Static-Dynamic Co-Teaching for Class-Incremental 3D Object Detection

Na Zhao  Gim Hee Lee
Department of Computer Science, National University of Singapore
{nazhao, gimhee.lee}@comp.nus.edu.sg

Abstract
Deep learning-based approaches have shown remarkable performance in the 3D object detection task. However, they suffer from a catastrophic performance drop on the originally trained classes when incrementally learning new classes without revisiting the old data. This “catastrophic forgetting” phenomenon impedes the deployment of 3D object detection approaches in real-world scenarios, where continuous learning systems are needed. In this paper, we study the unexplored yet important class-incremental 3D object detection problem and present the first solution - SDCoT, a novel static-dynamic co-teaching method. Our SDCoT alleviates the catastrophic forgetting of old classes via a static teacher, which provides pseudo annotations for old classes in the new samples and regularizes the current model by extracting previous knowledge with a distillation loss. At the same time, SDCoT consistently learns the underlying knowledge from new data via a dynamic teacher. We conduct extensive experiments on two benchmark datasets and demonstrate the superior performance of our SDCoT over baseline approaches in several incremental learning scenarios.

Introduction
The success of deep learning are seen in many computer vision tasks that include point cloud-based 3D object detection. Many deep learning-based approaches (Li, Zhang, and Xia 2016; Chen et al. 2017; Beltrán et al. 2018; Yan, Mao, and Li 2018; Yang, Luo, and Urtasun 2018; Zeng et al. 2018; Zhou and Tuzel 2018; Chen et al. 2019; Lang et al. 2019; Qi et al. 2019; Shi, Wang, and Li 2019; Yang et al. 2019; Zhou et al. 2019; Yang et al. 2020; Zheng et al. 2021) are proposed and have shown impressive performance in localizing and categorizing objects of interest in the point cloud of a scene. However, these approaches suffer from “catastrophic forgetting”, i.e., a significant performance degradation on the old classes (c.f. Row 3 of Table 1 and 2) when applied in a class-incremental scenario where new classes are added incrementally while old data might be unavailable due to storage limitation or privacy issue. The “catastrophic forgetting” phenomenon largely limits the use of these models in real-world applications, where intelligent machines are required to continually learn new knowledge without forgetting the old one. For example, the detection system on a domestic robot is initially trained to detect several base classes such as ‘chair’ and ‘picture’ (see the left example in Figure 1). Subsequently, when the examples of novel classes such as ‘sofa’ and ‘table’ become available, the system needs to incrementally learn to detect these novel classes without losing the ability to detect the base classes (see the right example in Figure 1). Furthermore, the ability to do class-incremental learning of 3D object detection gives machines a learning capability closer to humans since we do not forget old concepts after learning new ones.

Although class-incremental learning has been studied in several computer vision tasks (Li and Hoiem 2017; Shmelkov, Schmid, and Alahari 2017; Michieli and Zanuttigh 2019; Dong et al. 2021), especially image classification, class-incremental learning of 3D object detection remains unexplored. To our best knowledge, we are the first to study this unexplored yet important problem, and to present an effective Static-Dynamic Co-Teaching solution named SDCoT. Our SDCoT is able to incrementally detect new classes without revisiting any old data or annotations of the old classes. A challenge in class-incremental learning of object detection is the high chance of old (in the background without labels) and new (with labels) classes co-occurring in the new training samples. This causes the model to wrongly suppress the old classes and thus expedites the catastrophic forgetting process. To overcome this challenge, SDCoT leverages the previous model trained on old data to generate pseudo annotations of old classes in the new training samples. Consequently, a mixture of pseudo labels of the old classes and the ground-truth labels of new classes, i.e., “mixed labels” is used to train the current model.

A naive way of pseudo label generation leads to inaccurate and incomplete pseudo labels that deteriorate the detection performance. Our SDCoT alleviates this problem by introd-
ing co-teaching from two teachers: a static teacher and a dynamic teacher. Specifically, the static teacher is a frozen copy of the previous model, which teaches to distill previously learned knowledge from old data with a distillation loss. On the other hand, the dynamic teacher is an ensemble of the current model across its up-to-date training steps, which teaches to consistently learn the underlying knowledge from the new data with a consistency loss. As a result, our SDCoT trains the current model with supervision from the “mixed labels” and regularizations from the two adversarial teachers. We conduct extensive experiments on SUN RGB-D and ScanNet datasets. The performance improvements over the baselines under different incremental learning scenarios demonstrate the effectiveness of our SDCoT in class-incremental 3D object detection. Additionally, we validate the contribution of static and dynamic teachers in knowledge exploitation by evaluating different variants of our SDCoT. Finally, we also show our SDCoT is compatible with examples from old data once they are available.

Related Work

Class-incremental learning is a classical machine learning problem, which refers to the continuous addition of new classes into a model. Most existing class-incremental learning methods focus on image classification task, which can be classified into two main categories: 1) regularization based methods minimize the discrepancy between either the data (Li and Hoiem 2017; Hou et al. 2019) or parameters (Kirkpatrick et al. 2017; Aljundi et al. 2018) of the previous model and the current model; 2) rehearsal/replay-based methods store a subset of exemplars from previous classes (Rebuffi et al. 2017; Castro et al. 2018; Wu et al. 2019) or produce synthesized exemplars for previous classes using a generative model (Shin et al. 2017; Ostapenko et al. 2019).

Recently, several works apply class-incremental learning on image-based object detection task. Most of them (Shmelkov, Schmid, and Alahari 2017; Chen, Yu, and Chen 2019) address this problem by exploring knowledge distillation on network responses (i.e. data-based regularization). For example, the first study on class-incremental image object detection (Shmelkov, Schmid, and Alahari 2017) leverages Fast R-CNN as object detector and applies distillation losses on the predictions of classification layer and bounding box regression layer. Built upon this first work, CiFRCN (Hao et al. 2019) additionally distills the intermediate features of RPN by adopting Faster R-CNN. However, these knowledge distillation methods are specifically designed for 2D object detection backbone; how to apply knowledge distillation (e.g. what to distill) on the point cloud-based 3D object detection backbone is unknown. We adapt a standard 3D object detector to class-incremental 3D object detection task, and further show the effects of different choices in employing knowledge distillation on adapted 3D object detector. More recently, IncDet (Liu et al. 2020) adapts Elastic weight consolidation (EWC) (Kirkpatrick et al. 2017), a parameter-based regularization method, to class-incremental image object detection task. IncDet circumvents the co-occurrence challenge in class-incremental object detection by using pseudo bound-box annotations of old classes in new training samples. Similar to IncDet, we also utilize pseudo annotations of old classes to prevent the current model from mistakenly classifying old class objects as background in the new samples. Nonetheless, unlike its image-based counterpart, the generated pseudo annotations in 3D scenario are not very accurate and may cause performance degradation. We solve this issue by proposing a static-dynamic co-teaching technique.

Our Methodology

Problem Definition

In the class-incremental 3D object detection task, there are two non-overlapped sets of classes: base classes set \(C_{\text{base}}\) and novel classes set \(C_{\text{novel}}\). A set of data \(D_{\text{base}}\) is available for \(C_{\text{base}}\), and another set of data \(D_{\text{novel}}\) is available for \(C_{\text{novel}}\). We define the class-incremental 3D object detection task as follows: given a well-trained 3D object detector \(\Phi_B\) (i.e. base model) on \(D_{\text{base}}\), our goal is to learn an incremental 3D object detector \(\Phi_{BUN}\) (i.e. incremental model) using only \(D_{\text{novel}}\), such that \(\Phi_{BUN}\) is able to detect objects from all the classes seen so far, i.e. \(C_{\text{base}} \cup C_{\text{novel}}\).

To this end, we propose SDCoT: a novel Static-Dynamic Co-Teaching framework to achieve class-incremental learning on 3D object detection.

Anatomy of VoteNet

We use VoteNet (Qi et al. 2019) as the prototype of our 3D object detector because of its efficiency and simplicity in point cloud-based 3D object detection. In this section, we dissect the anatomy of VoteNet to reveal two observations that we leverage to adapt VoteNet for the design of our SDCoT.

Observation 1. VoteNet inherently includes two sub-sampling steps: 1) sub-sample \(M\) seeds (denoted as \(S\) in Figure 2) from \(N\) input points via a feature learning backbone; and 2) sub-sample \(K\) votes from \(V\) as cluster centers to generate \(K\) proposals by aggregating neighboring votes. Due to the stochasticity of these sub-sampling steps in VoteNet, different sets of proposals are produced from the same input point cloud at different times.

Remark. The stochasticity of VoteNet implies that the sets of proposals generated from the base and the incremental models, respectively, are not aligned even for the same input point cloud. This impedes a direct comparison of the proposals, which is essential for training an incremental model via knowledge distillation. To circumvent this problem, we store all the indices of the sampled points and the indices of the sampled votes from the incremental model, and re-use these indices in the base model. Consequently, the two sets of proposals produced from the two models are aligned and can be compared to measure the output discrepancy.

Observation 2. After obtaining the proposal features (denoted as \(P\) in Figure 2), VoteNet adopts one multi-layer perceptron (MLP) layer to yield prediction scores for each pro-
positional. The prediction scores consist of 2 objectness scores, 3 center offsets, 2NH heading scores (NH heading bins), 4NC box size scores (NC size templates), and NC category scores. Note that the box size scores include 1 classification score and 3 size offsets for each size template, and the size templates correspond to the class categories. The size of prediction scores is fixed after VoteNet is trained.

**Remark.** The fixed prediction scores size of VoteNet after training is problematic for class-incremental learning. To enroll new classes in class-incremental learning, the weights for class-aware predictions need to be dynamically updated according to the addition of novel classes. We solve this problem by first decoupling the last MLP layer into two parts (i.e., regressor and classifier in Figure 2) to separate the category prediction from the predictions of other scores, and then adding new weights to the classifier according to the novel classes. We concurrently replace the class-aware size prediction with class-agnostic one to achieve a simpler implementation for class-incremental 3D object detection.

Our SDCoT

**Pseudo Label Generation.** A challenge in class-incremental learning of object detection is the high possibility of co-occurrence of different classes in some scenes. Concretely, there is a high probability that instances belonging to the base classes appear as background in the samples of DN. As a result, these regions that contain the old class objects are wrongly suppressed during incremental class training and thus expedite catastrophic forgetting. Moreover, the presence of base classes without annotations confuses the incremental learning model.

To overcome the co-occurrence challenge, we take a frozen copy of the base model ΦB to generate pseudo labels w.r.t. Cbase for each training sample in DN. The generation of pseudo labels from ΦB can also be considered as a way to exploit previous knowledge. More specifically, after obtaining the predicted 3D bounding boxes (bboxes) from ΦB, we filter out low-confidence bboxes by setting two thresholds with respect to the objectness score and classification probability, denoted as τo and τc. Unfortunately, the resulting pseudo labels with the hard thresholding strategy are often inaccurate and incomplete, i.e., there are missing annotations for some objects of base classes (see examples in Figure 3). Consequently, these inaccurate and incomplete labels can affect the learning of the incremental model. We alleviate the detrimental effects of these labels by a static-dynamic co-teaching strategy.

**Static-Dynamic Co-Teaching.** We design our static-dynamic co-teaching strategy based on the conjecture that the incremental model is less susceptible to noisy and incomplete labels when it is able to largely exploit the underlying knowledge from the base model and new data. Generally, the well-trained base model encodes valuable knowledge of base classes. In view of this, we adopt a frozen copy of the base model as our static teacher. Through the use of pseudo labels, we impede the catastrophic forgetting of base classes caused by the absence of base class annotations in novel training samples. To further exploit more knowledge from the base model, we introduce a distillation scheme with the aim of keeping responses from the base and incremental models to be as close as possible. Specifically, our distillation scheme targets the predicting layer and computes a distillation loss that measures the difference between the classification logits with respect to Cbase from the base and incremental models. This knowledge distillation scheme can compensate for the missing labels with respect to Cbase when the base class objects co-occur in a scene of DN. Furthermore, the responses, i.e., classification logits with respect to Cbase, can provide some useful information of the background, i.e., dark knowledge (Furlanello et al. 2018; Hinton, Vinyals, and Dean 2015), even when there is no base class object.

To exploit more information from the new data, we also design a dynamic teacher that is able to consistently learn the underlying knowledge in terms of both base and novel classes. The design of our dynamic teacher is inspired by Mean Teacher (Tarvainen and Valpola 2017), a self-ensembling technique that is originally proposed to effectively exploit unlabeled data for reducing over-fitting in semi-supervised learning. SESS (Zhao, Chua, and Lee 2020) adapts this self-ensembling technique to semi-supervised 3D object detection task by proposing a perturbation scheme and a consistency loss that enforces the consensus of locations, sizes, categories of the output proposals between a student and a teacher network. More importantly, they show that their superior performance under 100% labeled data is due to the consistency regularization of the mean-teacher paradigm, which gives their framework the capability to exploit additional underlying knowledge from the data. Thus, we incorporate the dynamic teacher, and adopt the perturbation scheme and the consistency loss of SESS in our SDCoT for a deeper knowledge exploitation of the new data. Consequently, the dynamic teacher guides the incremental model to be more robust against imperfect pseudo labels in new data and also concurrently to be more expressive on new classes.

**SDCoT Details.** The architecture of our SDCoT is illustrated in Figure 2. It consists of three networks: one student, one static teacher, and one dynamic teacher. Both the student and two teacher networks are 3D object detectors that use the modified VoteNet as backbone. Particularly, the student is the incremental detector ΦBN that incrementally learns from Cnovel. It is co-taught by the static and dynamic teachers. The static teacher is a frozen copy of the base model.
We normalize the classification logits by subtracting its mean $\Phi$ is applied on $\{Y_i\}$ from $X$. More formally, the distillation loss is computed as:

$$L_{\text{dis}} = \frac{1}{K} \sum_{i=1}^{K} || \hat{p}_{B,i}^T - \bar{p}_{B,i}^T ||_2.$$  

where $\bar{p}_{B,i}^T$ is a $|C_{\text{base}}|$-dimensional vector, which represents normalized classification logits of $i$-th 3D object proposal. On the other hand, the output proposals of the student network (i.e. $Y_{BUN}$ in Figure 4) are compared with: 1) the mixed labels $\{Y_B, Y_N\}$ transformed by the same augmentation step that is applied on $X^j$ to compute a supervised loss $L_{\text{sup}}$ similar as the multi-task loss in VoteNet; and 2) the output proposals of the dynamic teacher network $Y_{BUN}$ transformed by the same augmentation step as above to compute a consistency loss $L_{\text{con}}$ as in SESS, respectively.

At each training iteration $t$, the student network is updated by the stochastic gradient descent based on a weighted sum of the three losses:

$$L = \lambda_s L_{\text{sup}} + \lambda_d L_{\text{dis}} + \lambda_c L_{\text{con}}.$$  

After updating the student network, the dynamic teacher is updated as an exponential moving average (EMA) of the student parameters: $\Phi_t = \alpha \Phi_{t-1} + (1 - \alpha) \Phi_t$, where $\alpha$ is a hyperparameter to determine the amount of information taken from the student network. At inference time, an input point cloud is directly passed to the dynamic teacher network to predict a set of 3D bounding boxes, which are post-processed by a 3D NMS module.

**Discussion.** Interestingly, the static teacher and the dynamic teacher are opposing each other. The conservative former is preventing the student from deviating too much from the base model, while the radical latter is pushing the student to update with new knowledge. Nonetheless, an equilibrium would be reached by the knowledge distilling static teacher and the consistency regularizing dynamic teacher when the co-training converges.

---

**Experiments**

**Datasets and Settings.** We evaluate SDCoT on the SUN RGB-D 3D object detection benchmark and ScanNet dataset. **SUN RGB-D** (Song, Lichtenberg, and Xiao 2015) consists of 5,285 training samples and 5,050 validation samples for hundreds of object classes. To be consistent with the standard evaluation protocol in prior works (e.g. VoteNet), we perform evaluation on the 10 most common categories. **ScanNet** (Dai et al. 2017) consists of 1,201 training samples and 312 validation samples, where there is no amodal oriented 3D bounding boxes but point-level semantic segmentation labels. We follow VoteNet to derive the axis-aligned bounding boxes from the point-level labeling and adopt the same 18 object classes for evaluation. The differences between the two datasets are highlighted in the supplementary material.

---

The details of $L_{\text{sup}}$ are provided in the supplementary material.

1 The subscripts of $\Phi_{BUN}$ and $\Phi_{BUN}$ are omitted for brevity.
2 Both the student and dynamic teacher networks can be used for prediction during inference. We empirically found that the dynamic teacher gives better prediction results and thus use it for inference.
Table 1: *Batch incremental* 3D object detection performance (mAP@0.25) on SUN RGB-D val set. All the methods listed in the middle table incrementally learn on \(|C_{novel}|\) novel classes. Base training is with (10 – \(|C_{novel}|\)) base classes and joint training is with all 10 classes.

| Method          | \(|C_{novel}| = 0\) | \(|C_{novel}| = 5\) | \(|C_{novel}| = 1\) |
|-----------------|---------------------|---------------------|---------------------|
|                 | Base | Novel | All | Base | Novel | All | Base | Novel | All |
| 1 Base training | 57.58 | – | – | 53.73 | – | – | 55.10 | – | – |
| 2 Freeze and add | 54.24 | 10.61 | 32.42 | 51.94 | 12.64 | 40.16 | 54.63 | 0.9 | 49.26 |
| 3 Fine-tuning   | 3.48 | 54.09 | 28.79 | 4.1 | 60.17 | 20.92 | 14.86 | 1.38 | 12.51 |
| 4 SDCoT w/o \(L_{dis}\) & \(L_{con}\) | 52.17 | 50.12 | 51.14 | 38.96 | 63.68 | 46.38 | 26.83 | 24.77 | 26.63 |
| 5 SDCoT w/o \(L_{dis}\) & \(L_{con}\) | 50.35 | 59.88 | 55.12 | 37.91 | 66.39 | 46.45 | 30.85 | 29.96 | 30.76 |
| 6 SDCoT w/o \(L_{dis}\) & \(L_{con}\) | 52.92 | 57.11 | 55.01 | 41.81 | 63.45 | 48.30 | 31.61 | 25.78 | 31.02 |
| 7 SDCoT         | 53.61 | 60.80 | 57.21 | 44.48 | 67.41 | 51.36 | 36.81 | 42.69 | 37.40 |
| 8 Joint training | 58.92 | 58.80 | 58.86 | 54.80 | 68.33 | 58.86 | 55.36 | 90.36 | 58.86 |

Table 2: *Batch incremental* 3D object detection performance (mAP@0.25) on ScanNet val set. All the methods listed in the middle table incrementally learn on \(|C_{novel}|\) novel classes. Base training is with (18 – \(|C_{novel}|\)) base classes and joint training is with all 18 classes.

| Method          | \(|C_{novel}| = 0\) | \(|C_{novel}| = 5\) | \(|C_{novel}| = 1\) |
|-----------------|---------------------|---------------------|---------------------|
|                 | Base | Novel | All | Base | Novel | All | Base | Novel | All |
| 1 Base training | 60.75 | – | – | 53.14 | – | – | 56.89 | – | – |
| 2 Freeze and add | 58.85 | 4.22 | 31.53 | 49.85 | 3.15 | 39.47 | 56.24 | 0.29 | 53.14 |
| 3 Fine-tuning   | 1.91 | 52.39 | 27.15 | 1.09 | 59.44 | 14.05 | 0.25 | 12.98 | 0.96 |
| 4 SDCoT w/o \(L_{dis}\) & \(L_{con}\) | 53.09 | 46.42 | 49.76 | 48.27 | 63.87 | 51.74 | 47.91 | 27.89 | 46.80 |
| 5 SDCoT w/o \(L_{dis}\) & \(L_{con}\) | 51.21 | 53.58 | 52.39 | 48.45 | 69.82 | 53.19 | 48.60 | 30.07 | 47.57 |
| 6 SDCoT w/o \(L_{dis}\) & \(L_{con}\) | 53.51 | 51.22 | 52.26 | 48.54 | 67.52 | 52.76 | 49.31 | 30.52 | 48.26 |
| 7 SDCoT         | 53.75 | 54.91 | 54.33 | 49.50 | 70.85 | 54.25 | 52.01 | 31.71 | 50.89 |
| 8 Joint training | 58.90 | 54.13 | 56.51 | 53.16 | 68.23 | 56.51 | 57.83 | 34.16 | 56.51 |

*Setup.* To customize the datasets to the class-incremental learning setting, we take a subset of classes in alphabetical order from each dataset as \(C_{base}\) and treat the remaining as \(C_{novel}\), following the class splitting strategy in class-incremental image-based object detection (Schmelkov, Schmid, and Alahari 2017). \(D_{base}\) is composed of training samples that contain any class of \(C_{base}\) and ignores annotations for \(C_{novel}\). \(D_{novel}\) is constructed in a similar way. Note that \(D_{base}\) and \(D_{novel}\) may contain the same sample, but the annotations of this sample are different due to the change of interest on the classes.

*Evaluation metric.* We adopt the widely used metric in 3D point cloud object detection, i.e. mean average precision (mAP). By default, we report mAP under 3D IoU threshold 0.25, denoted as mAP@0.25, in the following experiments.

*Implementation Details.*

We set \(\tau_r\) and \(\tau_t\) that control the selection of pseudo labels as 0.95 and 0.9, respectively. The weights in the loss function (i.e. Eq. 2) are set as \(\lambda_{c}=10, \lambda_{r}=1, \lambda_{\ell}=10\). We adopt a ramp-up technique (Tarvainen and Valpola 2017) to schedule the respective contributions of \(\lambda_{c}\) and \(\lambda_{r}\). Specifically, \(\lambda_{c}\) and \(\lambda_{r}\) ramp up from 0 to their corresponding maximum value 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. Following SESS, we set \(\alpha\) in EMA as 0.99 during the ramp-up period and raise it to 0.999 in the following training. The base model \(\Phi_B\) and the student network \(\Phi_{B,U,N}\) are trained by an Adam optimizer. The initial learning rate for \(\Phi_B\) is set to 0.001 and then decayed by 0.1 at the 80th and 120th epoch. The initial learning rate for \(\Phi_{B,U,N}\) varies based on the settings of class-incremental learning.

*Baselines.*

We design two direct and naive baselines for class-incremental 3D object detection. The first is “freeze and add”: freeze the base model \(\Phi_B\) that is well-trained with \(D_{base}\), and then add a new classifier for \(C_{novel}\) trained on \(D_{novel}\) to the classifier branch of \(\Phi_B\). The other is “fine-tuning”: fine-tune all parameters of the base model (except the old classifier) as well as a new classifier for \(C_{novel}\) (randomly initialized) with \(D_{novel}\). In addition to the two naive baselines, we also compare our SDCoT with its three variants, i.e. without either the distillation loss (\(L_{dis}\)) or the consistency loss (\(L_{con}\)). Concretely, we remove the entire dynamic teacher when w/o \(L_{dis}\) is applied; and the static teacher is just used to generate pseudo labels when w/o \(L_{dis}\) is applied. Finally, joint training that is trained on all the classes serves as the upper-bound.

*Quantitative Results.*

We evaluate the effectiveness of SDCoT in class-incremental 3D object detection task by designing two different scenarios: 1) *batch incremental learning:* all the novel classes are available at once for \(\Phi_{B,U,N}\) to update; and 2) *sequential incremental learning:* the novel classes are split into subsets and become available sequentially. Note that the next static teacher network is updated by the current learned student network in sequential incremental learning. Furthermore, we consider different settings on the number of novel classes in batch incremental learning to eliminate the bias caused by particular classes. Specifically, we evaluate on three settings: a) \(|C_{novel}| = |C_{base}|\); b) \(|C_{novel}| < |C_{base}|\) and \(|C_{novel}| > 1\); c) \(|C_{novel}| = 1\).

*Batch incremental learning.* Table 1 and 2 show the comparison results of batch incremental 3D object detection performed under the three settings on SUN RGB-D and ScanNet, respectively. In each table, the upper part is a standard training on \(C_{base}\), the middle part lists the results when \(C_{novel}\) is incrementally added, and the bottom part is an upper-bound jointly trained on \(C_{base} \cup C_{novel}\). As can be seen from the tables, the two naive solutions (i.e. freeze and add, and fine-tuning) lead to extremely poor performance on either novel
classes or base classes in all settings on both datasets. It is apparent that the “freeze and add” solution leads to sub-optimal results on $C_{novel}$, although it can largely preserves the performance on $C_{base}$. On the other hand, “fine-tuning” the model with new object classes leads to catastrophic forgetting of old classes.

It is notable that incorporating pseudo labels into ground-truth labels (see 4th row of Table 1 and 2) can greatly help the incremental model preserve the knowledge from the previous classes. Furthermore, compared to only using mixed labels, the addition of the distillation loss (see 6th row of Table 1 and 2) gains various improvements on the base classes in different settings. This shows that the distillation loss do help exploit extra knowledge from the static teacher. We also notice that the performance with $L_{dis}$ surpasses that without $L_{dis}$ on the novel classes in most settings. The outperformance may be due to the advantage of the distillation loss in preventing background regions from confusing the incremental model. When the consistency loss is added (see 5th row of Table 1 and 2), we observe consistent and significant improvements on the novel classes on all settings. The improvements show that the dynamic teacher is very useful in learning the underlying knowledge from new data. Finally, despite the dataset and setting differences, our SDCoT combining the three losses (see 7th row of Table 1 and 2) achieves the best performance on both base and novel classes compared to its three variants. This clearly demonstrates the superiority of SDCoT in adapting to novel knowledge while maintaining the previous knowledge. It also empirically agrees with our conjecture, i.e. the deep distillation of knowledge from the new data and base model makes the model be less susceptible to noisy and incomplete pseudo labels.

It is interesting to see that in some settings, e.g. $|C_{novel}| = 5$ on SUN RGB-D and $|C_{novel}| = 9$ on ScanNet, SDCoT outperforms the upper-bound on novel classes. We attribute this outperformance to the cooperation of consistency regularization provided by the dynamic teacher and the confusion alleviation supported by the static teacher. Another interesting finding is the large performance gap between SDCoT and the upper-bound when only the “toilet” class is added (i.e. $|C_{novel}| = 1$) on SUN RGB-D. This is likely due to the “toilet” class having very few instances (c.f. Table 1 in the supplementary material) in the training set, which are insufficient for the model to learn well.

### Table 3: Per-class performance (AP@0.25) of SDCoT on SUN RGB-D val set. Setting: sequential incremental learning of 5 novel classes. B[1-5] denotes standard training on 5 base classes. B[1-10] denotes joint training on all classes.

| Setting | Base Novel All |
|---------|----------------|
| B[1-5]  | 77.97 84.17 23.83 62.83 16.94 26.04 57.34 59.75 | 75.93 |
| B[1-10] | 78.49 84.31 32.62 73.75 25.44 30.90 51.11 50.88 40.36 | 70.85 |

### Table 4: Per-class performance (AP@0.25) of SDCoT on ScanNet val set. Setting: sequential incremental learning of 4 novel classes. B[1-14] denotes standard training on 14 base classes. B[1-18] denotes joint training on all classes.

| Setting | Base Novel All |
|---------|----------------|
| B[1-14] | 52.53 60.37 56.45 |
| B[1-18] | 52.54 60.22 56.38 |

### Table 5: Effects of different distillation targets under the setting of $|C_{novel}| = 5$ on SUN RGB-D dataset.

#### Sequential incremental learning. In Table 3 and 4 we show per-class average precision (AP) of SDCoT when novel classes are added sequentially for class-incremental learning. We evaluate with two consecutive subsets of novel classes on SUN RGB-D and ScanNet, respectively. The incremental model adapts to the first subset of classes from the previous base model, which is subsequently treated as the base model and adapts to the second subset of classes. On SUN RGB-D, we achieve 44.13% mAP on all classes (see last entry of 3rd row in Table 3) after adding 5 novel classes in two consecutive batches, which is lower than 57.21% achieved by adding the 5 classes at once (see the entry at 4th column and 7th row of Table 1). Similar pattern is found on ScanNet: the performance (i.e. 40.89% mAP) after sequentially adding 4 novel classes is lower than 54.25% obtained by adding 4 classes together. According to the performance of each individual base class in Table 3 and 4, we find that the classes which undergoes severe performance degradation during sequential incremental learning usually have relatively poor detection ability at the beginning stage, i.e. base training. Despite the performance drop of sequential incremental learning compared to batch incremental learning, it does not cause a severe catastrophic forgetting like fine-tuning.

#### Design Choices of Distillation Loss

We investigate the effects of various designs of the distillation loss. More specifically, we study different distillation targets (i.e. classification logit, bounding box regression values including center and size) and alternative loss functions (i.e. cross-entropy and knowledge distillation losses).

#### What to distill? Table 5 summarizes the effects of using different distilled targets when computing the final distillation loss. Note that we compute the mean square error between the corresponding outputs from $\Phi_B$ and $\Phi_{B\cup N}$ for size- and center-aware distillation losses, in addition to our original class-aware distillation loss. As can be seen from the table, the size- and center-aware distillation are unable to extract...
more useful information from the previous knowledge. In fact, they slightly harm the performance on the base classes in the given setting. Consequently, we only distill knowledge from the classification logits.

**How to distill?** To evaluate the effects of different loss functions, we replace the L2 norm loss in Eq. 1 with cross-entropy loss and knowledge distillation loss (Hinton, Vinyals and Dean 2015) that is a cross-entropy loss with temperature, respectively. Figure 5 shows that the L2 norm loss is a better choice for class-incorporative 3D object detection.

**Qualitative Results**

Figure 6 and 7 show the qualitative results of our SDCoT on SUN RGB-D and ScanNet, respectively. Despite the very challenging (e.g. partially visible objects and cluttered scenes) and diverse (e.g. bedroom, bathroom, and conference room) scenes, our SDCoT is able to nicely detect the novel classes as well as greatly retain the detection capacity on the base classes in all these scenes. In addition, we provide some failure examples in the supplementary material.

**Compatibility with Replayed Exemplars**

In the class-incremental learning of image classification task, it is common to store a small set of samples from old data (i.e. exemplars) to prevent catastrophic forgetting. However, the amount of its contribution in the class-incremental 3D object detection task is unclear. We adopt the simplest but effective strategy, i.e. random sampling (Chaudhry et al. 2018), to select exemplars from old data.

Interestingly, our SDCoT can easily incorporate these exemplars into the “mixed labels” as labeled instances without any change to the framework. To demonstrate the effects of different number of replayed exemplars in class-incremental 3D object detection, we sample different ratios of old data and compare the results with the baseline method (i.e. fine-tuning) on the two datasets, as shown in Figure 8. As can be seen, when more replayed exemplars are added, fine-tuning baseline achieves significant improvements on base classes while our SDCoT only gets very slight improvements. This indicates that our method is capable of persevering old knowledge, which makes it less sensitive to the addition of replayed exemplars. Furthermore, it can be seen that our SDCoT consistently outperforms fine-tuning over all percentages of replaying (c.f. the supplementary material for the numerical comparisons).

**Conclusion**

This paper studies the new and practical problem of class-incremental 3D object detection. To this end, we proposed SDCoT: an effective static-dynamic co-teaching method to incrementally detect novel classes without revisiting any previous training sample. Our SDCoT greatly addresses the catastrophic forgetting issue and further helps the model adapt to the novel classes. We demonstrated the effectiveness of SDCoT over a variety of class-incremental 3D object detection scenarios on SUN RGB-D and ScanNet datasets. We hope that our study serves as a motivation for future works on this practical problem.
Acknowledgement
This research is supported in part by the National Research Foundation, Singapore under its AI Singapore Program (AISG Award No: AISG2-RP-2020-016) and the Tier 2 grant MOET2EP20120-0011 from the Singapore Ministry of Education.

Supplementary Material
In this appendix, we provide some observations on benchmark datasets, more details on our backbone such as architecture and loss function, per-class evaluation under batch incremental learning, some failure cases, and numerical comparison results with replayed exemplars.

Observations on Benchmark Datasets
It is worth to highlight the two differences between the two datasets. First, the average number of object instances in one scene of SUN RGB-D dataset is much lower than that of ScanNet dataset, since SUN RGB-D collects a scene by single-view scanning, while ScanNet reconstructs a complete scene from RGB-D video. This difference can be observed in our qualitative examples (cf. Figure 6 and 7 in the main paper), where the scenes in ScanNet are more cluttered. Second, the ground-truth (GT) bounding boxes in SUN RGB-D are oriented and complete despite most objects are partially visible (see example in Figure 3 left in the main paper). In contrast, the GT bounding boxes in ScanNet are axis-aligned and fit to the visible parts.

Table 6 and 7 list the class names in alphabetical order and the statistics of per-class instance number as well as scan number (i.e. the number of scans containing the corresponding class) on training set of SUN RGB-D and ScanNet, respectively. As can be see from the two tables, the two datasets are both highly unbalanced across classes. This unbalanced data can cause the insufficient training problem when the added novel class has very few samples, e.g. the addition of ‘toilet’ class with only 174 training samples does not perform well under batch incremental 3D object detection setting (see the results under \(|C_{\text{novel}}| = 1\) setting in Table 1 of the main paper).

More Details on Our Backbone
Architecture Details. Our backbone, i.e. modified VoteNet, is comprised of four modules (as shown in Figure 2 of the main paper): 1) feature learning backbone that sub-samples \(M\) points (i.e. seeds) from \(N\) input points with enriched features via PointNet++ (Qi et al. 2017); 2) voting module that generates expected centers of objects (i.e. votes) from seeds via several MLP layers; 3) proposal generator that samples \(K\) votes and groups a set of neighboring votes for each sampled votes to form vote clusters, which are subsequently passed to a light PointNet network to generate \(K\) object proposals with aggregated features (i.e. proposal features); 4) predicting module including a regressor and a classifier, which takes proposal features as input and yields prediction scores of box parameters (i.e. 2 objectness scores, 3 center offsets, 3 size offsets, and \(2NH\) heading scores) and box category \((NC\) category scores), respectively. In SUN RGB-D, \(NH = 12\); in ScanNet, \(NH = 1\). \(NC\) is dynamically updated according to the setting of class-incremental learning, i.e. how many novel classes to be added.

The architectures of the feature learning backbone and the voting modules are the same as the original VoteNet (Qi et al. 2019). The proposal generator module is a set abstraction (SA) layer followed by two MLP layers with output sizes of 128, 128, which is same as the proposal module in VoteNet but without the last MLP layer. The regressor in predicting module is a Linear layer with output size of \((2 + 3 + 3 + 2NH)\), while the classifier is a Linear layer without a bias.

Supervised Loss Function. Similar to VoteNet, the supervised loss \(L_{\text{sup}}\) is a weighted sum of a voting loss, an objectiveness loss, a 3D bounding box estimation loss and a semantic classification loss:

\[
L_{\text{sup}} = L_{\text{vote-reg}} + \lambda_1 L_{\text{obj-cls}} + \lambda_2 L_{\text{box}} + \lambda_3 L_{\text{sem-cls}}.
\]  

We set \(\lambda_1 = 0.5, \lambda_2 = 1, \) and \(\lambda_3 = 0.2\). The vote regression loss \(L_{\text{vote-reg}}\), objectness loss \(L_{\text{obj-cls}}\), and semantic classification loss \(L_{\text{sem-cls}}\) are the same as defined in VoteNet. Following VoteNet, the box loss \(L_{\text{box}}\) is computed as:

\[
L_{\text{box}} = L_{\text{center-reg}} + 0.1 L_{\text{angle-clss}} + L_{\text{angle-reg}} + L_{\text{size-reg}},
\]

with the exception that the size classification loss is removed, as we replace the class-aware size prediction with class-agnostic one. We refer the readers to VoteNet for more details on the computation of each loss term.

Per-class Evaluation
In this section, we report per-class performance (Average Precision with 0.25 box IoU) comparison results of batch incremental 3D object detection performed under the three settings on SUN RGB-D and ScanNet, respectively.

SUN RGB-D. Table 8 and 9 list the per-class comparison results of batch incremental 3D object detection when \(|C_{\text{novel}}| = 5\), \(|C_{\text{novel}}| = 3\) and \(|C_{\text{novel}}| = 1\) respectively.

ScanNet. Table 11, 12 and 13 list the per-class comparison results of batch incremental 3D object detection when \(|C_{\text{novel}}| = 9\), \(|C_{\text{novel}}| = 4\) and \(|C_{\text{novel}}| = 1\) respectively.

From the per-class performance under different batch incremental settings on both datasets, we can see that our SDCoT is able to achieve optimal results on novel classes compared to the “Freeze and add” baseline, and largely prevent the model from catastrophic forgetting of base classes compared to the “Fine-tuning” baseline. Furthermore, compared to its three variants, our SDCoT achieves the best performance in most classes and show comparable performance in the remaining classes.

Failure Cases
In Figure 9, we show three failure examples on SUN RGB-D dataset. As can be seen from the first example (1st row in Figure 9), our SDCoT fails to detect the bathtub (i.e. base class) that is extremely truncated. Nonetheless, it is interesting to
see that our SDCoT is able to detect the chair (i.e., base class) in the second example (2\textsuperscript{nd} row in Figure 9) despite only a small portion of it is visible and ground-truth annotation is not available. We suspect the difference in the detection capacity between the two classes (i.e., ‘bathtub’ and ‘chair’) is caused by the imbalanced number of training samples. The ‘chair’ class has much more training examples than the ‘bathtub’ class (compare 4\textsuperscript{th} column with 1\textsuperscript{st} column in Table 6). In the second example, our SDCoT mistakenly detects the desk (i.e., base class) as table (i.e., novel class). This is likely due to the very similar geometric structures of these two classes; and the learned model is prone to recognize such geometric structure as the class that is recently seen and with more training samples (i.e., ‘table’). In the last example, we show an extremely challenging scenario, where our SDCoT fails to detect the sofa that is totally invisible.

We also show three failure examples on ScanNet dataset in Figure 10. We use these examples to illustrate several common reasons of failure in our SDCoT: 1) fail to detect thin objects such as door and picture (see the missed doors in the bottom right of the third example); 2) fail to detect partly visible objects (see the missed bookshelf in the upper right and missed chair in the middle right of the second example); 3) hard to detect the entire objects with larger shapes (see the sofa in the lower part of the first example and the bookshelf in the upper part of the second example); and 4) wrongly detect background areas as objects (see the red bounding box on the right wall in the third example, where our SDCoT mistakenly detects the wall part as a door).

Numerical Results with Replayed Exemplars

In Table 14 and 15 we list the numerical results of comparisons between fine-tuning baseline and our SDCoT on the two datasets with varying ratios of replayed exemplars, for a clear illustration. We also list the exact number of the selected replayed exemplars under different ratios on both datasets.

References

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory aware synapses: Learning what (not) to forget. In Proceedings of the European Conference on Computer Vision (ECCV), 139–154.

Beltrán, J.; Guindel, C.; Moreno, F. M.; Cruzado, D.; García, F.; and De La Escalera, A. 2018. Birdnet: a 3d object detection framework from lidar information. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 3517–3523. IEEE.

Castro, F. M.; Marín-Jiménez, M. J.; Guil, N.; Schmid, C.; and Alahari, K. 2018. End-to-end incremental learning. In Proceedings of the European conference on computer vision (ECCV), 233–248.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In Proceedings of the European Conference on Computer Vision (ECCV), 532–547.

Chen, L.; Yu, C.; and Chen, L. 2019. A new knowledge distillation for incremental object detection. In 2019 International Joint Conference on Neural Networks (IJCNN), 1–7. IEEE.

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

Chen, Y.; Liu, S.; Shen, X.; and Jia, J. 2019. Fast Point R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, 9775–9784.


| Method                | bathtub | bed  | bookshelf | chair | desk  | dresser | nightstand | sofa  | table | toilet | mAP  |
|-----------------------|---------|------|-----------|-------|-------|---------|------------|-------|-------|--------|------|
| 1 Base training       | 73.97   | 84.71| 30.19     | 75.09 | 23.93 | 57.58   |             |       |       |        |      |
| 2 Freeze and add      | 72.86   | 74.53| 30.32     | 70.29 | 23.38 | 0.36    | 0.09        | 10.25 | 12.04 | 30.11  | 32.42|
| 3 Fine-tuning         | 0.02    | 2.64 | 0.05      | 1.95  | 12.36 | 21.37   | 58.16       | 57.16 | 47.87 | 85.95  | 28.79|
| 4 SDCoT w/o Ldis & Lcon | 68.63   | 83.30| 23.56     | 68.63 | 16.72 | 20.43   | 40.25       | 57.17 | 47.33 | 85.43  | 51.14|
| 5 SDCoT w/o Ldis      | 69.35   | 80.07| 16.66     | 69.51 | 16.23 | 34.28   | 61.42       | 62.32 | 50.94 | 90.45  | 55.12|
| 6 SDCoT w/o Lcon      | 73.09   | 81.29| 23.78     | 68.57 | 11.54 | 33.79   | 53.80       | 64.06 | 48.12 | 88.78  | 55.01|
| 7 SDGT                | 75.41   | 82.15| 22.13     | 70.58 | 17.81 | 35.56   | 61.80       | 62.98 | 52.99 | 90.67  | 57.21|
| 8 Joint training      | 73.40   | 84.51| 32.62     | 73.73 | 25.44 | 30.90   | 58.11       | 64.15 | 50.48 | 90.36  | 58.86|

Table 8: Per-class performance (AP@0.25) comparison on SUN RGB-D val set. Setting: batch incremental learning of 5 novel classes.

| Method                  | bathtub | bed  | bookshelf | chair | desk  | dresser | nightstand | sofa  | table | toilet | mAP  |
|-------------------------|---------|------|-----------|-------|-------|---------|------------|-------|-------|--------|------|
| 1 Base training         | 72.85   | 83.55| 30.83     | 75.14 | 26.56 | 27.87   | 59.36       |       |       |        |      |
| 2 Freeze and add        | 74.76   | 76.28| 31.02     | 72.60 | 24.04 | 27.93   | 56.89       | 11.95 | 13.43 | 12.56  | 40.16|
| 3 Fine-tuning            | 0.52    | 16.47| 0.34      | 2.54  | 7.99  | 0.40    | 0.61        | 56.16 | 48.76 | 75.61  | 20.92|
| 4 SDCoT w/o Ldis & Lcon | 71.46   | 77.28| 16.21     | 67.32 | 11.59 | 0.24    | 28.64       | 61.18 | 46.36 | 83.40  | 46.38|
| 5 SDCoT w/o Ldis        | 74.70   | 73.28| 12.11     | 63.97 | 12.67 | 0.31    | 26.30       | 60.15 | 49.39 | 89.05  | 46.45|
| 6 SDCoT w/o Lcon        | 71.28   | 77.36| 20.17     | 67.60 | 13.87 | 7.32    | 35.08       | 61.65 | 47.28 | 81.43  | 48.30|
| 7 SDGT                  | 70.61   | 78.10| 18.82     | 68.09 | 18.28 | 13.43   | 44.00       | 62.99 | 50.29 | 88.94  | 51.56|
| 8 Joint training        | 78.49   | 84.31| 32.62     | 73.73 | 25.44 | 30.90   | 58.11       | 64.15 | 50.48 | 90.36  | 58.86|

Table 9: Per-class performance (AP@0.25) comparison on SUN RGB-D val set. Setting: batch incremental learning of 3 novel classes.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5828–5839.

Dong, J.; Cong, Y.; Sun, G.; Ma, B.; and Wang, L. 2021. 3DOL: Incremental 3D Object Learning without Catastrophic Forgetting. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 6066–6074.

Furlanello, T.; Lipton, Z.; Tschannen, M.; Itti, L.; and Anandkumar, A. 2018. Born again neural networks. In International Conference on Machine Learning, 1607–1616. PMLR.

Hao, Y.; Fu, Y.; Jiang, Y.-G.; and Tian, Q. 2019. An end-to-end architecture for class-incremental object detection with knowledge distillation. In 2019 IEEE International Conference on Multimedia and Expo (ICME), 1–6. IEEE.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a unified classifier incrementally via rebalancing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 831–839.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences, 114(13): 3521–3526.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. PointPillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 12697–12705.

Li, B.; Zhang, T.; and Xia, T. 2016. Vehicle detection from 3D lidar using fully convolutional network.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence, 40(12): 2935–2947.

Liu, L.; Kuang, Z.; Chen, Y.; Xue, J.-H.; Yang, W.; and Zhang, W. 2020. Incdet: in defense of elastic weight consolidation for incremental object detection. IEEE transactions on neural networks and learning systems.

Michieli, U.; and Zanuttigh, P. 2019. Incremental learning techniques for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision Workshops, 0–0.

Ostapenko, O.; Puscas, M.; Klein, T.; Jahnichen, P.; and Nabi, M. 2019. Learning to remember: A synaptic plasticity driven framework for continual learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 11321–11329.

Peng, C.; Zhao, K.; and Lovell, B. C. 2020. Faster ILOD: Incremental learning for object detectors based on faster R-CNN. Pattern Recognition Letters, 140: 109–115.

Qi, C. R.; Litany, O.; He, K.; and Guibas, L. J. 2019. Deep Hough Voting for 3D Object Detection in Point Clouds. In Proceedings of the IEEE International Conference on Computer Vision.

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

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2001–2010.

Shi, S.; Wang, X.; and Li, H. 2019. Pointrcnn: 3d object proposal generation and detection from point cloud. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–779.
| Method               | bath  | bed   | booksh | chair  | desk  | dresser | nightstand | sofa   | table  | toilet | mAP   |
|---------------------|-------|-------|--------|--------|-------|----------|------------|--------|--------|--------|-------|
| 1 Base training     | 71.77 | 85.01 | 33.19  | 73.76  | 25.73 | 29.41    | 61.31      | 64.79  | 50.91  |         | 55.10 |
| 2 Freeze and add    | 67.33 | 84.09 | 32.83  | 70.36  | 23.11 | 29.59    | 63.06      | 65.08  | 50.43  | 0.90   | 49.26 |
| 3 Fine-tuning        | 2.54  | 17.72 | 0.82   | 53.81  | 5.99  | 1.46     | 11.21      | 17.86  | 22.28  | 1.38   | 13.51 |
| 4 SDCoT w/o Ldis & Lcon | 70.38 | 28.06 | 5.60   | 58.11  | 10.22 | 2.42     | 3.72       | 30.25  | 32.72  | 5.99   | 26.63 |
| 5 SDCoT w/o Ldis     | 67.32 | 54.57 | 7.28   | 59.28  | 7.31  | 1.73     | 7.05       | 47.09  | 26.03  | 29.96  | 30.76 |
| 6 SDCoT w/o Lcon     | 71.82 | 26.03 | 9.63   | 60.64  | 11.93 | 8.58     | 28.95      | 30.39  | 36.47  | 25.76  | 31.02 |
| 7 SDCoT              | 72.63 | 52.04 | 10.92  | 64.55  | 14.99 | 4.65     | 26.86      | 45.58  | 37.71  | 42.69  | 37.40 |
| 8 Joint training     | 78.90 | 84.31 | 32.62  | 73.73  | 25.44 | 30.90    | 58.11      | 64.15  | 50.36  | 58.86  |        |

Table 10: Per-class performance (AP@0.25) comparison on SUN RGB-D val set. Setting: batch incremental learning of 1 novel classes.

| Method               | bath  | bed   | booksh | chair  | desk  | dresser | nightstand | sofa   | table  | toilet | mAP   |
|---------------------|-------|-------|--------|--------|-------|----------|------------|--------|--------|--------|-------|
| 1 Base training     | 77.91 | 85.15 | 47.95  | 40.36  | 88.17 | 57.80    | 35.69      | 69.38  | 44.34  |       | 60.75 |
| 2 Freeze and add    | 74.76 | 85.83 | 43.95  | 42.41  | 87.04 | 49.38    | 56.27      | 66.62  | 44.06  | 1.01  | 4.62  | 1.21 | 9.28  | 1.75  | 2.31  | 51.51 |
| 3 Fine-tuning        | 0.01  | 2.06  | 0.94   | 2.45   | 1.26  | 0.13     | 1.11       | 8.05   | 0.51   | 35.24 | 6.02  | 56.06 | 58.27 | 65.31 | 86.04 | 58.36 | 95.01 | 30.44 | 27.15 |
| 4 SDCoT w/o Ldis & Lcon | 54.15 | 85.50 | 39.95  | 27.29  | 86.22 | 33.53    | 36.03      | 64.91  | 29.33  | 40.69 | 9.41  | 45.59 | 61.22 | 48.48 | 85.65 | 64.02 | 96.66 | 30.45 | 52.39 |
| 5 SDCoT w/o Ldis     | 54.15 | 85.50 | 39.95  | 27.29  | 86.22 | 33.53    | 36.03      | 64.91  | 29.33  | 40.69 | 9.41  | 45.59 | 61.22 | 48.48 | 85.65 | 64.02 | 96.66 | 30.45 | 52.39 |
| 6 SDCoT w/o Lcon     | 53.40 | 82.77 | 42.79  | 29.90  | 84.67 | 45.50    | 38.05      | 62.45  | 38.28  | 39.93 | 5.31  | 32.46 | 37.84 | 47.92 | 87.14 | 57.05 | 92.57 | 31.68 | 52.26 |
| 7 SDCoT              | 57.10 | 82.61 | 42.59  | 30.44  | 86.16 | 45.77    | 31.16      | 65.26  | 36.67  | 42.14 | 6.12  | 51.48 | 61.06 | 35.11 | 89.07 | 58.30 | 29.87 | 54.33 |
| 8 Joint training     | 70.85 | 85.12 | 46.70  | 37.37  | 85.79 | 54.15    | 40.83      | 66.08  | 43.17  | 41.37 | 5.84  | 50.55 | 58.62 | 57.85 | 85.22 | 55.05 | 98.50 | 34.16 | 56.51 |

Table 11: Per-class performance (AP@0.25) comparison on ScanNet val set. Setting: batch incremental learning of 9 novel classes.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual learning with deep generative replay. In Advances in Neural Information Processing Systems, 2990–2999.
Shmelkov, K.; Schmid, C.; and Alahari, K. 2017. Incremental learning of object detectors without catastrophic forgetting. In Proceedings of the IEEE International Conference on Computer Vision, 3400–3409.
Song, S.; Lichtenberg, S. P.; and Xiao, J. 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In Proceedings of the IEEE conference on computer vision and pattern recognition, 567–576.
Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Advances in neural information processing systems, 1195–1204.
Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; and Fu, Y. 2019. Large scale incremental learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 374–382.
Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. Sensors, 18(10): 3337.
Yang, B.; Luo, W.; and Urtasun, R. 2018. Pixor: Real-time 3d object detection from point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7652–7660.
Yang, Z.; Sun, Y.; Liu, S.; and Jia, J. 2020. 3dssd: Point-based 3d single stage object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11040–11048.
Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; and Jia, J. 2019. STD: Sparse-to-Dense 3D Object Detector for Point Cloud. In Proceedings of the IEEE International Conference on Computer Vision.
Zeng, Y.; Hu, Y.; Liu, S.; Ye, J.; Han, Y.; Li, X.; and Sun, N. 2018. R3d: Real-time 3d vehicle detection in lidar point cloud for autonomous driving. IEEE Robotics and Automation Letters, 3(4): 3434–3440.
Zhao, N.; Chua, T.-S.; and Lee, G. H. 2020. SESS: Self-Ensembling Semi-Supervised 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11079–11087.
Zheng, W.; Tang, W.; Chen, S.; Jiang, L.; and Fu, C.-W. 2021. CIA-SSD: Confident IoU-Aware Single-Stage Object Detector From Point Cloud. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 3555–3562.
Zhou, J.; Tan, X.; Shao, Z.; and Ma, L. 2019. FVNet: 3D Front-View Proposal Generation for Real-Time Object Detection from Point Clouds. In 2019 12th International Congress on Image and Signal Processing, Biomedical Engineering and Informatics (CISP-BMEI), 1–8. IEEE.
Zhou, Y.; and Tuzel, O. 2018. Voxelsnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4490–4499.
Table 12: Per-class performance (AP@0.25) comparison on ScanNet val set. Setting: batch incremental learning of 4 novel classes.

Table 13: Per-class performance (AP@0.25) comparison on ScanNet val set. Setting: batch incremental learning of 1 novel classes.

Figure 10: Failure cases from ScanNet val set. Green and Blue bboxes are GT annotations w.r.t $C_{base}$ and $C_{novel}$, respectively. We show three examples from left to right.

Table 14: Comparison with fine-tuning baseline on SUN RGB-D val dataset with varying ratios of old data. Setting: batch incremental 3D object detection of 3 novel classes (i.e. $|C_{novel}| = 3$). Note that the number associated with the percentage indicates the number of replayed exemplars.

Table 15: Comparison with fine-tuning baseline on ScanNet val dataset with varying ratios of old data. Setting: batch incremental 3D object detection of 4 novel classes (i.e. $|C_{novel}| = 4$). Note that the number associated with the percentage indicates the number of replayed exemplars.