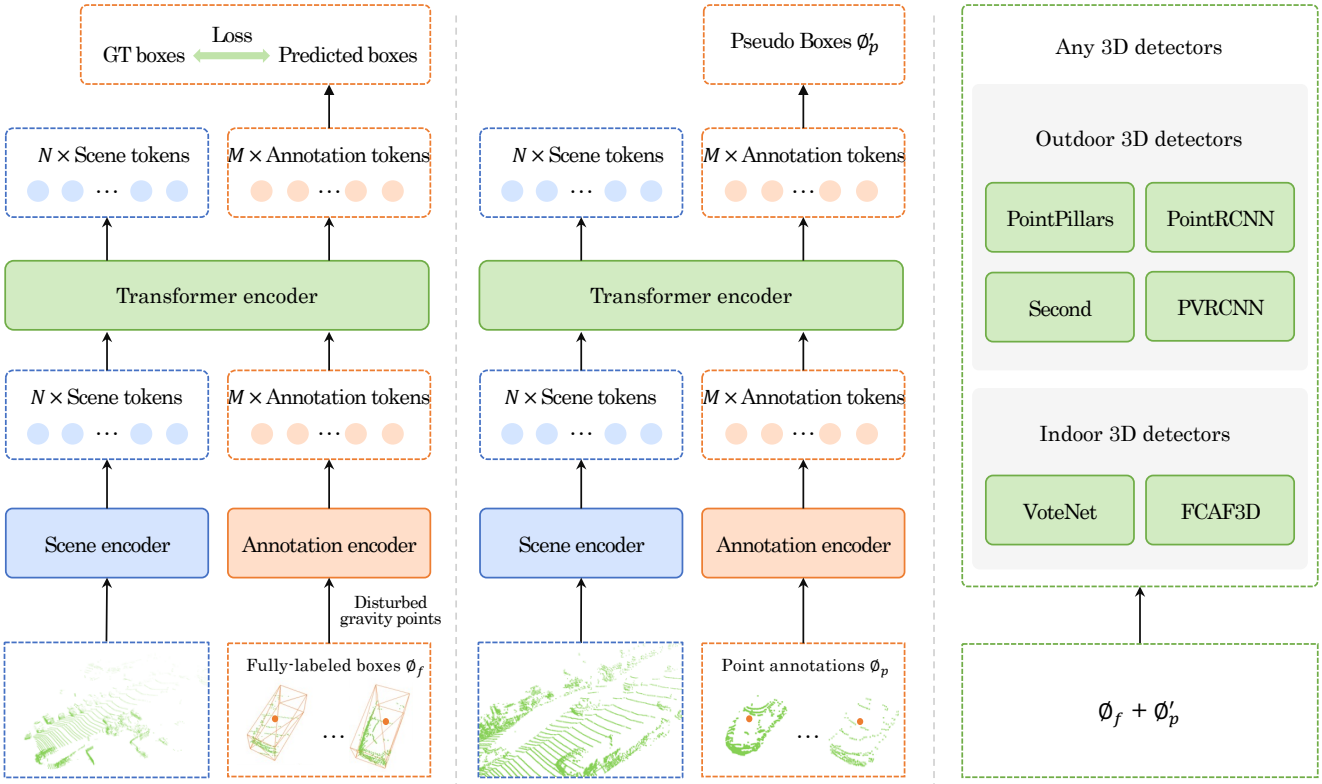
The overview of our method is shown in Fig. 2. The overall process contains three stages:

- Train a point-to-box converter as the teacher model on a small amount of fully-annotated bounding boxes ϕ_f. At this stage, we use the center point with random disturbance R as the simulated point annotations, forcing the converter to recover boxes from noisy points, as shown in Fig. 2 (a).

- Reason pseudo-3D bounding boxes ϕ_p' from massive point annotations ϕ_p using the trained teacher, as shown in Fig. 2 (b). Note that no 3D bounding box is available at this stage, and the trained teacher has to reconstruct the full 3D bounding boxes with only access to point annotations.

- Train any student detectors on fully-annotated bounding boxes ϕ_f together with pseudo boxes ϕ_p' in a fully supervised manner (Fig. 2 (c)). Note that since the whole paradigm makes no assumption about students, **the teacher is entirely independent of students, and they are trained separately.**

In order to better leverage the point annotations, one basic idea of our method is to utilize the point annotations in the forward pass directly and interact with the point cloud features through a plain and non-hierarchical vision transformer [5]. The simple and compact converter contains



(a) Training a point-to-box converter

(b) Transforming the point annotations to the pseudo GT boxes

(c) Using the small fully label data and massive pseudo label data to train a student

Figure 2. The overall framework of our method. (a) We first train a point-to-box converter, which encodes the point annotations and makes global interactions. (b) We then reason the pseudo boxes from the weak point annotations through the trained converter. (c) Finally, we combine the fully labeled boxes and pseudo boxes to train any student in a fully supervised manner, despite the architecture and representation type of students.

a scene encoder, an annotation encoder, a transformer encoder, and a simple detection head. In the following sections, we will introduce our designs for each module and the flexibility of our method in detail.

3.3. Point tokenization

The purpose of tokenization is to embed point cloud and point annotations of a given scene into a meaningful token sequence. The inputs are scene points $I \in \mathbb{R}^{S\times(3+C)}$ and annotation points $\phi_p \in \mathbb{R}^{M\times 3}$, where I is transformed into scene tokens $Z_s \in \mathbb{R}^{N\times D}$ through a scene encoder and ϕ_p is embedded into annotation tokens $Z_a \in \mathbb{R}^{M\times D}$ through an annotation encoder, followed by a simple integration to form a token sequence $Z_0 \in \mathbb{R}^{(N+M)\times D}$.

Scene encoder is used to embed the unordered scene points I into informative tokens named scene tokens Z_s that contain the comprehensive features of a scene. We group scene points into N local patches and map them to the feature space. Specifically, we first use the farthest point sampling (FPS) algorithm to select N key points from original scene points I, then use the kNN algorithm to select k nearest neighbors for each key point to form N patches. To aggregate the local information, points within each local patch are normalized by subtracting the key point of the patch to get relative coordinates. We finally map the unbiased local patches to feature space using mini-PointNet, obtaining scene tokens $Z_s \in \mathbb{R}^{N\times D}$.

Annotation encoder aims to encode the point annotations ϕ_p to useful tokens named annotation tokens Z_a that carry the vital information of point annotations. Because the point annotations contain rich priors about object locations, we want to leverage information as much as possible and not disturb them. Thus we do not group them or make them normalized. Instead, we utilize the naive mini-PointNet as the encoder to embed point annotations into annotation tokens $Z_a \in \mathbb{R}^{M\times D}$. Note that we pad point annotations with zeros to the same M for batch processing since the number of objects varies across scenes.

After the scene and annotation encoding, we need to integrate the outputs into a meaningful token sequence for the subsequent process of the transformer.

$$Z_0 = \left[Z_s^1, ..., Z_s^N; Z_a^1, ..., Z_a^M\right] + E_p. \tag{1}$$

As the Eq. 1 shows, we first stack annotation tokens and scene tokens, then obtain position embedding E_p by applying a multilayer perceptron (MLP) on the center point of each token and add them to get the token sequence $Z_0 \in \mathbb{R}^{(N+M)\times D}$. This simple operation preserves all information in the scene and annotation tokens, which is helpful for feature extraction.

3.4. Transformer encoder

Since we want annotation tokens Z_a interact with scene tokens Z_s without barriers and treat them equally, we use the plain and non-hierarchical ViT [5] to extract features from the input token sequence Z_0. Each transformer encoder layer contains a multi-head attention (MSA), MLP, and two layer normalizations (LN), with residual bypasses inserted after the MSA and MLP, formally written as:

$$Z_l' = \mathrm{MSA}(\mathrm{LN}(Z_{l-1})) + Z_{l-1}, \tag{2}$$

$$Z_l = \mathrm{MLP}(\mathrm{LN}(Z_l')) + Z_l', \tag{3}$$

where Z_l is the output tokens of l-th encoder layer. After the last layer of the encoder, we only output tokens originally from annotation tokens, whose states serve as the feature representations of objects, formally described as:

$$Z_d = \left[Z_L^{N+1}, ..., Z_L^{N+M}\right] \tag{4}$$

where L is the depth of the transformer encoder, Z_L^i is the i-th tokens of the output from the last encoder layer, and Z_d are the detection tokens used for predicting final results.

The plain and non-hierarchical transformer encoder treats scene tokens and annotation tokens equally, enables the direct mutual interaction between them without additional components (*e.g.*, cross-attention), and assists implicit context (*e.g.*, scene layout) representation learning through self-interaction within each type of token. Besides, such a design makes the model easier to scale up. One can tune the model complexity by simply changing the depth of the transformer encoder or modifying the feature dimensions. Moreover, our method can also use the prevalent 2D ViT pre-training paradigm to boost performance without extra cost. In short, our model can evolve with the advance of the 2D vision transformers.

3.5. Detection heads

Thanks to our straightforward design, the detection heads do not need a complicated network architecture and hand-crafted label assignment. Due to the efficiency brought by point annotations and transformer encoder, it is sufficient to use MLPs as detection heads in our method. More specifically, we divide a 3D box into three parts: center, dimension, and orientation, and use an MLP over Z_d to predict each part. For label assignment, since there is little overlap between 3D objects, we assign a ground truth to the prediction corresponding to its center directly to get rid of bipartite matching, which is elegant and practical. We adopt the widely used smooth L1 loss for regression and focal loss [18] for classification to train the teacher.

Table 1. The detailed settings of the model architecture.

Name	ViT-S	ViT-B
Transformer encoder		
Depth	12	12
# Heads	6	6
Feature dims	384	768
Detection head		
# MLPs	3	3
MLP dims	384	768

4. Experiments

4.1. Datasets and metrics

KITTI [9] is one of the most prevalent outdoor datasets centered on autonomous driving-related tasks, which contains 7481 training samples and 7518 testing samples. We divide the training samples into a train split (3712 samples) and a validation split (3769 samples). We use the mAP with 40 recall points and 3D IoU threshold of 0.7(mAP_{40}@0.7), 0.5(mAP_{40}@0.5) and 0.5(mAP_{40}@0.5) for Car, Pedestrian and Cyclist categories, respectively.

SUN RGB-D [32] is an indoor dataset to advance the state-of-the-arts in all major scene understanding tasks, which is captured by four different sensors and contains 10,335 RGB-D images. The whole dataset is split into 5285 samples for training and 5050 samples for validation. We use the mAP with 3D IoU threshold 0.25 (mAP@0.25) as the metrics for 3D detectors.

4.2. Implementation details

We adopt ViT-Small (ViT-S) and ViT-Base (ViT-B [5]) as the default transformer encoders, the detailed settings of the model architecture shown in Tab. 1. There are minor differences between the ViT-S and ViT-B in our method. We only need to change the parameters to get a much more powerful model. This convenience is due to our simple design.

We use ViT-S when there are few fully-labeled scenes (2%, 5% for KITTI and 5% for SUN RGB-D) otherwise ViT-B, since the larger transformer encoder is more powerful and more likely to overfit. For the scene encoder, we set

Table 2. The Comparison results of students with and without pseudo labels under different data settings on KITTI *val* split. We report the mAP_{40}@0.7, mAP_{40}@0.5 and mAP_{40}@0.5 for Car, Pedestrian (Ped.) and Cyclist (Cyc.) categories, respectively.

Students	Settings	Car			Ped.			Cyc.			Overall		
		Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
PointPillars [15]	100% Full	87.1	76.4	73.3	52.6	46.0	41.5	81.0	62.9	58.8	73.6	61.8	57.9
PointPillars [15]	2% Full	51.9	42.4	38.4	16.1	14.1	12.5	28.1	16.8	16.3	32.1	24.4	22.4
PointPillars [15]	2% Full + 98% Weak	75.5	65.4	58.9	50.0	44.1	39.6	53.3	35.4	33.0	59.7	48.3	43.8
PointPillars [15]	5% Full	76.1	64.3	58.9	21.3	18.1	16.4	47.3	28.3	26.6	42.4	33.4	30.7
PointPillars [15]	5% Full + 95% Weak	86.1	74.7	70.2	51.2	45.8	41.1	81.2	55.4	51.9	72.8	58.6	54.4
PointPillars [15]	10% Full	81.8	69.0	64.7	33.7	29.2	26.1	62.8	38.7	35.5	59.9	45.7	42.1
PointPillars [15]	10% Full + 90% Weak	87.1	75.7	71.1	52.4	47.4	42.9	81.4	57.9	54.2	73.6	60.4	56.1
SECOND [39]	100% Full	88.9	79.5	76.1	62.5	54.6	48.6	80.6	64.2	60.6	77.3	66.1	61.7
SECOND [39]	2% Full	85.1	69.2	62.0	38.4	33.7	28.8	73.8	49.5	47.6	65.8	50.8	46.1
SECOND [39]	2% Full + 98% Weak	86.6	70.1	62.3	55.9	49.0	43.0	74.5	51.2	48.9	72.3	56.8	51.4
SECOND [39]	5% Full	86.9	75.3	70.7	53.8	39.8	34.2	79.8	53.1	50.1	73.5	56.1	51.7
SECOND [39]	5% Full + 95% Weak	87.9	75.9	72.9	57.0	51.1	45.5	81.3	58.3	54.8	75.4	61.8	57.7
SECOND [39]	10% Full	87.1	76.2	71.9	53.0	45.9	39.6	82.1	58.1	55.1	74.1	60.1	55.5
SECOND [39]	10% Full + 90% Weak	87.1	77.6	72.9	60.6	53.9	47.6	81.6	60.6	56.4	76.4	64.0	59.0
PointRCNN [31]	100% Full	91.7	80.2	79.6	65.1	59.8	53.7	90.3	72.1	67.7	82.4	70.7	67.0
PointRCNN [31]	2% Full	83.9	70.9	67.7	37.8	34.2	29.1	73.3	51.4	47.6	65.0	52.2	48.1
PointRCNN [31]	2% Full + 98% Weak	83.5	72.7	70.1	57.1	51.9	46.8	72.8	55.0	51.6	73.0	59.8	56.1
PointRCNN [31]	5% Full	87.2	76.1	69.5	45.8	39.9	35.6	77.2	51.5	48.9	70.2	55.8	51.3
PointRCNN [31]	5% Full + 95% Weak	90.6	79.2	76.9	65.6	59.4	53.6	89.7	63.2	59.4	82.0	67.2	63.3
PointRCNN [31]	10% Full	88.8	78.8	74.6	54.2	47.6	40.7	88.8	62.7	59.1	77.2	63.1	58.2
PointRCNN [31]	10% Full + 90% Weak	90.2	79.4	76.8	65.5	61.0	54.5	92.4	68.7	65.0	82.2	69.7	65.5
PVRCNN [29]	100% Full	91.9	83.0	82.4	64.9	57.8	52.9	87.8	70.6	66.3	81.5	70.4	67.2
PVRCNN [29]	2% Full	90.4	76.3	70.6	44.4	39.5	34.8	61.1	38.8	36.6	65.3	51.5	47.3
PVRCNN [29]	2% Full + 98% Weak	84.5	75.8	71.1	61.0	53.4	48.2	67.4	49.7	46.2	71.0	59.6	55.2
PVRCNN [29]	5% Full	91.9	80.1	77.2	55.6	48.4	41.5	75.1	45.7	43.1	74.2	58.1	53.9
PVRCNN [29]	5% Full + 95% Weak	91.3	82.2	79.5	59.9	51.6	47.2	84.3	56.9	53.1	78.5	63.6	59.9
PVRCNN [29]	10% Full	91.2	79.9	77.2	58.2	49.9	44.2	89.2	59.9	56.1	79.5	63.2	59.2
PVRCNN [29]	10% Full + 90% Weak	91.0	82.2	79.8	64.6	58.0	53.4	91.4	67.4	63.1	82.3	69.2	65.4

Table 3. The comparison results of 3DIoUMatch and our method on KITTI *val* split. We take the PVRCNN as the student and report mAP_{40} under the moderate difficulty.

Method	Setting	Car	Ped.	Cyc.	Overall
PVRCNN [29]	100% Full	83.0	57.8	70.6	70.4
PVRCNN [29]	2% Full	76.3	39.5	38.8	51.5
3DIoUMatch [35]	2% Semi	76.9	46.0	45.4	56.1
Ours	2% Full + 98% Weak	75.8	53.4	49.7	59.6
PVRCNN [29]	5% Full	80.1	48.4	45.7	58.1
3DIoUMatch [35]	5% Semi	81.6	48.5	51.7	60.6
Ours	5% Full + 95% Weak	82.2	51.6	56.9	63.6
PVRCNN [29]	10% Full	79.9	49.9	59.9	63.2
3DIoUMatch [35]	10% Semi	82.0	55.0	64.9	67.3
Ours	10% Full + 90% Weak	82.2	58.0	67.4	69.2

the number of nearest neighbors $k = 32$ and the number of scene tokens $N = 2048$ by default. For the annotation encoder, we set $M = 100$ for KITTI and $M = 300$ for SUN RGB-D. We set the random disturbance $R = 0.1$ and $R = 0.0$ for KITTI and SUN RGB-D respectively. When training the teacher, we use horizontal random flip, global rotate scale transform, point shuffle, and GT-Sampling [39] data augmentations on KITTI. Note that we re-generate the database of GT-Sampling for every training set to avoid the labeled data leaking risks. On SUN RGB-D, we only use horizontal random flip and global rotate scale transform augmentations. We train our teacher using 2 NVIDIA GeForce 3090 GPUs, and we need about 8 hours on KITTI and 4 hours on SUN RGB-D to train the teacher.

To perform semi-weakly supervised learning, we first use a fixed stride to uniformly sample fully-labeled scenes from the original dataset, and then prepare weakly-annotated scenes as Sec. 3.1 describes.

4.3. Quantitative results

KITTI. To measure the effectiveness of our method, we choose three ratio levels of fully-labeled scenes (10%,

Table 4. The Comparison results of students with and without pseudo labels under different data settings on SUN RGB-D *val* split. We report mAP@0.25 for all categories.

Students	Settings	Bed	Table	Sofa	Chair	Toilet	Desk	Dresser	Nightstand	Bookshelf	Bathtub	Overall
VoteNet [25]	100% Full	84.5	49.6	68.3	78.0	90.2	25.3	29.2	62.3	35.4	75.1	59.8
VoteNet [25]	5% Full	74.0	32.6	43.6	59.6	66.3	9.1	2.0	38.0	2.2	37.8	36.5
VoteNet [25]	5% Full + 95% Weak	82.8	42.7	59.6	73.8	71.5	22.0	25.0	57.7	12.2	76.4	52.4
VoteNet [25]	10% Full	77.1	35.4	48.2	63.0	73.5	9.3	7.4	45.0	3.1	45.4	40.7
VoteNet [25]	10% Full + 90% Weak	84.6	44.6	63.3	74.4	88.1	22.2	26.6	63.4	21.3	81.7	57.0
VoteNet [25]	20% Full	80.0	43.2	57.9	70.1	78.7	14.6	13.0	50.0	12.7	53.2	47.4
VoteNet [25]	20% Full + 80% Weak	85.9	48.8	65.1	73.2	89.5	27.2	26.9	63.7	29.4	78.0	58.8
FCAF3D [28]	100% Full	87.6	53.9	70.0	81.6	91.9	35.6	38.4	70.1	34.9	76.1	64.0
FCAF3D [28]	5% Full	78.0	41.8	50.7	67.6	71.0	13.3	9.1	44.8	1.0	44.4	42.2
FCAF3D [28]	5% Full + 95% Weak	85.2	45.1	61.4	79.3	84.0	29.3	30.8	62.3	21.3	77.0	57.6
FCAF3D [28]	10% Full	79.3	42.8	56.3	72.0	81.0	17.8	18.0	53.7	15.4	54.5	49.1
FCAF3D [28]	10% Full + 90% Weak	87.0	48.5	66.8	80.4	89.5	32.1	31.5	69.2	26.5	77.0	60.8
FCAF3D [28]	20% Full	82.8	45.6	62.5	74.6	83.6	25.8	25.2	61.2	23.1	70.7	55.5
FCAF3D [28]	20% Full + 80% Weak	87.2	49.2	67.0	80.4	91.3	32.8	34.7	67.8	27.6	72.2	61.0

Table 5. The comparison results of 3DIoUMatch and our method on SUN RGB-D *val* split. We take the VoteNet as the student and report mAP@0.25.

Method	Setting	Overall.
VoteNet [25]	100% Full	59.8
VoteNet [25]	5% Full	36.5
3DIoUMatch [35]	5% Semi	40.0
Ours	5% Full + 95% Weak	52.4
VoteNet [25]	10% Full	40.7
3DIoUMatch [35]	10% Semi	45.0
Ours	10% Full + 90% Weak	57.0
VoteNet [25]	20% Full	47.4
3DIoUMatch [35]	20% Semi	48.8
Ours	20% Full + 80% Weak	58.8

Table 6. Ablation study of scene tokens number N on KITTI *val* split. We report mAP_{40}@0.7, mAP_{40}@0.5, and mAP_{40}@0.5 for Car, Pedestrian and Cyclist under the moderate difficulty.

Scene Token Num N	Student	Car	Ped.	Cyc.	Overall
1024	PointPillars [15]	72.8	44.1	43.7	53.5
	PointRCNN [31]	74.9	53.4	57.3	61.9
2048	PointPillars [15]	74.7	45.8	55.4	58.6
	PointRCNN [31]	79.2	59.4	63.2	67.2
3072	PointPillars [15]	74.8	47.1	56.2	59.4
	PointRCNN [31]	79.4	59.5	63.8	67.6

5% and 2%) to evaluate the performance gain of students brought by our method. We select four typical 3D detectors as students on KITTI that utilize different representations: pillar-based PointPillars [15], voxel-based SECOND [39], point-based PointRCNN [31] and hybrid-style PVRCNN [29].

As shown in Tab. 2, competing with detectors that only trained on a small amount of fully-labeled scenes, pseudo labels generated from our method significantly boost the performance of students. For example, under the 2% full data setting and moderate difficulty, our method helps PointPillars, SECOND, PointRCNN, and PVRCNN gain *23.9%*, *6.0%*, *7.6%* and *8.1%* mAP improvements of overall performance, respectively. Moreover, our method deeply closes the gap between students and their 100% full data counterparts. Note that students under 10% full with 90% weak setting can achieve comparable performance with 100% full baselines, demonstrating the superior quality of pseudo labels from our method.

For each category, we find that Pedestrian and Cyclist are much more sensitive to the ratio of labeled data than Car, and our method works better on these two categories. On Pedestrian and Cyclist, our method greatly benefits all four students. Note that under the 10% full data setting, our method helps all of these detectors achieve comparable or even better performance than baselines trained on 100% fully-labeled data on Pedestrian. Even on less-sensitive Car, our method can still boost all these detectors by a noticeable margin, especially for neat PointPillars.

Furthermore, we conduct comparison experiments between our method and 3DIoUMatch [35], a recent SOTA semi-supervised method on KITTI. We use our split lists to reproduce the 3DIoUmatch. As Tab. 3 shows, there is no significant difference on Car, while our method surpasses 3DIoUMatch conspicuously on data-sensitive Pedestrian and Cyclist. These improvements come from the contribution of point annotations, which brings about a considerable leverage effect with only marginal extra costs, showing the superiority of our method.

SUN RGB-D. To verify the generality of our method, we conduct experiments on the indoor SUN RGB-D

dataset. We select two typical 3D detectors as students on this dataset: point-based VoteNet [25] and voxel-based FCAF3D [28]. Following [35], we use three ratio levels of fully-labeled scenes (20%, 10% and 5%) to evaluate our method on the indoor dataset.

As shown in Tab. 4, our method can essentially improve the performance of students. Compared to labeled-data-only detectors under 5%, 10%, and 20% full data settings, VoteNet gains *15.9%*, *16.3%*, and *11.4%* improvements, and FCAF3D gains *15.4%*, *11.7%*, and *5.5%* improvements on mAP, respectively. Similar to the results on KITTI, our method dramatically narrows the gap between students and 100% full data counterparts.

For each category, our method guides students to achieve significantly better results, especially on data-sensitive categories. For example, when there is only 5% full data available, VoteNet and FCAF3D both only achieve single-digit performance on dresser and bookshelf, while our method doubles their performance several times over, dramatically improving the overall mAPs.

When competing with 3DIoUMatch, the advantage of our approach becomes more significant on SUN RGB-D, shown as Tab. 5. Our method outperforms it by more than *10%* mAP under all settings. This further validates that using point annotations with small costs is worthwhile, and its benefits are universal, whether on indoor or outdoor datasets. Note that the improvement from 3DIoUMatch is minor when there are a large amount of fully-labeled data (*e.g.*, 20%), while our method still boosts students significantly, indicating the superiority of our method.

4.4. Qualitative results

We visualize the pseudo labels from different methods for an intuitive comparison, shown in Fig. 3. On outdoor KITTI, PVRCNN† outputs many false positives (circled in bubbles), influencing the pseudo labels' quality of 3DIoUMatch since the latter is built on PVRCNN. It shows similar issues on indoor SUN RGB-D, where VoteNet† misses some objects, making pseudo labels from 3DIoUMatch differs significantly (marked with arrows) with GTs. The visualization demonstrates the better quality of pseudo labels from our method.

4.5. Ablation study

We conduct ablation experiments on the KITTI dataset under the 5% data setting. We choose PointPillars and PointRCNN as the students of our method and report mAP@0.7, mAP@0.5, and mAP@0.5 for Car, Pedestrian, and Cyclist under the moderate difficulty, respectively.

Effects of the number of scene tokens. To figure out the effects of different numbers of scene tokens N, we conduct ablation experiments, and the results are listed in Tab. 6. It is obvious that too few scene tokens will cause a noticeable performance drop despite what the student is. When using $N = 1024$, the performance drop is up to *5.4%* overall mAP, compared to $N = 2048$. It also depicts that more scene tokens do not necessarily mean much better performance. Using $N = 3072$ only brings marginal improvements (*e.g.*, *0.4%* for PointRCNN) while consuming more computing resources. Thus, $N = 2048$ is a sweet point that balances the performance and resource consumption.

Effects of random disturbance. As Sec. 3.1 describes, we add random disturbances R to the gravities of the 3D bounding boxes to simulate the noisy point annotations. We conduct experiments to study the effects of random disturbance, and results are shown in Tab. 7. The performance of students drops when a greater disturbance is involved in the point annotations, but the slope is slight. Concretely, when $R \leq 0.15m$, the performance drop of both students is no more than *1.1%* overall mAP, compared to $R = 0m$ (*i.e.*, no disturbance) counterparts. The results show that our method is robust and can still achieve stable and favorable performance with noisy point annotations, further loosening the labeling requirements restrictions.

Effects of 2D pre-training. Although 2D images and 3D point clouds are different modalities, they may share some visual concepts. We argue that 2D pre-training can in-

Table 7. Ablation study of random disturbance R on KITTI *val* split. We report mAP$_{40}$ under the moderate difficulty.

Disturbance R	Student	Car	Ped.	Cyc.	Overall
0	PointPillars [15]	75.2	47.1	55.9	59.4
	PointRCNN [31]	79.9	60.5	63.4	67.9
0.05	PointPillars [15]	74.8	46.9	55.8	59.2
	PointRCNN [31]	79.9	59.6	63.3	67.6
0.1	PointPillars [15]	74.7	45.8	55.4	58.6
	PointRCNN [31]	79.2	59.4	63.2	67.2
0.15	PointPillars [15]	74.0	45.7	55.1	58.3
	PointRCNN [31]	78.5	59.3	62.7	66.8

Table 8. Ablation study of different pre-trained ViT weights on KITTI *val* split. We report mAP$_{40}$ under the moderate difficulty. Note that MAE [11] does not provide the official pre-trained weight for ViT-S, so we use DeiT [33] instead.

Data	Teacher	Pre-trained	Student	Car	Ped.	Cyc.	Overall
5%	ViT-S [5]	-	PointPillars [15]	72.8	45.8	51.9	56.8
			PointRCNN [31]	78.4	55.3	62.7	65.5
	ViT-S [5]	DeiT [33]	PointPillars [15]	74.7	45.8	55.4	58.6
			PointRCNN [31]	79.2	59.4	63.2	67.2
10%	ViT-B [5]	DeiT [33]	PointPillars [15]	74.9	46.8	57.2	59.6
			PointRCNN [31]	79.2	59.8	62.3	67.1
	ViT-B [5]	MAE [12]	PointPillars [15]	75.7	47.4	57.9	60.4
			PointRCNN [31]	79.4	61.0	68.7	69.7

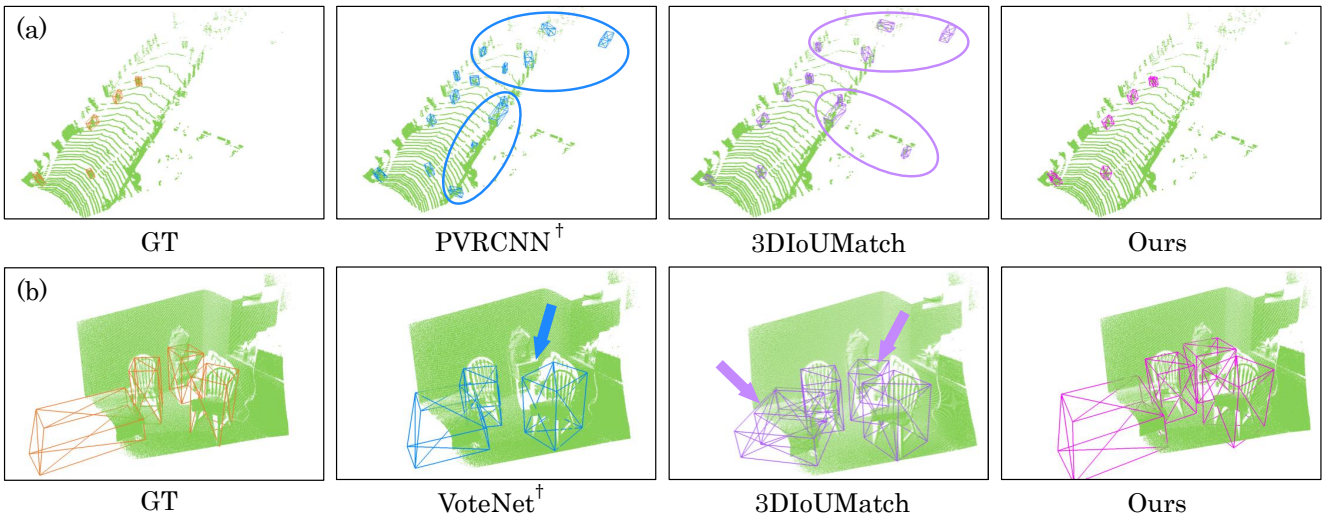


Figure 3. The visualization of pseudo labels from different methods. (a) 2% full data on KITTI. (b) 5% full data on SUN RGB-D. † means training detectors on fully-labeled data and then use them to infer pseudo boxes. The qualitative results demonstrate the better quality of pseudo labels from our method.

ject visual priors into our method and thus help to produce high-quality pseudo labels. Fortunately, our elegant ViT-style architecture makes it feasible to leverage 2D ViT pre-training without modification. We conduct ablation studies using different pre-trained weights. Note that we use ViT-S for the 5% setting and ViT-B for the 10% setting since the larger transformer encoder is more powerful and more likely to overfit. In Tab. 8, it shows that using 2D ViT pre-trained weights brings significant improvements. Under the 5% full data setting with ViT-S, pre-trained DeiT [33] weight helps PointPillars and PointRCNN gain considerable performance. We can also find that using better pre-trained weight means better student performance, especially on categories with fewer data. Under the 10% full data setting with ViT-B, using pre-trained MAE [12] weight is way better than using DeiT weight for PointRCNN on Cyclist, outperforming by *6.4%* mAP. The results indicate that our method has the potential to benefit from the advancement of 2D ViT (*e.g.*, 2D ViT pre-training), which brings sustainability to our approach.

4.6. Limitation

The main limitations are from two aspects: 1) our method can not work well in the few shot scenes, *e.g.*, only 0.5% full labeled data on KITTI (about 18 frames), since the transformer usually requires a certain amount of data to learn. 2) Due to directly transforming the original point cloud into a series of tokens, it may be inefficient to process the large-scale point cloud scenes. In the future, we would like to explore the few-shot WSS3D task and design efficient point tokenization.

5. Conclusion

In this paper, we present the weakly semi-supervised 3D object detection setting (*i.e.*, WSS3D), which supposes the datasets consist of a small number of fully labeled and massive point-labeled data. To fully leverage the point annotations, we propose a simple yet effective transformer baseline named ViT-WSS3D. It employs a plain and non-hierarchical vision transformer to construct global interactions between scene point clouds and point annotations. As a result, high-quality pseudo-bounding boxes are generated, which can be fed into any 3D detectors without extra modification. Extensive experiments on the indoor SUN RGB-D and outdoor KITTI datasets demonstrate the effectiveness and superiority of the proposed ViT-WSS3D.

Acknowledgements. This work was supported by the National Science Fund for Distinguished Young Scholars of China (Grant No.62225603).

References

[1] Benjamin Caine, Rebecca Roelofs, Vijay Vasudevan, Jiquan Ngiam, Yuning Chai, Zhifeng Chen, and Jonathon Shlens. Pseudo-labeling for scalable 3d object detection. *arXiv preprint arXiv:2103.02093*, 2021. 1

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. of European Conference on Computer Vision*, pages 213–229. Springer, 2020. 3

[3] Liangyu Chen, Tong Yang, Xiangyu Zhang, Wei Zhang, and Jian Sun. Points as queries: Weakly semi-supervised object

detection by points. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 8823–8832, 2021. 2

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *Proc. of Intl. Conf. on Learning Representations*, 2020. 2

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *Proc. of Intl. Conf. on Learning Representations*, 2020. 2, 3, 5, 8

[6] 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 *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 8458–8468, 2022. 2, 3

[7] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *Proc. of Advances in Neural Information Processing Systems*, 34:26183–26197, 2021. 2

[8] Yongtao Ge, Qiang Zhou, Xinlong Wang, Chunhua Shen, Zhibin Wang, and Hao Li. Point-teaching: weakly semi-supervised object detection with point annotations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 667–675, 2023. 2

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 2, 5

[10] Chenhang He, Ruihuang Li, Shuai Li, and Lei Zhang. Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 8417–8427, 2022. 2

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 2, 8

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 8, 9

[13] Cheng-Ju Ho, Chen-Hsuan Tai, Yi-Hsuan Tsai, Yen-Yu Lin, and Ming-Hsuan Yang. Learning object-level point augmentor for semi-supervised 3d object detection. *Proc. of British Machine Vision Conference*, 2022. 3

[14] Jordan SK Hu, Tianshu Kuai, and Steven L Waslander. Point density-aware voxels for lidar 3d object detection. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 8469–8478, 2022. 2

[15] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 2, 6, 7, 8

[16] Jingyu Li, Zhe Liu, Jinghua Hou, and Dingkang Liang. Dds3d: Dense pseudo-labels with dynamic threshold for semi-supervised 3d object detection. *arXiv preprint arXiv:2303.05079*, 2023. 1, 3

[17] Dingkang Liang, Wei Xu, and Xiang Bai. An end-to-end transformer model for crowd localization. In *Proc. of European Conference on Computer Vision*, pages 38–54. Springer, 2022. 3

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 2980–2988, 2017. 5

[19] Chuandong Liu, Chenqiang Gao, Fangcen Liu, Jiang Liu, Deyu Meng, and Xinbo Gao. Ss3d: Sparsely-supervised 3d object detection from point cloud. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 8428–8437, June 2022. 3

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 10012–10022, 2021. 3

[21] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 3164–3173, 2021. 3

[22] Qinghao Meng, Wenguan Wang, Tianfei Zhou, Jianbing Shen, Luc Van Gool, and Dengxin Dai. Weakly supervised 3d object detection from lidar point cloud. In *Proc. of European Conference on Computer Vision*, pages 515–531. Springer, 2020. 3

[23] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 2906–2917, 2021. 3

[24] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 604–621. Springer, 2022. 3

[25] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 9277–9286, 2019. 2, 7, 8

[26] Zengyi Qin, Jinglu Wang, and Yan Lu. Weakly supervised 3d object detection from point clouds. In *Proc. of ACM Multimedia*, pages 4144–4152, 2020. 3

[27] Zhongzheng Ren, Ishan Misra, Alexander G Schwing, and Rohit Girdhar. 3d spatial recognition without spatially labeled 3d. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 13204–13213, 2021. 1, 2, 3

[28] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Fcaf3d: fully convolutional anchor-free 3d object detection.

In *Proc. of European Conference on Computer Vision*, pages 477–493. Springer, 2022. 7, 8

[29] 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 *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 2, 3, 6, 7

[30] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *International Journal of Computer Vision*, pages 1–21, 2022. 2, 3

[31] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 770–779, 2019. 2, 6, 7, 8

[32] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 567–576, 2015. 2, 5

[33] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proc. of Intl. Conf. on Machine Learning*, pages 10347–10357. PMLR, 2021. 8, 9

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proc. of Advances in Neural Information Processing Systems*, 30, 2017. 3

[35] He Wang, Yezhen Cong, Or Litany, Yue Gao, and Leonidas J Guibas. 3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 14615–14624, 2021. 1, 2, 3, 6, 7, 8

[36] Yikai Wang, TengQi Ye, Lele Cao, Wenbing Huang, Fuchun Sun, Fengxiang He, and Dacheng Tao. Bridged transformer for vision and point cloud 3d object detection. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 12114–12123, 2022. 3

[37] Xiuwei Xu, Yifan Wang, Yu Zheng, Yongming Rao, Jie Zhou, and Jiwen Lu. Back to reality: Weakly-supervised 3d object detection with shape-guided label enhancement. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 8438–8447, 2022. 1, 2, 3

[38] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Proc. of Advances in Neural Information Processing Systems*, 2022. 2, 3

[39] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2, 6, 7

[40] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 11040–11048, 2020. 2, 3

[41] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 11784–11793, 2021. 2

[42] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 19313–19322, 2022. 3

[43] Dingyuan Zhang, Dingkang Liang, Hongcheng Yang, Zhikang Zou, Xiaoqing Ye, Zhe Liu, and Xiang Bai. Sam3d: Zero-shot 3d object detection via segment anything model. *arXiv preprint arXiv:2306.02245*, 2023. 3

[44] Shilong Zhang, Zhuoran Yu, Liyang Liu, Xinjiang Wang, Aojun Zhou, and Kai Chen. Group r-cnn for weakly semi-supervised object detection with points. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 9417–9426, 2022. 2

[45] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 18953–18962, 2022. 2, 3

[46] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 16259–16268, 2021. 3

[47] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Sess: Self-ensembling semi-supervised 3d object detection. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 11079–11087, 2020. 1, 2, 3

[48] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021. 3

[49] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 2

[50] 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 *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1816–1821. IEEE, 2019. 2