SESS: Self-Ensembling Semi-Supervised 3D Object Detection

Na Zhao Tat-Seng Chua Gim Hee Lee
Deaprtment of Computer Science, National University of Singapore
{nazhao, chuats, gimhee.lee}@comp.nus.edu.sg

Abstract

The performance of existing point cloud-based 3D object detection methods heavily relies on large-scale high-quality 3D annotations. However, such annotations are often tedious and expensive to collect. Semi-supervised learning is a good alternative to mitigate the data annotation issue, but has remained largely unexplored in 3D object detection. Inspired by the recent success of self-ensembling technique in semi-supervised image classification task, we propose SESS, a self-ensembling semi-supervised 3D object detection framework. Specifically, we design a thorough perturbation scheme to enhance generalization of the network on unlabeled and new unseen data. Furthermore, we propose three consistency losses to enforce the consistency between two sets of predicted 3D object proposals, to facilitate the learning of structure and semantic invariances of objects. Extensive experiments conducted on SUN RGB-D and ScanNet datasets demonstrate the effectiveness of SESS in both inductive and transductive semi-supervised 3D object detection. Our SESS achieves competitive performance compared to the state-of-the-art fully-supervised method by using only 50% labeled data. Our code is available at https://github.com/Na-Z/sess.

1. Introduction

Point cloud-based 3D object detection is the task to estimate the object category and oriented 3D bounding box for all objects in the scene. This task has always been a great interest to computer vision and robotics communities due to its potential real-world applications in many areas such as autonomous driving, domestic robotics, augmented/virtual reality, etc. In recent years, many deep learning-based approaches for point cloud-based 3D object detection [1, 7, 9, 11, 12, 16, 17, 18, 24, 27, 29] have emerged and achieved high performances on various benchmark datasets [2, 3, 19]. Despite the impressive performances, most of the existing deep learning-based approaches for 3D object detection on point clouds are strongly supervised and require the availability of a large amount of well-annotated 3D data that is often time-consuming and expensive to collect.

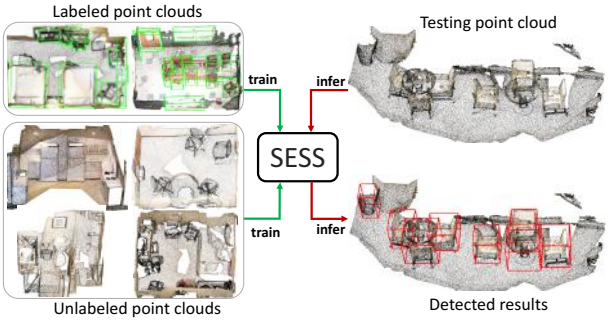


Figure 1: **Semi-supervised 3D object detection pipeline**. Our SESS can predict 3D bounding boxes and semantic labels of objects for an unlabeled scene after training with a mixture of labeled data and unlabeled data.

Semi-supervised learning is a promising alternative to strongly supervised learning for point cloud-based 3D object detection. This is because semi-supervised learning requires only few labeled data, and this largely alleviates the difficulty to collect enormous amount of labeled data. Furthermore, the available few strong labels can still provide the necessary supervision to guide the deep network into learning the correct information for 3D object detection. Information from the few strong labels can also be propagated to the unlabeled data to improve learning. A complete removal of strong labels in the training data would be extremely challenging for the deep network to learn anything meaningful. This is due to the inherent difficulty for a deep network to precisely detect 3D bounding boxes of objects in the point cloud, where points are sparsely distributed, and/or the scene is partially visible and incomplete due to occlusions and 3D amodal perception. To the best of our knowledge, [21] is currently the only existing work to learn a deep network for point cloud-based 3D object detection without strong supervision. More specifically, they propose a cross-category semi-supervised learning where 3D ground truth labels are needed for a set of object classes, and 2D ground truth labels are required for all object classes. Although promising results are achieved in [21], the approach requires RGB-D input and does not work on pure 3D point

clouds. Moreover, it still requires a large amount of 3D labels on the strong object classes.

In view of the potential of semi-supervised learning and limitations in [21], we address the in-category semi-supervised 3D object detection problem with 3D point cloud as the only input in this paper. In contrast to cross-category semi-supervision, in-category semi-supervision means that the training data contains few strongly labeled point clouds and a large number of unlabeled point clouds. Furthermore, the strongly labeled point clouds are assumed to contain all object classes of interests, albeit few examples per object class. To this end, we propose SESS - a **S**elf-**E**nsembling **S**emi-**S**upervised 3D object detection framework for point clouds. More specifically, our SESS achieves semi-supervision with a Mean Teacher paradigm [22] that contains a teacher and student 3D object detection network. The teacher guides the predictions of the student to be consistent with its predictions under random perturbations, where these predictions are sets of 3D object proposals. In other words, we want the 3D object proposals from both teacher and student networks to be aligned at the end of the training stage. We propose three consistency losses based on the center, class and size of the 3D object proposals to encourage alignment of the 3D object proposals from the teacher and student networks. Our three consistency losses encode both geometry and semantic information to guide the network towards learning precise coordinates of the 3D bounding boxes and accurate object categories. We conduct experiments of our SESS framework on two benchmark datasets. Promising results over baseline and strongly supervised approaches validate our semi-supervised learning approach for the challenging task of point cloud-based 3D object detection in both inductive and transductive semi-supervised learning settings.

2. Related work

2.1. 3D Object Detection

A number of approaches have been proposed for 3D object detection task, which can be briefly summarized into three different types based on their input data formats: 2D projection [8, 9, 18, 26], voxel grid [1, 7, 15, 20, 16, 25, 29], and point cloud [5, 11, 12, 17, 24, 27, 28]. The 2D projection and voxel grid based methods are proposed to circumvent the difficulty in processing irregular point clouds by either projecting 3D data into 2D representations (*e.g.* front-view, or bird's eye view) or voxelizing it into regular grids. To efficiently localize 3D objects in the point cloud of a 3D space, [5, 12, 24] leverage on mature 2D object detectors to trim a 3D bounding frustum for each detected object, for 3D search space reduction, while [11, 17, 27, 28] explore the sparsity of 3D data and generate 3D proposals around seed points that are determined by different manners (*e.g.* segmenting [17] or voting [11]). Despite the significant improvement achieved by the existing detection models, a large number of high-quality 3D ground truths are required for training. This limits their applicability in practice, where the ground truths are expensive to acquire.

In order to alleviate the limitation and leverage the abundant unlabeled data that are easier to access, semi-supervised 3D object detection is a promising direction to exploit. However, there is no existing semi-supervised point cloud-based 3D object detection approach that only involves a small set of labeled data. The most closely related work is proposed recently by Tang and Lee in [21]. They propose a cross-category semi-supervised 3D object detection method. However, it requires all the 2D box labels and some of the 3D box labels. We consider this setting as "mix supervised" to differentiate with our semi-supervised setting where few labeled samples are used with plentiful of unlabeled samples. Furthermore, [21] follows the two-step pipeline in [12] to restrict the object localizing space: the first step is 2D object detection on RGB images and the second step is 3D object detection in the frustum point clouds yielded from the 2D detections. This two-step pipeline means that the performance is tightly dependent on the performance of the 2D detector. In this work, we directly process the raw point cloud in one step to remove the dependency on 2D modality.

2.2. Semi-Supervised Learning

Semi-Supervised Learning (SSL) attracts growing interest in a wide range of research areas (*e.g.* image classification and segmentation) by virtue of its aim to learn from both labeled and unlabeled data simultaneously. Many approaches have been proposed to solve SSL. Due to the space limitation, we only review self-ensembling based approaches, the most promising direction in SSL recently.

The idea behind self-ensembling approaches is to improve the generalization of a model by encouraging consensus among ensemble predictions of unknown samples under small perturbations of inputs or network parameters. For instance, Γ model [13], a variation of ladder network [23], consists of two identical parallel branches that respectively take one image and the corrupted version of the image as input. The consistency loss is computed based on the difference between the (pre-activated) predictions from the clean branch and the (pre-activated) corrupted branches processed by an explicit denoising layer. In contrast to Γ model, Π model [6] discards the explicit denoising layer and inputs the same image with different corruption conditions into a single branch. Virtual Adversarial Training [10] shares similar idea with the Π model but uses adversarial perturbation instead of independent noise. Temporal model [6], an extension of Π model, forces the consistency between the recent network output and the aggregation of network pre-

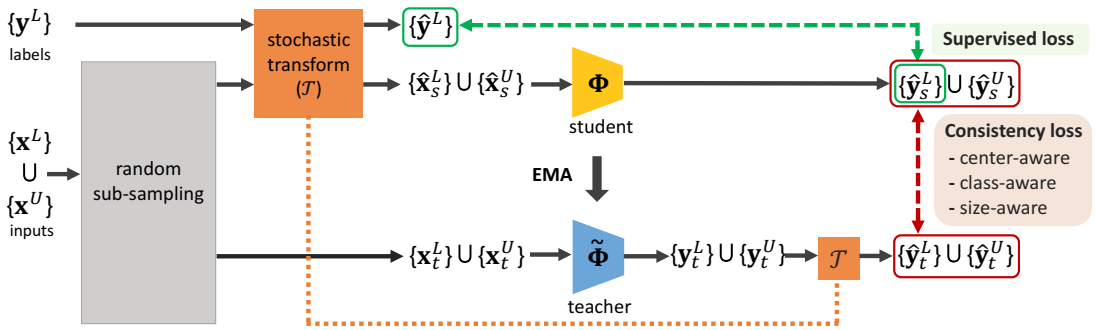


Figure 2: The architecture of our SESS. In this figure, a training batch, including a set of labeled sample $\{\mathbf{x}^L\}$ and a set of unlabeled samples $\{\mathbf{x}^U\}$, is passed through different perturbations and then input into the student and the teacher network, respectively. The predictions of the student network are compared with the corresponding ground truth labels $\{\mathbf{y}^L\}$ processed by the same transformation $\mathcal{T}$ using supervised loss and with the teacher predictions processed by the same transformation $\mathcal{T}$ using consistency loss.

dictions over multiple previous training epochs rather than predictions from auxiliary corrupted input. However, this model becomes cumbersome when applied to large dataset because it needs to maintain a per-sample moving average of the historical network predictions. Mean Teacher [22] tackles the weakness of temporal model by replacing network prediction average with network parameter average. It contains two network branches - *teacher* and *student* with the same architecture. The parameters of the teacher are the exponential moving average of the student network parameters that are updated by stochastic gradient descent. The student network is trained to yield consistent predictions with the teacher network. We choose the Mean Teacher paradigm as the basis of our framework, and adapt it to the 3D object detection task.

3. Our Method

3.1. Problem Definition

Given any point cloud of a scene as input, our objective is to classify and localize amodal 3D bounding boxes for objects in the 3D scene. In the semi-supervised setting, we have access to N training samples, including N_l labeled point clouds $\mathcal{P}^L = \{\mathbf{x}_i^L, \mathbf{y}_i^L\}_{i=1}^{N_l}$ and N_u unlabeled point clouds $\mathcal{P}^U = \{\mathbf{x}_i^U\}_{i=1}^{N_u}$. Here $\mathbf{x}_i \in \mathbb{R}^{n\times 3}$ denotes the point cloud of a 3D scene, containing n points with coordinates; and $\mathbf{y}_i^L$ denotes the ground truth annotations for all the interested objects in the 3D point cloud $\mathbf{x}_i^L$. Each object is represented by a semantic class s (1-of-K predefined classes) and an amodal 3D bounding box parameterized by its center $c = (c^x, c^y, c^z)$, size $d = (l, w, h)$, and orientation θ along the upright-axis.

3.2. SESS Architecture

The illustration of our SESS architecture is shown in Figure 2. We use the Mean Teacher paradigm [22] in our semi-supervised 3D object detection task, where the student and the teacher networks are 3D object detectors. The student and teacher networks take the perturbed point clouds as input and output the 3D object proposals, which represent the estimated classes and 3D bounding boxes of all the objects of interest in the point cloud. We adopt the state-of-the-art VoteNet[1] [11] as our backbone for the student and teacher networks. More specifically, SESS takes a training batch with a mixture of labeled and unlabeled point clouds: $\{\mathbf{x}_i^L\}_{i=1}^{B_l} \cup \{\mathbf{x}_i^U\}_{i=1}^{B_u}$, where B_l and B_u denote the labeled and unlabeled samples in a batch, respectively. We randomly sample M points from each training point cloud, *i.e.* $\mathbf{x}^L$ or $\mathbf{x}^U$, twice to get two sets of points. The first set of points $\mathbf{x}_s$ is perturbed into $\hat{\mathbf{x}}_s$ by a stochastic transformation $\mathcal{T}$ and then passed to the student network, while the second set of points $\mathbf{x}_t$ is directly passed to the teacher network. The output proposals from the teacher network $\mathbf{y}_t$ are further transformed to $\hat{\mathbf{y}}_t$ by the $\mathcal{T}$ applied on $\mathbf{x}_s$ previously. For each proposal in $\hat{\mathbf{y}}_t$, we find its closest alignment from the output proposals of the student network $\hat{\mathbf{y}}_s$ based on the Euclidean distance. Subsequently, the error between each aligned proposal pair is computed from three consistency losses. Concurrently, the set of ground truths $\mathbf{y}^L$ is also transformed by the same $\mathcal{T}$ applied on $\mathbf{x}_s^L$, and the transformed $\hat{\mathbf{y}}^L$ is compared with the labeled output of the student network $\hat{\mathbf{y}}_s^L$ using a supervised loss. Finally, the parameters of the student network Φ is updated via gradient descent at training step t, and then the updated parameters from the student network are used in an exponential moving average (EMA) to update the parameters of the teacher network $\tilde{\Phi}$:

$$\tilde{\Phi}_{t+1} = \alpha\tilde{\Phi}_t + (1-\alpha)\Phi_t, \tag{1}$$

where α is a smoothing hyper-parameter that controls how much information the teacher takes from the student network. For supervised loss, we take the same multi-task loss

[1]It is worth highlighting that instead of designing a specific detector model, our proposed framework is model-agnostic and any existing point cloud-based 3D object detection network can be used.

as in [11]. We will introduce our perturbation scheme and consistency losses for adapting the Mean Teacher paradigm into the 3D object detection task below.

3.3. Perturbation Scheme

As mentioned in [6, 22], input perturbation or data augmentation play an essential role in the success of self-ensembling approaches. The perturbation schemes of the Mean Teacher on image-based tasks, *e.g*. image recognition, include random translations and horizontal flips of the input images, adding Gaussian noises on the input layer, and applying dropouts within the network. However, none of the image-based perturbation schemes can be used directly for our point cloud-based 3D object detection task. Consequently, we propose a perturbation scheme suitable for point cloud-based 3D object detection in this paper.

Random Sub-sampling. We apply random sub-sampling on the input point cloud to both the student and teacher networks as part of our perturbation scheme. The local geometrical relationship of the points in two random sub-samples of a given point cloud might differ significantly, but the global geometry, *i.e*. the 3D bounding box locations of the objects, in the sub-sampled point clouds should remain the same. As a result, our model is trained to exploit the underlying geometry in the global context by forcing the consistency between the stochastic outputs from the student and teacher networks.

Stochastic Transform. We apply stochastic transformations that include flipping, rotation and scaling on the randomly sub-sampled point cloud for the student network to prevent the network from memorizing unintended properties of the training point clouds, *e.g*. the absolute position of each point. More specifically, we formulate the transformation operations as a set of stochastic variables $\mathcal{T} = \{\mathcal{F}_x, \mathcal{F}_y, \mathcal{R}, \mathcal{S}\}$. Here $\mathcal{F}_x$ represents a random flip along the x-axis, and its binary value is determined by:

$$\mathcal{F}_x = \begin{cases} 1 & if\ \epsilon > 0.5, \\ 0 & otherwise, \end{cases} \tag{2}$$

where ϵ is a random variable uniformly sampled from $[0, 1]$. $\mathcal{F}_y$ represents a random flip along the y-axis and is generated the same way as $\mathcal{F}_x$. $\mathcal{R}$ denotes the rotation around the upright-axis, paramterized by a rotation angle ω sampled uniformly from $[-\vartheta, +\vartheta]$:

$$\mathcal{R}(\omega) = \begin{bmatrix} \cos(\omega) & -\sin(\omega) & 0 \\ \sin(\omega) & \cos(\omega) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

And $\mathcal{S}$ that is uniformly sampled from $[a, b]$ represents the scaling of the points. Finally, a $\mathcal{T}_i$ is randomly sampled and applied on each input training point cloud $\mathbf{x}_s$ to the student network as: $\hat{\mathbf{x}}_s = \mathcal{T}_i * \mathbf{x}_s$. Note that the ground truth labels $\mathbf{y}_i^L$ of the labeled input point cloud $\mathbf{x}_i^L$ are also transformed by the corresponding $\mathcal{T}_i$ before computing the supervised loss. Additionally, the output proposals $\mathbf{y}_t$ from the teacher network are also transformed by $\mathcal{T}_i$ to enable the alignment between outputs of the two networks.

3.4. Consistency Loss

Unlike the direct computation of consistency between class predictions of perturbed images in the context of recognition task [22], the consistency between two sets of 3D object proposals cannot be computed directly. We circumvent this problem by pairing up the predicted proposals from the student and teacher networks with an alignment scheme, followed by applying three consistency losses on the paired proposals. The objective of the three consistency losses is to enforce the consensus of object locations, semantic categories and sizes. Let $\hat{C}_s = \{\hat{c}_s\}$ denotes the centers of the predicted 3D bounding boxes from the student network, and $\hat{C}_t = \{\hat{c}_t\}$ denotes those from the teacher network after transformation. For each $\hat{c}_t \in \hat{C}_t$, we do the alignment by searching for the its nearest neighbor in $\hat{C}_s$ based on the minimum Euclidean distance between the centers of the bounding boxes. We further use $\hat{C}_s^{\mathcal{A}}$ to denote the elements from $\hat{C}_s$ that are aligned with each element in $\hat{C}_t$. More formally,

$$\begin{aligned} &\hat{C}_s^{\mathcal{A}} = \{\cdots, \hat{c}_{s_j}^{\mathcal{A}}, \cdots\} : \\ &\hat{c}_{s_j}^{\mathcal{A}} = \arg\min_{\hat{c}_s} \left\| \hat{c}_s - \hat{c}_{t_j} \right\|_2, \ \forall \hat{c}_s \in \hat{C}_s. \end{aligned} \tag{4}$$

Similarly, we can also collect $\hat{C}_t^{\mathcal{A}}$ with elements from $\hat{C}_t$ that are aligned with each element in $\hat{C}_s$. It is important to note that the alignments $\hat{C}_s^{\mathcal{A}}$ and $\hat{C}_t^{\mathcal{A}}$ are not bijective, hence $\hat{C}_s^{\mathcal{A}} \neq \hat{C}_t^{\mathcal{A}}$. Intuitively, the alignment errors, *i.e*., the total distance between all corresponding elements in $\hat{C}_s^{\mathcal{A}} \leftrightarrow \hat{C}_t$ and $\hat{C}_t^{\mathcal{A}} \leftrightarrow \hat{C}_s$, should be zero when the bounding boxes predicted by the teacher and student networks are consistent. Thus, we propose the **center-aware consistency loss**:

$$\mathcal{L}_{center} = \frac{\sum_{\hat{c}_s} \|\hat{c}_s - \hat{c}_t^{\mathcal{A}}\|_2 + \sum_{\hat{c}_t} \|\hat{c}_t - \hat{c}_s^{\mathcal{A}}\|_2}{|\hat{C}_s| + |\hat{C}_t|}, \tag{5}$$

to minimize the alignment errors between the teacher and student network.

In addition to center consistency, we also consider two other properties of the 3D proposals: semantic class and size to enforce the consistency between two sets of proposals. Following the principle in classic self-ensembling learning, where the teacher network produces targets for the student to learn, we only consider a uni-directional alignment, *i.e*., $\hat{C}_t$ to $\hat{C}_s^{\mathcal{A}}$ in computing the class- and size-aware consistency losses. More specifically, let $\hat{P}_s = \{\hat{p}_s\}$ and $\hat{P}_t = \{\hat{p}_t\}$ denote the class probabilities of the predicted objects from the student and the teacher network, respectively. The aligned $\hat{P}_s^{\mathcal{A}} = \{\hat{p}_s^{\mathcal{A}}\}$ is easily obtained based

on minimum center distance. We define the **class-aware consistency loss** as the Kullback-Leibler (KL) divergence between $\hat{P}_s^{\mathcal{A}}$ and $\hat{P}_t$:

$$\mathcal{L}_{class} = \frac{1}{|\hat{P}_t|} \sum D_{KL}(\hat{p}_s^{\mathcal{A}} \parallel \hat{p}_t). \quad (6)$$

In similar vein, the sizes of the bounding boxes predicted by the student and the teacher networks are denoted as $\hat{D}_s = \{\hat{d}_s\}$ and $\hat{D}_t = \{\hat{d}_t\}$, respectively. We use the same minimum center distance to get the aligned $\hat{D}_s^{\mathcal{A}} = \{\hat{d}_s^{\mathcal{A}}\}$. The **size-aware consistency loss** can now be computed as the Mean Square Error (MSE) between $\hat{D}_s^{\mathcal{A}}$ and $\hat{D}_t$:

$$\mathcal{L}_{size} = \frac{1}{|\hat{D}_t|} \sum (\hat{d}_s^{\mathcal{A}} - \hat{d}_t)^2. \quad (7)$$

Finally, the total consistency loss is a weighted sum of all the three consistency terms described earlier:

$$\mathcal{L}_{consistency} = \lambda_1 \mathcal{L}_{center} + \lambda_2 \mathcal{L}_{class} + \lambda_3 \mathcal{L}_{size}, \quad (8)$$

where λ_1, λ_2, and λ_3 are the weights to control the importance of the corresponding consistency term.

4. Experiments

4.1. Datasets

We evaluate our SESS on SUN RGB-D and ScanNet for semi-supervised 3D object detection.

SUN RGB-D [19] is an indoor benchmark dataset for 3D object detection. It contains 10,335 single-view RGB-D images, which are officially split into 5,285 training samples and 5,050 validation samples, where 3D bounding box annotations for hundreds of object classes are available. Followed the standard evaluation protocol [5, 11, 12, 15, 21], we perform evaluation on the 10 most common categories for comparing with the previous methods. By using the provided camera parameters, the depth images are converted to point clouds as our inputs.

ScanNetV2 [2] contains 1,513 reconstructed meshes from 707 unique indoor scenes, which are officially split into 1,201 training samples and 312 validation samples. Each scene is well annotated with semantic segmentation masks. Since there is no existing amodal or orientated 3D bounding box in ScanNetV2 dataset, we derive the axis-aligned bounding boxes from the point-level labeling as in [4, 11]. We adopt the same 18 object classes out of the 21 semantic classes as proposed in [4, 11]. The input point clouds are generated by sampling vertices from meshes.

For both datasets, we evaluate on different proportions of labeled data randomly sampled from all the training data. We ensure that all classes are present, or otherwise we re-sample until all the K classes are covered in the labeled set. We keep the remaining data as unlabeled data for training in our semi-supervised framework.

4.2. Implementation Details

Framework Details. We feed training batches of point clouds with 5,000 points to our framework. To construct a batch, we randomly sample B_l labeled samples from $\mathcal{P}^L$ and B_u unlabeled samples from $\mathcal{P}^U$. In the experiments, B_l is set to 2 and B_u to 8. During the perturbation step, the number of randomly sub-sampled points is 4,000; the ϑ is set to 30° on SUN RGB-D and 5° on ScanNetV2; the random scale range is bounded by $a = 0.85$ and $b = 1.15$. The weights in the consistency loss function are set as $\lambda_1 = 1$, $\lambda_1 = 2$, $\lambda_3 = 1$. As suggested in [22], we ramp up the coefficient of consistency cost from 0 to its maximum value of 10 during the first 30 epochs, using a sigmoid-shaped function $e^{-5(1-T)^2}$, where T increases linearly from 0 to 1 during the ramp-up period. In terms of EMA decay α, we set $\alpha = 0.99$ during the ramp-up period, and $\alpha = 0.999$ for the rest of the training, following [22].

Training. We adopt the exact network structure of VoteNet [11] as the structure of our student and teacher network. We pre-train VoteNet with all the available labeled samples. We then initialize the student and teacher networks with the pre-trained weights, and train the student network on both the labeled and unlabeled data by minimizing the supervised loss as well as consistency loss. The student network is trained by an ADAM optimizer with an initial learning rate of 0.001. The learning rate is decayed by 0.1 at the 80^{th} epoch. In general, the model converges at around 100 epochs. The number of generated 3D proposals is 128.

Inference. During inference, we forward the point cloud of a scene to the student network[2] to generate the proposals. Following the same protocol as described in [11], we post-process those predicted proposals by a 3D NMS module with an 3D Intersection-over-Union(IoU) threshold of 0.25. For the evaluation metric, we adopt the widely used mean average precision (mAP). By default, mAP@0.25 (3D IoU threshold 0.25) is reported in the following experiments.

4.3. Comparison with Fully-supervised Methods

Baselines. To the best of our knowledge, there are no other 3D object detection approaches sharing the same semi-supervised setting as us. Hence, we compare our semi-supervised SESS to the state-of-the-art fully-supervised 3D object detection method, VoteNet [11], which can be considered as an upper bound of our semi-supervised method since we share the same network backbone. By drawing varying ratios of labeled data out of the

[2]Note that the teacher network can also be used to detect objects. In the experiments, we find the student and the teacher network give similar performance.

Table 1: Comparison with VoteNet on SUN RGB-D val set and ScanNetV2 val set with varying ratios of labeled data. mAP@0.25 are reported as mean±standard deviation, based on 3 runs with random sampling. And the improvement (Improv.) is computed based on the mean performances over 3 runs. Note that our SESS is initialized by the VoteNet weights pre-trained on the corresponding labeled data.

Dataset	Model	10%	20%	30%	40%	50%	70%	100%
SUNRGB-D	VoteNet [11]	34.43±1.07	41.13±0.36	47.70±0.17	50.77±0.19	52.5±0.19	56.13±0.18	57.7
	SESS	**42.87±1.01**	**47.87±0.48**	**53.17±0.63**	**54.73±0.26**	**56.37±0.22**	**58.97±0.17**	**61.1**
	Improv.(%)	24.51↑	16.39↑	11.47↑	7.80↑	7.37↑	5.06↑	5.89↑
ScanNetV2	VoteNet [11]	30.97±0.79	41.60±0.46	45.57±0.38	49.2±0.33	52.57±0.07	54.97±0.07	58.6
	SESS	**39.67±0.91**	**47.93±0.39**	**52.20±0.09**	**54.93±0.27**	**57.77±0.41**	**59.20±0.08**	**62.1**
	Improv.(%)	28.09↑	15.22↑	14.55 ↑	11.64 ↑	9.89 ↑	7.70 ↑	5.97↑

Table 2: Comparison with fully-supervised methods on SUN RGB-D and ScanNetV2 val sets with 100% training labels.

Method	SUN RGB-D	ScanNetV2
DSS [20]	42.1	15.2
COG [15]	47.6	–
2D-driven [5]	45.1	–
F-PointNet [12]	54.0	19.8
GSPN [5]	–	30.6
3D-SIS [4]	–	40.2
VoteNet [11]	57.7	58.6
SESS	**61.1**	**62.1**

entire training set, we train VoteNet with the available labeled data in a fully-supervised way, and SESS with the available labeled data as well as the remaining unlabeled data in a semi-supervised way. Additionally, we also evaluate our semi-supervised SESS based on a wide-ranging comparison with existing fully-supervised 3D object detection methods. Deep Sliding Shapes (DSS) [20] and Cloud of gradients (COG) [15] are both sliding window based methods, where DSS is a 3D extension of Faster R-CNN pipeline [14], and COG designs a 3D HoG-like feature to model the 3D geometry and appearance. 2D-driven [5] and F-PointNet [12] both depend on 2D detection in associated RGB images to reduce the search space of 3D localization. GSPN [5] and 3D-SIS [4] both target on 3D instance segmentation task but incorporate 3D object detection as an auxiliary task. Note that all the aforementioned methods use both point clouds and RGB images as inputs except VoteNet and our SESS that only require point clouds.

Results. Table 1 lists the comparison results against VoteNet under different ratios of labeled data on the two datasets, respectively. SESS significantly outperforms VoteNet under each ratio setting. The improvements verify the effectiveness of our proposed semi-supervised framework. On both datasets, as the proportion of labeled samples decreases, the performance gap between our SESS and the fully-supervised VoteNet becomes larger. Given 10% labeled data, our SESS gains around 24.51% and 28.09% improvement over VoteNet on SUN RGB-D and ScanNetV2, respectively. This indicates that our framework is able to learn knowledge from unlabeled data, and our benefit is larger when the number of labeled data is scarce.

Table 3: Transductive leaning on SUN RGB-D and ScanNetV2 unlabeled training sets, compared with fully-supervised VoteNet. The percentage indicates the ratio of labeled data for training.

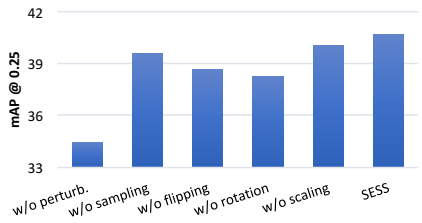
Dataset	Model	10%	20%	30%	40%	50%	70%
SUNRGB-D	VoteNet	33.5	39.8	47.5	49.7	51.6	55.2
	SESS	**40.7**	**46.1**	**53.3**	**54.3**	**55.1**	**59.0**
ScanNetV2	VoteNet	37.8	47.7	52.1	56.9	61.2	64.3
	SESS	**46.7**	**55.4**	**59.5**	**63.9**	**67.5**	**69.6**

It is interesting to see that by using only 50% labeled samples, our SESS achieves close to the upper-bound performance obtained by the fully-supervised VoteNet with 100% labeled samples on both datasets. Furthermore, it is worth pointing out that when given all the labeled training data, our SESS is able to further improve the performance beyond the upper-bound performance of VoteNet. We attribute the outperformance of SESS to its consistency regularization mechanism, where the 3D detector is trained to be robust to various perturbations, and the proposed three consistency losses that encode both geometry and semantic information guide the 3D detector towards producing more accurate predictions. This further indicates that our consistency losses are complementary to supervised loss, and our framework might be integrated with any supervised 3D object detector to enhance the detection accuracy.

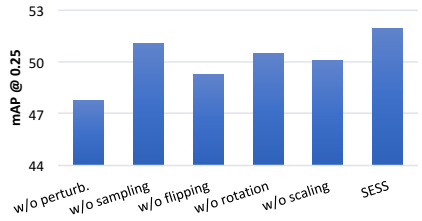
In Table 2, we further list the performance comparison between SESS and various fully-supervised methods on the two datasets, by using all the training samples.

4.4. Transductive Semi-supervised Learning

Generally, semi-supervised learning may refer to either inductive learning or transductive learning. In inductive learning, the goal is to generalize correct labels for new unseen data. In transductive learning, the goal is infer the labels restricted to the given unlabeled data. Our previous experiments conducted on unseen validation set can



(a) SUN RGB-D



(b) ScanNetV2

Figure 3: Effects of different perturbations.

be considered as inductive learning. In Table 3 we show that our SESS is also effective in transductive learning on both datasets. Our SESS consistently outperforms the fully-supervised VoteNet under different proportions of labeled samples. This demonstrates that our proposed SESS is a general framework that is not specific to inductive or transductive solution.

4.5. Ablation Studies

In this section, we explore the effects of perturbations and consistency losses. The training of ablation experiments is conducted on SUN RGB-D with 10% labeled data and ScanNetV2 with of 30% labeled data. The evaluation is on the corresponding validation set.

Perturbations. We study the effect of each perturbation by removing it from the framework, and report the performance after the removal. We also evaluate an extreme case that removes the perturbation scheme altogether. Figure 3 illustrates the resultant performances. Obviously, the performance drops greatly on both datasets when the entire perturbation scheme is removed. The effect may vary between the datasets for each individual perturbation. For example, the rotating perturbation contributes less to performance on ScanNet than SUN RGBD, as the bounding boxes of objects in ScanNet are axis-aligned. The scaling perturbation gives less improvement on SUN RGB-D than that on ScanNet. We suspect that this is because the partial scenes in SUN RGB-D are all with similar scales and thus are less sensitive to scaling perturbation. In contrast, the scales of the scenes in ScanNet are quite diverse.

Consistency Losses. We further investigate the effects of our three consistency losses by experimenting with different combinations. The comparison is reported in Table 4. From

Table 4: Ablation study on consistency losses.

center	class	size	SUN RGB-D	ScanNetV2
✓	✗	✗	38.2	50.0
✗	✓	✗	39.2	50.2
✗	✗	✓	38.1	49.2
✗	✓	✓	40.3	50.7
✓	✗	✓	38.9	50.5
✓	✓	✗	40.0	51.5
✓	✓	✓	**40.7**	**52.0**

the perspective of individual consistency loss, the center-aware and class-aware consistency losses contribute more than the size-aware consistency loss. However, the combination of center-aware or class-aware with size-aware consistency loss helps to improve the performance to some extent. Finally, the integration of the three consistency losses gives us the best performance on both datasets. It indicates that the requirement of representing the predicted bounding boxes with correct geometries (*i.e.* center, size) as well as semantics (*i.e.* class) regularizes the model towards a better performance.

4.6. Qualitative Results and Analysis

Figure 7 and Figure 8 show the visualizations of the predictions by VoteNet and SESS with 30% labeled training data and 100% labeled training data on the ScanNet and SUN RGB-D scenes, respectively. As seen in Figure 7, the partial scene obtained by single-view scanning in SUN RGB-D is very challenging, where some objects are partly visible but annotated with amodal ground-truth bounding boxes (*e.g.* the "sofa" in Figure 7). Surprisingly, both our method and the strongly supervised VoteNet successfully detect the target objects in such a challenging scene. Similar to the strongly supervised VoteNet, our SESS is able to detect more objects than those provided by the ground-truth annotations, such as the partial table in front of the sofa and the heavily occluded chairs behind the sofa. Our SESS gives more accurate predictions than VoteNet in terms of unannotated objects with 30% labeled data. We attribute this to the exploitation of unlabeled data in our proposal approach. Our SESS detects more unannotated objects when 100% labeled data is used in training, and the predicted 3D bounding boxes are consistent with human perception.

In contrast to the partial scenes in SUN RGB-D, the scenes in ScanNet are more complete and include larger areas with cluttered objects. An example is shown in Figure 8, this scene contains 7 tables and 27 chairs. Our SESS correctly recognizes the 7 tables and 26 chairs with 30% labeled data, while the strongly supervised VoteNet only detects 6 tables and 24 chairs correctly. We argue that the proposed consistency losses, which guide the model with encoded geometric and semantic information, contribute to the better localization of the 3D bounding boxes. All 34 ob-

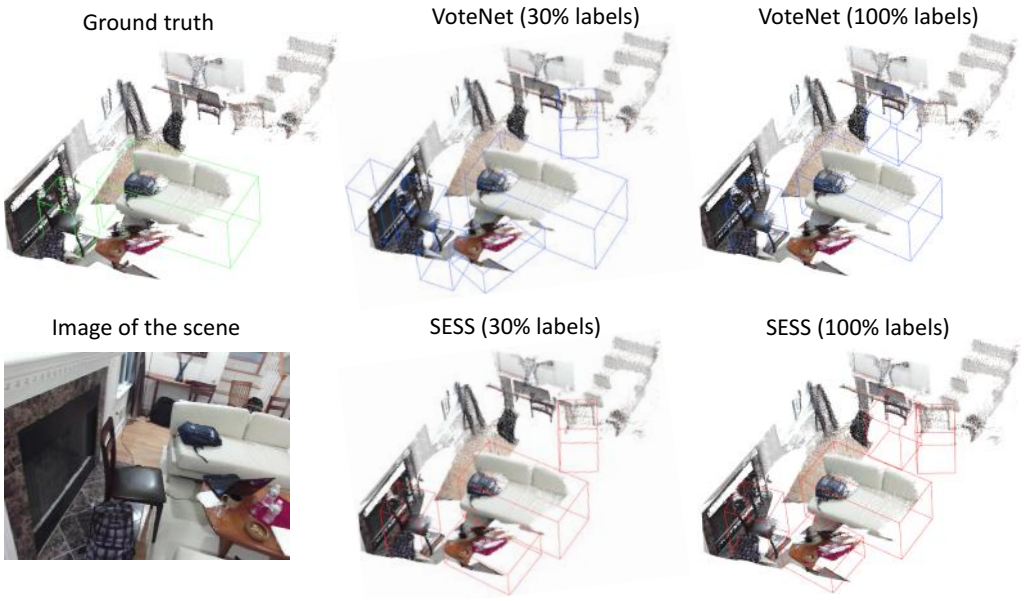


Figure 4: Qualitative comparison between the fully-supervised VoteNet and the proposed SESS on SUN RGB-D val set.

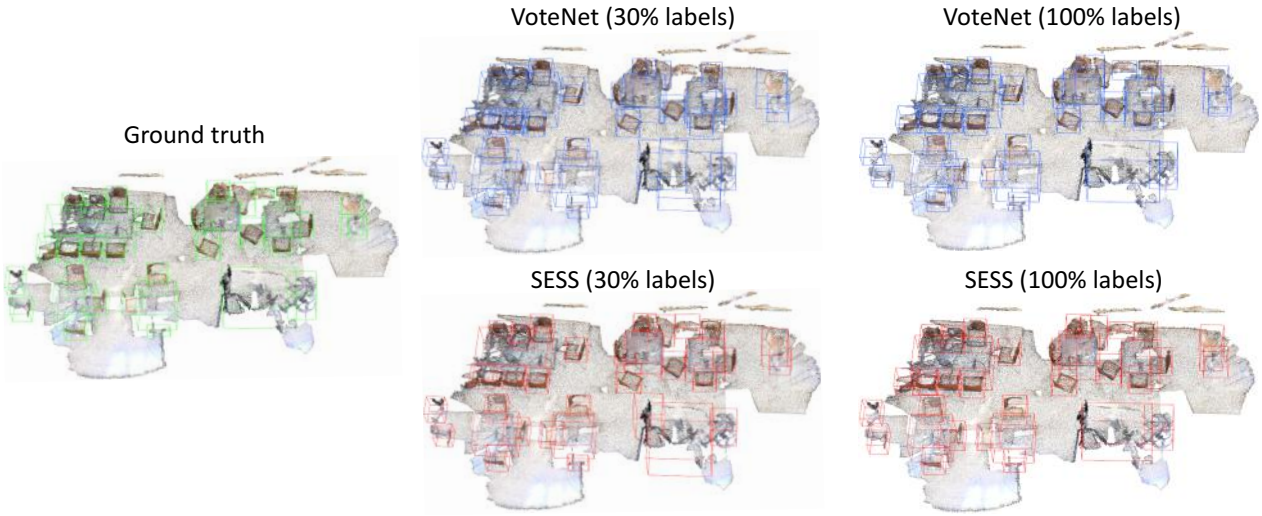


Figure 5: Qualitative comparison between the fully-supervised VoteNet and the proposed SESS on ScanNetV2 val set.

jects are completely detected with precise bounding boxes when our model is trained with 100% labeled data.

5. Conclusion

In this paper, we propose SESS, a novel self-ensembling semi-supervised point cloud-based 3D object detection framework. It does not require a large amount of strong labels that are often difficult to obtain. Our SESS follows the Mean Teacher paradigm, where we design a perturbation scheme specific to point-based data and three consistency losses that are able to force the network to generate more accuracy detections. The experimental results on two real-world datasets validate the effectiveness and advantage of our SESS. And we experimentally show that our method is a general framework that can be applied in both inductive and transductive semi-supervised 3D object detection.

Acknowledgements. This research is supported in part by the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. It is also partially supported by the Singapore MOE Tier 1 grant R-252-000-A65-114.

References

[1] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point r-cnn. In *ICCV*, pages 9775–9784, 2019.

[2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017.

[3] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[4] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *CVPR*, pages 4421–4430, 2019.

[5] Jean Lahoud and Bernard Ghanem. 2d-driven 3d object detection in rgb-d images. In *ICCV*, pages 4622–4630, 2017.

[6] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

[7] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019.

[8] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *CVPR*, pages 1019–1028, 2019.

[9] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *CVPR*, pages 7345–7353, 2019.

[10] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[11] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, 2019.

[12] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, pages 918–927, 2018.

[13] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, pages 3546–3554, 2015.

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[15] Zhile Ren and Erik B Sudderth. Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *CVPR*, pages 1525–1533, 2016.

[16] Zhile Ren and Erik B Sudderth. 3d object detection with latent support surfaces. In *CVPR*, pages 937–946, 2018.

[17] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, pages 770–779, 2019.

[18] Martin Simon, Stefan Milz, Karl Amende, and Horst-Michael Gross. Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. In *ECCV*, pages 197–209. Springer, 2018.

[19] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, pages 567–576, 2015.

[20] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *CVPR*, pages 808–816, 2016.

[21] Yew Siang Tang and Gim Hee Lee. Transferable semi-supervised 3d object detection from rgb-d data. In *ICCV*, 2019.

[22] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[23] Harri Valpola. From neural pca to deep unsupervised learning. In *Advances in Independent Component Analysis and Learning Machines*, pages 143–171. Elsevier, 2015.

[24] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In *IROS*. IEEE, 2019.

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

[26] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *CVPR*, pages 7652–7660, 2018.

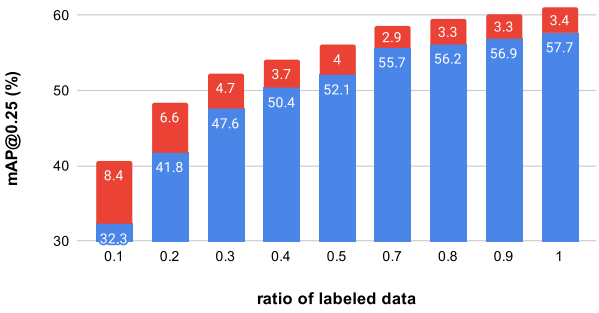
[27] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *ICCV*, 2019.

[28] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *CVPR*, pages 3947–3956, 2019.

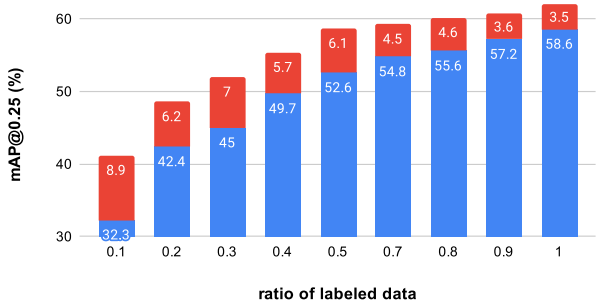
[29] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, pages 4490–4499, 2018.

In this appendix, we provide performance comparison between SESS and VoteNet with more diverse ratios of labeled data on the SUN RGB-D and ScaNetV2 val sets in Sec. A. We also provide additional evaluation metric (*i.e.* mAP@0.5 IoU) for both inductive and transductive semi-supervised learning in Sec. B. In Sec. C, we report per-class average precision on the SUN RGB-D and ScanNetV2 val set. Finally, more qualitative results are shown in Section D.

A. Additional Label ratios



(a) SUN RGB-D



(b) ScanNetV2

Figure 6: Comparison to VoteNet with more ratios of labeled data on the SUN RGB-D and ScanNetV2 val sets. The blue columns denote the performances of VoteNet, and the red columns denote the improved performance of SESS over VoteNet.

More ratios (*i.e.* 80% and 90%) of labeled data are included in the performance comparison of our SESS to the fully-supervised VoteNet on two datasets. The comparison results are illustrated in Figure 6. As can be seen from the figures, the performance margin (compared to the performance of using 100% labeled data) becomes smaller when the ratio of labeled data increases. This is because the same type of scenes (*e.g.* classrooms) share similar layout and/or objects, and thus the contribution of new labeled data to model training might be minor when similar types of samples/scenes have been seen by the model.

Table 5: Inductive leaning on SUN RGB-D and ScanNetV2 val sets compared with the fully supervised VoteNet, evaluated by mAP@0.5 IoU. The percentage indicates the ratio of labeled data for training.

Dataset	Model	10%	20%	30%	40%	50%	70%	100%
SUNRGB-D	VoteNet	10.6	14.7	23.3	25.6	27.2	30.0	31.1
	SESS	**14.4**	**20.6**	**28.5**	**29.0**	**30.6**	**33.4**	**37.3**
ScanNetV2	VoteNet	11.9	21.2	22.5	27.7	28.9	30.9	33.5
	SESS	**18.6**	**26.9**	**27.4**	**31.5**	**34.2**	**35.5**	**38.8**

Table 6: Transductive leaning on SUN RGB-D and ScanNetV2 unlabeled training sets compared with the fully supervised VoteNet, evaluated by mAP@0.5 IoU. The percentage indicates the ratio of labeled data for training.

Dataset	Model	10%	20%	30%	40%	50%	70%
SUNRGB-D	VoteNet	10.3	15.3	23.4	25.5	25.0	29.9
	SESS	**15.8**	**20.1**	**27.4**	**27.2**	**29.2**	**36.7**
ScanNetV2	VoteNet	13.8	25.3	28.6	32.7	35.2	38.3
	SESS	**23.2**	**31.3**	**34.3**	**37.6**	**41.6**	**42.6**

B. Additional Evaluation Metric

We additionally evaluate mean average precision with an IoU threshold of 0.5 on the SUN RGB-D and ScanNetV2 for both inductive (see Table 5) and transductive (see Table 6) semi-supervised 3D object detection. Consistent with the evaluation at an IoU threshold of 0.25, our SESS significantly outperforms the fully supervised VoteNet under different ratios of labeled data for both inductive and transductive learning.

C. Per-class Evaluation

We respectively report per-class average precision on 10 classes of SUN RGB-D and 18 classes of ScanNetV2 in Table 7 and 8, using all the training samples. Our SESS is superior than the fully supervised VoteNet on each class of SUN RGB-D and 14 classes of ScanNetV2 with the assistance of the proposed pertubation scheme and consistency losses.

D. More Qualitative Results and Discussions

Figure 7 and 8 demonstrate additional qualitative results on the SUN RGB-D and ScanNetV2 val datasets, respectively. As can be seen from the four examples in Figure 7, the heavy occlusion (*e.g.* the chairs at the back rows in the classroom), partial visibility (*e.g.* the leftmost cabinet in the bedroom), and extreme sparsity (*e.g.* the rightmost chair in the study space) make the detection on SUN RGB-D very difficult. Some of them are even hard for human to recognize without the reference of the associated RGB images, such as the leftmost chair in the second row in the classroom and the rightmost chair in the study space. Both VoteNet

Table 7: Per-class mAP@0.25 IoU on SUN RGB-D val set, with 100% training samples. The upper table lists the results obtained by five fully-supervised methods, and the lower table lists the results of our proposed semi-supervised method.

Method	bathtub	bed	bookshelf	chair	desk	dresser	nightstand	sofa	table	toilet	mAP
DSS	44.2	78.8	11.9	61.2	20.5	6.4	15.4	53.5	50.3	78.9	42.1
COG	58.3	63.7	31.8	62.2	**45.2**	15.5	27.4	51.0	51.3	70.1	47.6
2D-driven	43.5	64.5	31.4	48.3	27.9	25.9	41.9	50.4	37.0	80.4	45.1
F-PointNet	43.3	81.1	33.3	64.2	24.7	**32.0**	58.1	61.1	51.1	90.9	54.0
VoteNet	74.4	83.0	28.8	75.3	22.0	29.8	62.2	64.0	47.3	90.1	57.7
SESS	**76.9**	**84.8**	**35.4**	**75.8**	29.3	31.3	**66.9**	**66.4**	**51.8**	**92.3**	**61.1**

Table 8: Per-class mAP@0.25 IoU on ScanNetV2 val set, with 100% training samples. The upper table lists the results from two fully-supervised methods, and the lower table lists the results of our proposed semi-supervised method.

Method	cabin.	bed	chair	sofa	table	door	wind.	bkshf	pic.	cntr	desk	curt.	fridg.	showr.	toilet	sink	bath	ofurn.	mAP
3DSIS	19.8	69.7	66.2	71.8	36.1	30.6	10.9	27.3	0.0	10.0	46.9	14.1	53.8	36.0	87.6	43.0	84.3	16.2	40.2
VoteNet	36.3	87.9	**88.7**	89.6	58.8	47.3	38.1	44.6	7.8	**56.1**	71.7	**47.2**	45.4	57.1	94.9	**54.7**	92.1	37.2	58.6
SESS	**41.1**	**88.1**	85.9	**91.7**	**64.5**	**52.1**	**40.4**	**51.4**	**11.8**	51.9	**74.9**	45.9	**59.6**	**73.3**	**98.3**	53.9	**93.0**	**39.5**	**62.1**

and our SESS fail to detect these extremely challenging objects that come with no or few representative points. However, it is interesting to see that our SESS successfully detect most of the objects in these challenging scenarios, including those unannotated objects such as the chairs in the back of the classroom, and the table in front of the bed in the bedroom.

In Figure 8, we also show four more examples covering various scenarios on ScanNetV2 dataset. Objects with strong geometric cues (*e.g*. table, chair, bed, desk *etc*.) are easy to detect since both strongly supervised VoteNet and our SESS rely on only the geometric data (*i.e*. XYZ coordinates). In contrast, objects without explicit geometric features (*e.g*. door, picture, window) are difficult to recognize. Despite the challenge, our SESS is able to detect most of the difficult objects, such as bookshelves in the library and doors in the lounge. We argue that the proposed consistency losses, which encode not only geometric but also semantic information, guide the model to achieve better localization of the 3D bounding boxes.

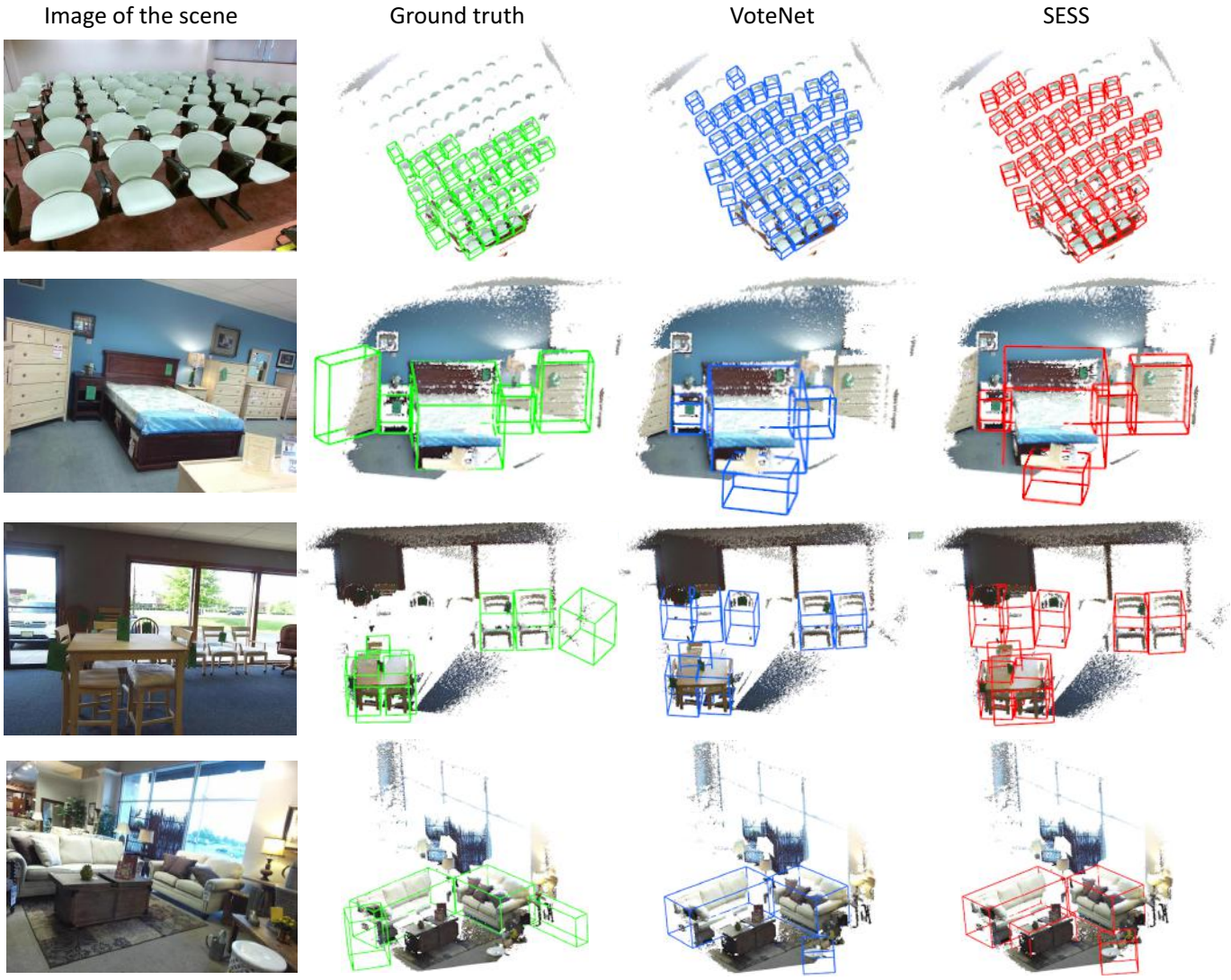


Figure 7: Additional Qualitative comparison between the fully-supervised VoteNet and the proposed SESS on SUN RGB-D *val* set, using 100% training samples. Four scene types are illustrated from the upper to bottom, they are *classroom*, *bedroom*, *study space*, and *living room*.

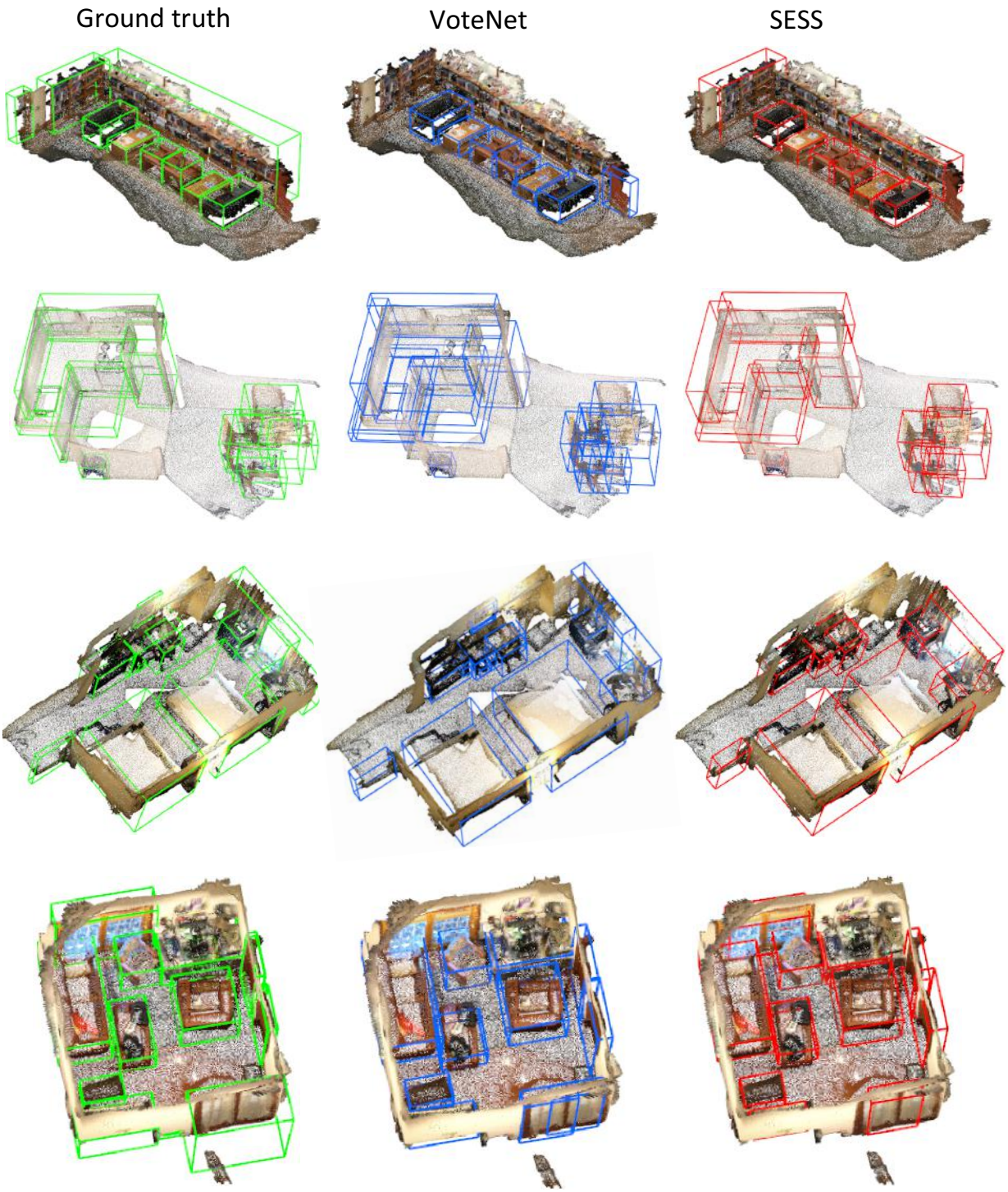


Figure 8: Additional Qualitative comparison between the fully-supervised VoteNet and the proposed SESS on ScanNetV2 *val* set, using 100% training samples. Four scene types are illustrated from the upper to bottom, they are *library*, *kitchen*, *hotel*, and *lounge*.