SDCoT++: Improved Static-Dynamic Co-Teaching for Class-Incremental 3D Object Detection

Na Zhao Peisheng Qian Fang Wu Xun Xu Xulei Yang Gim Hee Lee†

***Abstract*—Deep learning approaches have demonstrated high effectiveness in 3D object detection tasks. However, they often suffer from a notable drop in performance on the previously trained classes when learning new classes incrementally without revisiting the old data. This is the "catastrophic forgetting" phenomenon which impedes 3D object detection in real-world scenarios, where intelligent machines must continuously learn to detect previously unseen categories. Furthermore, frequent co-occurrences of old and new classes in scenes exacerbate catastrophic forgetting and cause model confusion. To address these challenges, we propose a novel static-dynamic co-teaching approach. Our framework involves a student model and two teacher models: a static teacher with fixed weights which imparts preserved old knowledge to the student, and a dynamic teacher with continuously updated weights which transfers underlying knowledge from new data to the student. To mitigate the issue of co-occurrence, we generate pseudo labels for base (*i.e.* old) classes from both static and dynamic sources during incremental learning. Additionally, to mitigate the negative impact of varying occurrence frequencies of classes on fixed thresholding during the selection of pseudo labels, we calibrate the probabilities of base classes to attain more balanced class probabilities. Moreover, our static-dynamic co-teaching framework is backbone-agnostic, making it compatible with different detection architectures. We demonstrate its backbone-agnostic nature by adapting three representative 3D object detectors: VoteNet, 3DETR and CAGroup3D. Extensive experiments showcase the superior performance of our proposed method compared to baseline approaches across indoor and outdoor benchmark datasets and applicability with different backbone models.**

***Index Terms*—Class incremental learning, object detection, 3D point clouds.**

I. Introduction

Deep learning has reached significant achievements in various computer vision tasks, notably in 3D point cloud-based object detection. Numerous deep learning-based approaches [1]–[10] have demonstrated remarkable effectiveness in localizing and classifying objects in the point cloud of a scene. These approaches typically follow a *static learning* process, where labeled data for all classes are available in a single training session. However, such static process is impractical for real-world applications in our dynamic physical world. Continuously emerging new concepts and data make it challenging to retrain models with both new and old data due to constraints such as computation costs, privacy concerns, and storage issues associated with old data. Consequently, it is imperative to develop the competency of incrementally learning new knowledge over time while retaining existing knowledge.

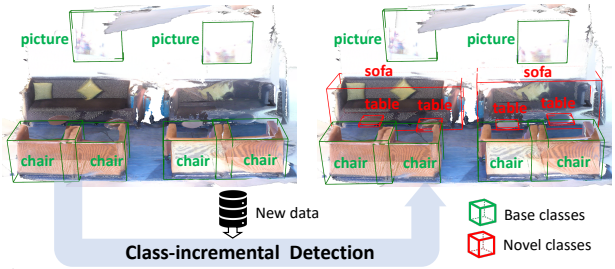

Fig. 1: An illustration of class-incremental 3D object detection.

For example, as illustrated in Figure 1, a domestic robot is initially taught to recognize only several base classes such as 'chair', 'picture', *etc*. Subsequently, the robot is expected to detect novel classes of objects 'sofa', 'table' and other new items added to the home when the training data of the novel classes are given incrementally. A naive retraining of the robot's detection model on this ever-growing dataset that includes previously learned classes incurs increasing computational complexity. Moreover, it may not always be possible to access the training data for previously seen classes due to storage limitation, or privacy and licensing restrictions. A direct training of the robot's detection model with training data of only the new classes inevitably leads to drastic drop in performance on the previously seen classes. Consequently, the robot must incrementally learn to recognize new classes of objects while preserving its detection capability on old classes without the need to access previously seen training data. This requirement of **class-incremental learning** for 3D object detection gives intelligent machines such as the domestic robot a "close-to-human brain" ability to retain previously acquired knowledge while assimilating new information.

While class-incremental learning has been explored in various computer vision tasks, such as image classification [11]–[16] and object detection [17]–[22], it still remains relatively under-explored for the task of 3D object detection. We thus investigate class-incremental 3D object detection in this paper. Particularly, we tackle a challenging scenario where old data is not available during incremental learning, potentially due to storage limitations or privacy concerns. The primary challenge in class-incremental learning is known as "*catastrophic for-*

Na Zhao is with Information Systems Technology and Design, Singapore University of Technology and Design (e-mail: na_zhao@sutd.edu.sg).

Peisheng Qian is with Information Systems Technology and Design, Singapore University of Technology and Design, and the Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore (e-mail: qian_peisheng@i2r.a-star.edu.sg).

Fang Wu is with College of Design and Engineering, National University of Singapore (e-mail: fang_wu@u.nus.edu).

Xun Xu and Xulei Yang are with the Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore (e-mail: xu_xun, yang_xulei@i2r.a-star.edu.sg).

Gim Hee Lee is with School of Computing, National University of Singapore (e-mail: gimhee.lee@comp.nus.edu.sg). † indicates corresponding author.

getting", which refers to a significant performance decline on old classes when new classes are added incrementally. This phenomenon poses a substantial challenge in class-incremental 3D object detection, particularly when old data is unavailable during incremental learning. Moreover, the frequent co-occurrences of both old and new classes in 3D detection datasets exacerbate the forgetting problem. This is because the inclusion of old, unlabeled classes and new, labeled classes in the new training samples can inadvertently suppress the detection of old classes while biasing towards the new classes, as illustrated in Rows 3 and 12 of Table I and II.

Potential solutions to address the forgetting issue in the complete absence of old data are to leverage the existing model trained on the old data to perform knowledge distillation or to create pseudo annotations for previously learned classes in the new training instances. However, the inherent occlusion and incompleteness of objects (*e.g.* the table and chairs in the left of Figure 2) coupled with highly imbalanced class-wise occurrence frequencies in point cloud data representation pose significant obstacles for both knowledge distillation and pseudo label generation. Specifically, the previous model may not provide sufficient knowledge to distill for some classes. Moreover, the pseudo labels generated in a naive manner from the frozen previous model using fixed thresholding can be inaccurate and incomplete, ultimately degrading the detection performance.

In view of these limitations, we propose the novel SDCoT++: Static-Dynamic Co-Teaching framework for class-incremental 3D object detection. In this framework, the incremental model is represented as a student, which is taught by two teachers: a static teacher which is a frozen copy of the previous model trained on old data, and a dynamic teacher which is an ensemble of the student model across its up-to-date training steps. Both teachers provide pseudo labels for the old classes to the student, balancing the potentially limited capacity of the static teacher for certain classes with the adaptive capabilities of the dynamic teacher. To alleviate the adverse effects of varying class occurrence frequencies on fixed thresholding during pseudo label selection, we opt for calibrating the probabilities of base classes according to their occurrence frequency, achieving more balanced class probabilities. This design allows us to use a single threshold for the balanced class probabilities, minimizing the number of required hyperparameters. Aside from providing explicit supervisions using pseudo labels, the static teacher offers distillation of knowledge from old data via a distillation loss, while the dynamic teacher transfers underlying knowledge from the new data to the student via a consistency loss. Notably, since the student and two teachers are 3D object detectors with identical architecture, our SDCoT++ framework can be seamlessly applied to any existing 3D object detection model with minimal implementation effort. To illustrate the invariance of our model to the backbone object detector, we showcase its effectiveness with three representative 3D object detection methods: VoteNet [1], 3DETR [2] and CAGroup3D [23].

Overall, our SDCoT++ achieves static-dynamic co-teaching by training the student model with supervision from the "mixed labels" (*i.e.* pseudo labels for base classes and ground-truth labels for novel classes) and regularization from the two

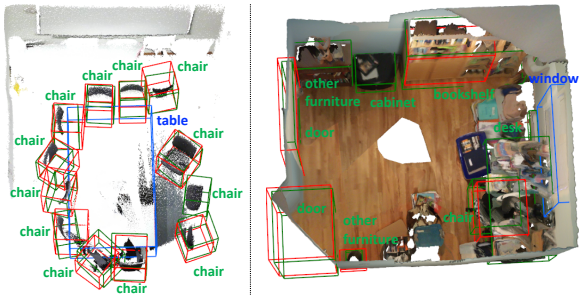

Fig. 2: Illustrations of pseudo annotations generated from SUN RGB-D (left) and ScanNet (right). Red bounding boxes represent pseudo labels w.r.t C_{base}. Green and Blue bounding boxes are ground truth labels w.r.t C_{base} and C_{novel}, respectively.

adversarial teachers. Our SDCoT++ demonstrates significant performance improvements over the baselines on extensive experiments conducted under both batch- and sequential-incremental learning settings with two indoor benchmark datasets: SUN RGB-D [24] and ScanNet [25]. Additionally, we conduct preliminary experiments on the outdoor KITTI [26] dataset to verify the applicability of our method in outdoor driving scenarios, further showcasing its versatility across different environments. Our contribution can be summarized as follows:

- We pioneer the exploration of the under-explored yet practical class-incremental 3D object detection task and propose a novel static-dynamic co-teaching-based method named SDCoT++.
- Our proposed SDCoT++ approach achieves static-dynamic co-teaching mechanism by (explicit) pseudo supervisions and (implicit) knowledge regularization from the two adversarial teachers.
- Our proposed SDCoT++ is backbone-agnostic, allowing for easy adaptation to any off-the-shelf 3D object detector to solve class-incremental learning problems.
- We conduct extensive experiments on two benchmark indoor datasets to demonstrate the effectiveness of our SDCoT++ approach in a variety of incremental learning scenarios. Additionally, we made an initial attempt to verify the effectiveness of our method in outdoor scenarios.

This work is an extension of our conference paper [27], presenting new contributions in three key aspects: 1) Unlike [27], which solely relies on the static teacher for pseudo-label generation, we enhance pseudo-label quality by a) integrating predictions from the dynamic teacher, and b) adjusting class probabilities based on object occurrences. This improvement in pseudo labels consistently enhances performance by annotating more base class objects missed by the static teacher. 2) We empirically validate the backbone-agnostic nature of our framework by employing VoteNet [1], 3DETR [2] and CAGroup3D [23] as object detectors, substantiating its efficacy through extensive experiments across two datasets under varying class-incremental learning scenarios. 3) We conduct a thorough analysis on the contribution of each component within SDCoT++, and demonstrate its effectiveness in both indoor and outdoor scenarios.

II. Related Work

A. 3D Object Detection

3D object detection methods can be categorized into three types based on the input scene representation: 1) Monocular image [28]–[32]; 2) RGB-D image [33]–[37]; 3) 3D point cloud [1], [2], [23], [38], [39]. In this paper, we focus on point cloud-based 3D object detection in indoor scenarios. Due to the irregular and sparse characteristics of 3D point clouds, it is challenging to localize 3D objects in an efficient way like its image-based counterparts. Moreover, 3D indoor scenarios pose greater detection challenges compared to outdoor environments due to their greater diversity and cluttered scenes with object overlapping. Several prior works [1], [38], [39] have addressed this challenge by exploring the sparsity of 3D data and generating 3D proposals around a set of seed points identified through voting. Among these voting based approaches, VoteNet [1] stands out as a representative with its simple yet efficient design. It employs PointNet++ [40] for point-wise feature extraction, and employs deep Hough voting and vote aggregation for object proposals. Unlike VoteNet, which primarily relies on single-scale feature aggregation, MLCVNet [38] enhances the voting mechanism by integrating features at multiple point patch levels and capturing a broader range of contextual details. H3DNet [39] expands intermediate features with geometric primitives including centers, edge centers and face centers of bounding boxes before voting object centers and generating 3D object proposals. Different from the geometric proximity based point clustering in VoteNet, CAGroup3D [23] generates proposals via class-aware point grouping, where it generates voxel-wise predictions as contextual cues that guide the grouping process. Recently, TR3D [41] explores real-time 3D object detection in indoor environments, highlighting the need for efficient processing in cluttered scenes. SPGroup3D [42] introduces a superpoint grouping network for indoor 3D object detection, focusing on enhancing the representation of geometric details in complex scenes.

Recently, inspired by the achievements of DETR [43] in 2D object detection, researchers have ventured into extending similar architectures to the 3D domain [2], [4]. These adaptations typically feature a query-based Transformer structure. However, unlike the learnable queries employed in DETR, the 3D counterparts typically utilize data-dependent priors to initialize queries, thereby reducing the search space in 3D data. For example, Groupfree [4] introduces KPS sampling to obtain initial object queries, followed by a Transformer decoder module to achieve final 3D object detection. 3DETR [2] replaces the PointNet++ backbone in Groupfree with a Transformer encoder, resulting in an end-to-end transformer-based solution for 3D object detection. Similar to Groupfree, it adopts non-parametric object queries, which are sampled from input point cloud using Farthest Point Sampling (FPS) [40]. Uni3DETR [44] presents a unified 3D object detector for both indoor and outdoor scenes, incorporating a mixture of parametric (*i.e.* learnable) and non-parametric query points. V-DETR [6] further improves the locality in the cross-attention mechanism by encoding relative positions of points to the predicted 3D boxes.

In this paper, we select three representative 3D object detection models: the voting based VoteNet, the DETR-based 3DETR, and the recent CAGroup3D as our detection backbones for class incremental learning. We identify several issues with the original VoteNet, 3DETR, and CAGroup3D when adapting to the class incremental learning setting, and demonstrate that these issues can be easily resolved with minimal implementation effort, as discussed in Section III-C.

B. Class-incremental Learning

Class-incremental learning is a well-established challenge in machine learning, characterized by the continuous integration of novel classes into a model [45]. Recent surge of popularity in deep learning techniques increases the importance of class-incremental learning as deep learning methods usually have the impractical requirement of large amounts of labeled data for all the classes/tasks for batch training. The majority of current class-incremental learning approaches are concentrated on the task of image classification, and can be categorized into three main types: 1) *Regularization-based methods* [11], [16], [46]–[49] that aim to minimize the discrepancy between either the data [11], [48] or parameters [16], [46], [47], [49] in the preceding and the current model; 2) *Rehearsal/replay-based methods* [12], [13], [50]–[53] that store a subset of examples from previous tasks [12], [13], [50], [51], [54], [55] or produces samples for previous tasks from a generative model [52], [53] to prevent forgetting of previous tasks; 3) Network expansion methods that dynamically adjust the network to accommodate the evolving data stream by adding neurons [56], duplicating backbones [57], [58], and expanding tokens and prompts in the Vision Transformer [14], [59]–[61]. In the scenario of 3D object detection, the replay-based methods are not a good choice because of the larger memory budget of 3D data and the difficulty of learning 3D generative models. Network expansion is a promising direction for 2D data but lacks crucial prerequisites including the foundation model for point clouds. In contrast, regularization-based methods are more practical as they do not need to store any old data. We therefore take inspiration from regularization-based method when designing our solution for class-incremental 3D object detection. Parameter-based regularization aims to restrain weight drift of important parameters in the previous model when fine-tuning on new data sets. These methods need to additionally estimate the importance of each parameter in the previous model. However, determining the importance of the parameters is not trivial. Data-based regularization aims to restrain activation drift of outputs from the previous model via knowledge distillation [62], which usually describes a strategy that transfers knowledge from a large network to a small network for efficient deployment. In this paper, we leverage on the concept of knowledge distillation to transfer knowledge from the previous 3D object detector to the current detector.

C. Class-incremental Object Detection

In recent studies, several works [17]–[22], [63]–[65] apply incremental learning on the task of image-based object detection. Among them, exemplar replay (retraining a subset of

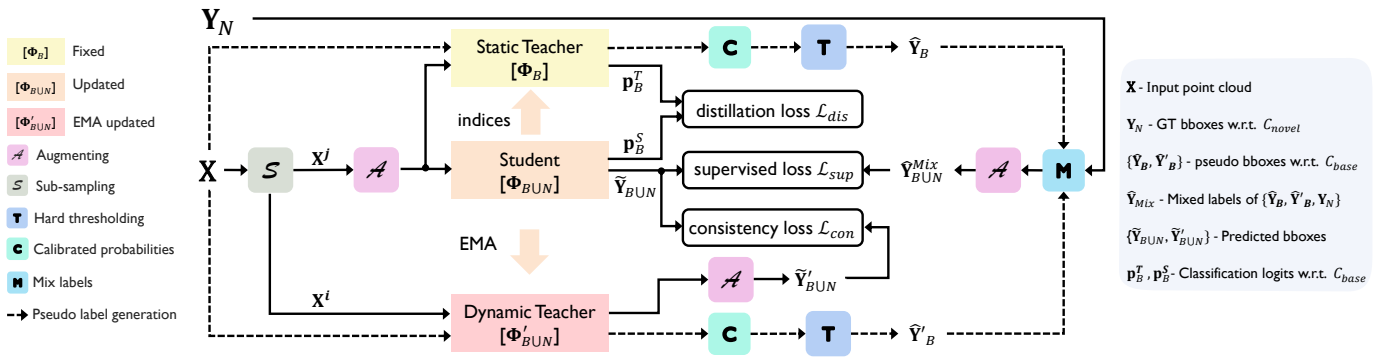

Fig. 3: An illustration of the proposed SDCoT++ architecture. Both student and two teacher models are 3D object detectors. The static and dynamic teachers generate pseudo labels for base classes, which are mixed with ground-truth novel class labels to supervise the student model. At the same time, the static teacher also distills base class knowledge to the student via $\mathcal{L}_{dis}$ and the dynamic teacher regularizes the student via $\mathcal{L}_{con}$.

examples from previous tasks) and knowledge distillation on network responses (*e.g.* outputs of the classifiers or features of intermediate layers) are the two most investigated approaches. Wang *et al.* [21] leverage memory replay with data compression (MRDC) for incremental object detection, and propose determinantal point processes (DPPs) to balance the quality-quantity trade-off in the memory buffer. Shmelkov *et al.* [17] pioneer the study of incremental image object detection. They employ Fast-RCNN [66] as the object detector and implement distillation losses on the both classification and proposal regression outputs for overcoming the catastrophic forgetting. However, this method is not very efficient as it requests an offline object proposal step that is computationally expensive. Feng *et al.* [20] improve knowledge distillation by selecting classification and regression outputs with different thresholds derived through Elastic Response Selection (ERS). Several works have explored the joint effect of exemplar replay and knowledge distillation for incremental object detection. Liu *et al.* [18] adapt Elastic Weight Consolidation (EWC) [46], which regularizes network parameters during incremental learning, to 2D object detection. They achieve the adaptation by introducing two modifications: 1) using pseudo annotations of bounding boxes for old classes in new training samples; 2) replacing quadratic regularization with Huber regularization in the regularized loss of EWC. More recently, CL-DETR [64] designs detector knowledge distillation (DKD) for Transformer-based incremental object detection and distribution-preserving calibration that matches classes in the exemplar set to the training distribution. Similar to [18], [64], pseudo labels of previous categories are also employed by us to deter the new model from incorrectly identifying objects of old classes as background in new samples. However, techniques specifically engineered for 2D images and 2D object detection backbones cannot be trivially adapted to the point cloud-based 3D object detection task. In contrast to the image-based context, the pseudo annotations generated in the 3D setting lack precision, which can lead to a decline in performance. To address this challenge, we introduce a novel static-dynamic co-teaching technique to alleviate deficiencies of pseudo labels as described in Section III.

III. Methodology

A. Problem Definition

We split all classes into two disjoint sets, *i.e.* *base classes* set C_{base} and *novel classes* set C_{novel} for class-incremental 3D object detection. D_{base} denotes the set of scenes for C_{base}, and D_{novel} represents the set of scenes for C_{novel}. The **class-incremental 3D object detection** problem is formulated as: with a 3D object detector Φ_B (the base model) pre-trained on D_{base}, we aim to acquire an incremental 3D object detector $\Phi_{B\cup N}$ (the incremental model) using only D_{novel}, which is capable of detecting objects of all classes seen so far, *i.e.* $C_{base} \cup C_{novel}$.

The key challenge in class-incremental 3D object detection is the high possibility of co-occurrence of both unlabeled base classes C_{base} and labeled novel classes C_{novel} in some scenes. As a result, these regions that contain C_{base} appear as background and are incorrectly suppressed when training the incremental model $\Phi_{B\cup N}$, which hasten forgetting. Moreover, the presence of unannotated C_{base} is confusing to $\Phi_{B\cup N}$.

A straightforward solution for missing C_{base} labels is to produce pseudo labels for C_{base} from predictions of a frozen pre-trained model Φ_B via a fixed threshold. While pseudo-labeling works in 2D domain based applications such as class-incremental 2d object detection [18], [64], this naive approach is less effective in our case for the following reasons. Firstly, compared to 2D images, occlusion and incompleteness of objects are more severe in 3D point clouds, leading to less accurate predictions. Secondly, the benchmark datasets for 3d object detection suffer from more pronounced class imbalance and co-occurence issues compared to their 2D counterparts, leading to biased predictions across different classes. The application of a uniform threshold to filter pseudo annotations across all classes is thus empirically sub-optimal for each individual class [67], [68]. As a result, this naive pseudo-labeling method would produce incomplete (missing objects) and inaccurate (wrong classification) pseudo labels.

We therefore propose static-dynamic co-teaching to circumvent the limitations of basic pseudo-labeling, which we will elaborate in the following sections.

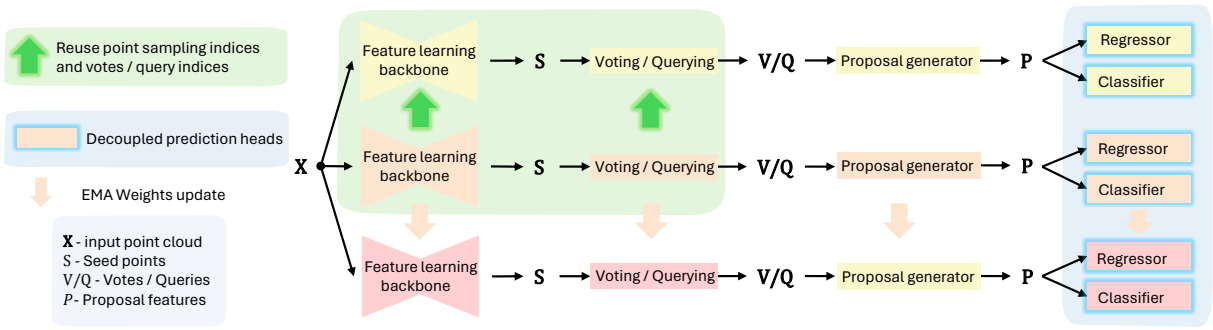

Fig. 4: An illustration of our modifications to backbones described in Section III-C, including 1) green represents the reuse of indices and 2) blue denotes the decoupled classification and localization. Sub-sampling and augmentation to the input point cloud X are omitted for brevity.

B. Static-Dynamic Co-Teaching

Figure 3 illustrates the framework of our proposed static-dynamic co-teaching framework, named SDCoT++, which is composed of three networks: a student $\Phi_{B\cup N}$, a static teacher Φ_B, and a dynamic teacher $\Phi'_{B\cup N}$. We develop our static-dynamic co-teaching approach on the premise that the student model $\Phi_{B\cup N}$ is less vulnerable to inaccurate and incomplete annotations when it can extensively utilize the old knowledge from the base model Φ_B and the underlying knowledge of new data D_{novel} from $\Phi'_{B\cup N}$. The static teacher shares the same architecture as the student and the dynamic teacher except for the classification head (*c.f.* Section III-C). The student is the incremental detector $\Phi_{B\cup N}$ that is progressively trained to detect C_{novel} objects and is jointly taught by both teachers. The static teacher is a frozen base model Φ_B, and the dynamic teacher $\Phi'_{B\cup N}$ is an exponential moving average (EMA) of the student. Both teacher models produce pseudo labels for C_{base} objects in D_{novel} and provide concurrent regularization to the student. We explain the design of co-teaching in this subsection, the modification to the backbone networks in Section III-C, the consolidation of pseudo labels in Section III-D, and the training and inference details in Section III-E.

Static Teacher. The pre-trained Φ_B captures essential knowledge of base classes C_{base}. Considering this, we utilize a frozen copy of Φ_B as our static teacher. By employing pseudo labels, we alleviate catastrophic forgetting of C_{base} due to the absence of base class labels in new training instances. For further extraction of knowledge from the static teacher, we implement a distillation strategy to align responses from the base and incremental models. Particularly, our distillation approach focuses on the prediction layer, calculating a distillation loss that quantifies the disparity in classification logits for C_{base} between the base and incremental models. This knowledge distillation scheme helps to offset the lack of labels for C_{base} objects that appear together with novel objects in a scene of D_{novel}. Additionally, the responses in the form of classification logits for C_{base} can offer some insights into the background, *i.e.* dark knowledge [62], [69], even in the absence of base class objects.

Dynamic Teacher. To leverage additional information from the new data, we also design a dynamic teacher $\Phi'_{B\cup N}$ capable of consistently acquiring the underlying knowledge associated with both base and novel classes. We draw inspiration for our dynamic teacher from Mean Teacher [70] which is a self-ensembling method initially introduced to utilize unlabeled data and mitigate over-fitting in semi-supervised learning scenarios. SESS [71] adapts the self-ensembling method for the semi-supervised 3D object detection task, introducing data perturbation and consistency regularization that ensures the consensus of locations, sizes, categories of the predicted proposals between the student and teacher models. Furthermore, they demonstrate superior performance with fully annotated data which is attributed to the consistency regularization inherent to the mean-teacher paradigm, which empowers their framework with the capability of extracting extra underlying knowledge from the training data. We thus integrate the dynamic teacher with consistency regularization to enhance the depth of knowledge extraction from the new data. In addition to the incorporation of consistency regularization, we posit that the dynamic teacher can offer adaptive knowledge *with respect to* the base classes along the training iterations. This dynamic teacher is therefore used to generate pseudo labels for the base classes to direct the student to be more robust against inaccurate pseudo labels in the new data while also improving its expressiveness on new classes. We initialize the parameters of the student and dynamic teacher networks from Φ_B, except for the additional weights in the classifier for novel classes which are initialized randomly.

Co-teaching Pipeline. In Figure 3, the dotted lines illustrate the process of pseudo-label generation and the solid lines indicate the flow of data and labels for model training and loss updates. For the static teacher, the input point cloud $\mathbf{X}$ is forwarded to the static teacher to generate K 3D bounding boxes for the base classes C_{base}. We calibrate the class probabilities of these 3D bounding boxes and then filter them with fixed thresholds to obtain the pseudo labels $\hat{\mathbf{Y}}_B$. Similarly, we obtain pseudo labels $\hat{\mathbf{Y}}'_B$ from the dynamic teacher except that only predictions for C_{base} are used. The pseudo labels $\hat{\mathbf{Y}}_B$ and $\hat{\mathbf{Y}}'_B$ are mixed with the ground-truth novel classes labels $\mathbf{Y}_N$ to create the set of consolidated labels $\hat{Y}^{Mix}_{B\cup N}$ for supervising the student model. As described in Section III-D,

overlapping pseudo labels are removed. Concurrently, our SDCoT++ performs two sub-sampling operations on the input $\mathbf{X}$ to generate two distinct point clouds, *i.e.* $\mathbf{X}^i$ and $\mathbf{X}^j$ in Figure 3. $\mathbf{X}^i$ is fed into the dynamic teacher network while $\mathbf{X}^j$ undergoes further augmentation before being input into the student and static teacher models. The sub-sampling and augmentation processes which include random flipping, rotation, and scaling, constitute parts of the perturbation scheme. This strategy enables the model to acquire valuable knowledge instead of merely memorizing the training data. Furthermore, we regularize the student model by 1) minimizing the discrepancies between the classification logits of proposals from the static teacher and the student, denoted as $\mathbf{p}_B^T$ and $\mathbf{p}_B^S$, respectively, and 2) ensuring consistency between the output proposals from the student and the dynamic teacher, referred to as $\tilde{\mathbf{Y}}_{B\cup N}$ and $\tilde{\mathbf{Y}}'_{B\cup N}$, respectively. The regularization loss functions are presented in III-E.

Discussion. Interestingly, the static and dynamic teachers play antagonistic roles in the learning process. The conservative static teacher ensures the student does not deviate too far from the knowledge of the base model. On the other hand, the radical dynamic teacher encourages the student to incorporate new knowledge. Despite these opposing forces, an equilibrium is achieved when the co-training converges due to the knowledge distillation from the static teacher and the consistency regularization by the dynamic teacher.

C. Anatomy of Backbones

Mainstream point-based 3D object detection networks either use voting-based or query-based methods. In this paper, we select one representative model for each type, *i.e.* VoteNet [1] for voting-based methods and 3DETR [2] for query-based methods as backbones of our 3D object detectors. Additionally, we adopt CAGroup3D [23], a recent two-stage method that demonstrates superior performance, as our backbone. Due to factors including randomness, coupled prediction and proposal alignment, the original VoteNet, 3DETR, and CAGroup3D cannot be directly used in our framework. In this section, we discuss the observed issues in these 3D object detectors that hinder incremental learning and distillation between the static teacher and the student. Subsequently, we demonstrate how these issues can be easily addressed by minor modifications to the detectors.

Issue 1: Stochasticity of Sub-sampling. Given the substantial volume of points present within a point cloud, sub-sampling becomes an indispensable step for object detection approaches in point clouds. For example, sub-sampling is performed within PointNet++ network in the case of utilizing PointNet++ [40] for point-wise feature extraction. Additionally, sub-sampling is employed during the selection of initial object proposals (*e.g.* queries). The sub-sampling step inherently introduces stochasticity regardless of employing random sampling or other operators such as farthest point sampling. Consequently, distinct sets of proposals are produced from the identical input point cloud at various iterations. This stochastic nature poses a challenge to the distillation process from the static teacher to the student model as the aligned sets of proposals are required for effective distillation.

Issue 2: Coupled Prediction. Some 3D object detection methods, such as VoteNet [1], MLCVnet [38], H3DNet [39] etc, couple the classification and localization prediction heads by estimating class-aware sizes for bounding boxes. However, this coupling mechanism is undesirable for incremental 3D object detection scenarios. This is primarily due to the continuous enrollment of new classes in class-incremental learning, necessitating dynamic updates to the weights associated with class-aware predictions. Unfortunately, the presence of a class-aware localization prediction head complicates the incremental model learning process. Consequently, decoupling the classification and localization prediction heads and resorting to class-agnostic size predictions yields a simplified implementation and alleviates the learning burden placed on the model. Note that this coupling issue is only present in VoteNet as 3DETR and CAGroup3D already decoupled the classification and localization heads.

Fix on VoteNet. VoteNet intrinsically incorporates two sub-sampling stages: 1) sub-sample M seeds from N input points via the feature learning backbone PointNet++; 2) sub-sample K **votes** from $\mathbf{V}$ as cluster centers to generate K proposals by aggregating neighboring votes. To circumvent stochasticity in these two sub-sampling steps, we retain all indices of the sampled points and votes in the student model, and re-use these indices for the static teacher. Consequently, the two sets of proposals produced from the two models are aligned and can be compared to measure the output discrepancy. Moreover, VoteNet employs one multi-layer perceptron (MLP) layer to process the proposal features and generate prediction scores for each proposal. The prediction scores includes $4NC$ box size scores and NC category scores. The box size scores comprise 1 classification score and 3 size offsets for each size template, which aligns with the corresponding class category. We solve the coupling issue by first splitting the final MLP layer into two layers, a regressor and a classifier, as illustrated in Figure 4. This segregation effectively separates the classification task from the predictions related to other targets. Subsequently, we only introduce new weights to the classifier for novel classes. In this way, the class-aware size prediction in vanilla VoteNet is replaced with a class-agnostic one, which facilitates class-incremental 3D object detection.

Fix on 3DETR. Similar to VoteNet, 3DETR also inherits the core sampling mechanism of PointNet++ [40], where M seeds are stochastically sub-sampled from N input points. In contrast to VoteNet, 3DETR sub-samples K **queries** from a larger set of generated queries after initial processing through transformer layers. Each query focuses on different regions or features of the input point cloud, guiding the model to identify potential object locations and characteristics. We fix the stochasticity in points and queries by storing all the indices of the sampled points and the position embeddings of queries from the student model for 3DETR, which are reused by the static teacher. Additionally, the coupling issue does not exist in 3DETR, and we incrementally add new weights for novel classes to the classifier in 3DETR.

Fix on CAGroup3D. CAGroup3D [23] voxelizes point clouds and adopts a voxel-based approach for proposal generation. The proposal generation involves 1) predicting offsets for voxels towards their instance centers, *i.e.*, the vote point prediction and 2) predicting semantic scores of the voxels to filter their vote points before grouping to produce proposals. To align proposals between the static teacher and the student model, we store the offsets and semantic scores from the student model and reuse them in the static teacher. In addition, the coupling issue does not apply to CAGroup3D. The classification head is incrementally expanded to accommodate novel classes in the experiments, facilitating class-incremental learning.

D. Enhanced Pseudo Labels Generation

Consolidated Pseudo Labels. The static teacher plays a pivotal role in strengthening base class knowledge in the context of novel scenes, while the dynamic teacher evolves through the learning process, gradually becomes more attuned and responsive to novel scenes, and offers increasingly refined guidance to the student. The adaptive knowledge from the continuously enhanced dynamic model contrasts with the relatively stable knowledge from the static teacher, which inspires us to consolidate outputs from the dynamic teacher into the existing set of pseudo labels. Specifically, we use both teachers to generate pseudo labels, *i.e.* $\hat{Y}_B$ from static teacher Φ_B and $\hat{Y}'_B$ from the dynamic teacher $\Phi'_{B\cup N}$ for C_{base} in each training sample in D_{novel}. Each pseudo label has an objectness score and a classification score, based on which we apply Non-Maximum Suppression (NMS) on filtered $\hat{Y}_B$ and $\hat{Y}'_B$ to remove overlapped bounding boxes. Furthermore, we use the same technique to filter out predicted bounding boxes that disagree with ground-truth labels for novel classes C_{novel} to reduce confusion between base and novel classes. Formally, the consolidated pseudo labels $\hat{Y}^{Mix}_{B\cup N}$ is formulated as follows:

$$\hat{Y}^{Mix}_{B\cup N} = \text{NMS}\big[Y_N, \text{NMS}[\psi(\hat{Y}_B \mid \tau_o, \tau_c), \psi(\hat{Y}'_B \mid \tau_o, \tau_c)]\big], \quad (1)$$

where $\psi(*|\tau_o, \tau_c)$ denotes the thresholding operator using two fixed thresholds τ_o and τ_c for objectness scores and classification probabilities, respectively. $\text{NMS}[a, b]$ denotes Non-Maximum Suppression (NMS) on the union of a and b.

Class Probability Calibration. As mentioned earlier, the significant class imbalance and co-occurrence issues prevalent in 3D object detection datasets pose challenges when employing fixed thresholding, particularly for filtering out low class probabilities represented by τ_c. Due to the class imbalance, certain classes may consistently exhibit low probabilities. Additionally, the use of softmax to obtain class probabilities introduces an exponential operation that amplifies discrepancies between logits and yields a more pronounced output, subsequently diminishing the probabilities associated with tail classes.

One approach to address this issue is to individually fine-tune class-specific thresholds $\{\tau^i_c\}^{C_{base}}_{i=1}$ for each class to determine their optimal values. However, the computational complexity involved in optimizing thresholds for all classes becomes prohibitively high. Consequently, we are inspired by [72] in proposing to calibrate the class probabilities of pseudo labels as an alternative solution instead of tuning class-wise thresholds. Subsequently, a single global threshold τ_c can be applied to the calibrated class probabilities.

Specifically, we utilize the object occurrence frequency of each class to calibrate the corresponding class probabilities. These occurrence frequencies are obtained from the class distribution of ground-truth objects during base training. The calibrated class probability is computed as:

$$\phi_j = \frac{n_j e^{\eta_j}}{\sum^k_{i=1} n_i e^{\eta_i}}, \quad (2)$$

where n_i stands for the number of objects belonging to class i in D_{base}, and η_j refers to the class logits before the Softmax operator.

Moreover, our approach distinguishes itself from methods like Calibrated Teacher [73]. We utilize object occurrence frequency to calibrate class probabilities based on the ground-truth distributions, ensuring that the calibrated probabilities reflect the inherent class imbalance in the dataset. This contrasts with Calibrated Teacher [73], which employs logistic regression for score adjustment without explicitly considering class distribution. Additionally, our method requires fewer trainable parameters and avoids the extra computation of fitting an additional calibration model. As a result, our method provides a data-driven calibration process particularly for scenarios with diverse class distributions and for incremental learning of new classes.

E. Training and Inference

Training. During training, the student model is updated for each training iteration t by a weighted sum of the three losses:

$$\mathcal{L} = \lambda_s \mathcal{L}_{sup} + \lambda_d \mathcal{L}_{dis} + \lambda_c \mathcal{L}_{con}, \quad (3)$$

where λ_s, λ_d, and λ_c are hyperparameters to control the contributions of each loss function.

The supervised loss $\mathcal{L}_{sup}$ is computed between the output proposals from the student $\tilde{\mathbf{Y}}_{B\cup N}$ and the mixed pseudo and ground-truth labels $\hat{\mathbf{Y}}^{Mix}_{B\cup N}$ that undergo the same augmentation process as $\mathbf{X}^j$. The supervised loss is computed using its original loss function when VoteNet [1] is adopted as the backbone. Similarly, we utilize the original supervised loss in [2] when 3DETR [2] and CAGroup3D [23] are the backbones.

The distillation loss $\mathcal{L}_{dis}$ is calculated to quantify the differences between the classification logits $\mathbf{p}^T_B$ and $\mathbf{p}^S_B$. To normalize the classification logits, we subtract their mean over class dimension, resulting in $\bar{\mathbf{p}}^T_B$ and $\bar{\mathbf{p}}^S_B$, respectively. The distillation loss is then expressed as follows:

$$\mathcal{L}_{dis} = \frac{1}{K} \sum^K_{i=1} ||\bar{\mathbf{p}}^S_{B,i} - \bar{\mathbf{p}}^T_{B,i}||_2, \quad (4)$$

where $\bar{\mathbf{p}}^*_{B,i}$ is a vector with a dimension of $|C_{base}|$ that denotes the normalized classification logits of i^{th} proposal.

Following SESS [71], the consistency loss $\mathcal{L}_{con}$ is computed by comparing the output proposals from the student with the output proposals of the dynamic teacher $\tilde{\mathbf{Y}}'_{B\cup N}$ augmented by the same steps as above. Formally, it is computed as:

$$\mathcal{L}_{\text{con}} = \lambda^c_1 \mathcal{L}_{\text{center}} + \lambda^c_2 \mathcal{L}_{\text{class}} + \lambda^c_3 \mathcal{L}_{\text{size}}, \quad (5)$$

where $\mathcal{L}_{center}$, $\mathcal{L}_{class}$ and $\mathcal{L}_{size}$ denote the center consistency loss, class consistency loss, and size consistency loss, respectively. We follow SESS to fix $\lambda_1^c = 1$, $\lambda_2^c = 2$ and $\lambda_3^c = 1$. Refer to SESS [71] for a more comprehensive explanation of the loss terms.

Once the student model is updated, the parameters of the dynamic teacher are updated using exponential moving average (EMA) of the parameters of the student: $\Phi'_t = \alpha\Phi'_{t-1} + (1-\alpha)\Phi_t$ [1]. α denotes a hyperparameter to determine the extent of information derived from the student.

Inference. During inference, an input point cloud is fed directly into the dynamic teacher to generate 3D bounding boxes. These boxes are then refined through a Non-Maximum Suppression (NMS) module.

IV. Experiments

A. Datasets and Settings

Datasets. We assess the 3D object detection performance of our SDCoT++ on two indoor datasets, SUN RGB-D [24] and ScanNet [25]. **SUN RGB-D** [24] comprises 5,285 training samples and 5,050 validation samples across hundreds of object classes. Following the established evaluation protocol used in prior works [1], [27], we focus our evaluation on the 10 most common categories. **ScanNet** [25] contains 1,201 training samples and 312 validation samples, where it does not provide oriented 3D bounding boxes but offers point-level semantic segmentation labels. We adopt the approach from VoteNet to extract axis-aligned bounding boxes from the point-level labels and evaluate on the same 18 object classes.

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 in SUN RGB-D does not perform well under batch incremental 3D object detection setting (see the results under $|C_{novel}| = 1$ setting in Table I. Furthermore, the number of instances per scan in ScanNet is larger than that in SUN RGB-D, which can be observed in our qualitative examples in the supplementary material, where the scenes in ScanNet are more cluttered.

Furthermore, we conduct preliminary experiments on the KITTI [26] outdoor driving dataset. The KITTI dataset is a widely used benchmark in autonomous driving. It comprises 7,481 training samples and 7,518 test samples. The training data is typically divided into a training split of 3,712 samples and a validation split of 3,769 samples.

Setup. We facilitate the datasets for class-incremental learning by selecting a subset of classes by the alphabetical order in one dataset as C_{base} and leaving the remainder as C_{novel}, aligning with the class splitting approach used in [17]. D_{base} consists of training samples that include any class of C_{base}, disregarding annotations for C_{novel}. D_{novel} is formed similarly for C_{novel}. D_{base} and D_{novel} might include duplicating point clouds. However, their difference in labels reflects the shifted focus on the classes.

[1]The subscripts of $\Phi_{B\cup N}$ and $\Phi'_{B\cup N}$ are omitted for brevity.

We assess the performance of our SDCoT++ under two distinct class-incremental 3D object detection scenarios. 1) ***Batch incremental learning.*** In this scenario, all novel classes are introduced simultaneously, allowing $\Phi_{B\cup N}$ to be updated with the full set of novel classes. To address potential biases due to specific class characteristics, we consider different numbers of novel classes in this scenario. Specifically, we split the classes by: a) $|C_{novel}| = |C_{base}|$; b) $|C_{novel}| < |C_{base}|$ and $|C_{novel}| > 1$; c) $|C_{novel}| = 1$. 2) ***Sequential incremental learning.*** The novel classes are divided into subsets and introduced to the system sequentially. In this scenario, the static teacher network is updated using the currently learned student network after each incremental step. The names of the base and novel classes across different splits are provided in the supplementary material.

In the outdoor scenario, we select *Car*, *Pedestrian* and *Cyclist* as the three base classes. We choose *Van* and *Truck* as the two novel classes for batch-incremental learning.

Evaluation Metric. The primary metric we use for evaluating 3D object detection performance is the mean average precision (mAP). For ScanNet and SUN RGB-D, our reported results are based on mAP calculated with a 3D Intersection over Union (IoU) threshold of 0.25, *i.e.* mAP@0.25. For the KITTI dataset, we use mAP with the same IoU thresholds as in [74] for *Car*, *Pedestrian*, and *Cyclist*, and apply the same IoU thresholds as *Car* for *Van* and *Truck*.

B. Implementation Details

In the experiments, we select three representative models, *i.e.* VoteNet [1] for voting-based methods, and 3DETR [2] for query-based methods, and two-stage CAGroup3D [23] as our 3D object detector backbones. We assign τ_o and τ_c for pseudo labels as 0.95 and 0.9, respectively. The weights assigned in the loss function Eq. 3 are λ_s=10, λ_d=1, and λ_c=10. A ramp-up strategy [70] is employed to gradually increase the impact of λ_d and λ_c during the first 30 epochs, utilizing a sigmoid-shaped function $e^{-5(1-t)^2}$, where t progresses linearly from 0 to 1 throughout the ramp-up phase. Following SESS [71], we assign α in EMA as 0.99 during ramp-up and adjust it to 0.999 for subsequent training. For VoteNet and 3DETR, the base network Φ_B and the student network $\Phi_{B\cup N}$ are optimized using the Adam optimizer. The initial learning rate for Φ_B is 0.001, which is reduced by a factor of 0.1 at the 80^{th} and 120^{th} epoch. For CAGroup3D, Φ_B is optimized with the AdamW optimizer [75]. The initial learning rate and the weight decay are 0.001 and 0.0001, respectively. The initial learning rate for $\Phi_{B\cup N}$ is adjusted according to the specific settings of class-incremental learning.

C. Baselines

We compare our approach with the following baselines for class-incremental 3D object detection.

- "*freeze and add*": freezes the base model Φ_B trained with D_{base} and appends a new classifier for C_{novel} trained on D_{novel} to the classification branch of Φ_B.

TABLE I: *Batch incremental* 3D object detection performance (mAP@0.25) in the **SUN RGB-D val** set with various backbones. No. 1, 10, and 19 are the base models trained only on C_{base} using different backbones. No. 2-8, No. 11-17, and No. 20-22 are incremental learning methods initialized by the corresponding base model and trained on C_{novel}. No. 9, 18, and 23 represent the corresponding model jointly trained on all classes.

	Method	Model	$\|C_{novel}\|$ = 5			$\|C_{novel}\|$ = 3			$\|C_{novel}\|$ = 1		
			Base	Novel	All	Base	Novel	All	Base	Novel	All
1	Base training	VoteNet	57.58	-	-	53.73	-	-	55.10	-	-
2	Freeze and add		54.24	10.61	32.42	51.94	12.64	40.16	54.63	0.90	49.26
3	Fine-tuning		3.48	54.09	28.79	4.10	60.17	20.92	14.86	1.38	13.51
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}		50.35	59.88	55.12	37.91	66.39	46.45	30.85	29.96	30.76
6	SDCoT w/o L_{con}		52.92	57.11	55.01	41.81	63.45	48.30	31.61	25.78	31.02
7	SDCoT [27]		53.61	60.80	57.21	44.48	**67.41**	51.36	36.81	**42.69**	37.40
8	**SDCoT++**		**53.95**	**61.78**	**57.87**	**44.88**	67.33	**51.62**	**38.44**	41.72	**38.77**
9	Joint training		58.92	58.80	58.86	54.8	68.33	58.86	55.36	90.36	58.86
10	Base training	3DETR	56.60	-	-	55.30	-	-	56.39	-	-
11	Freeze and add		54.15	10.18	32.17	55.01	11.95	42.09	55.48	0.65	50.00
12	Fine-tuning		3.83	53.94	28.89	4.40	60.71	21.29	15.92	1.31	14.46
13	SDCoT w/o L_{dis} & L_{con}		49.55	58.11	53.83	41.97	65.81	49.12	27.01	24.68	26.78
14	SDCoT w/o L_{dis}		50.44	60.64	55.54	42.13	67.94	49.87	31.19	29.91	31.06
15	SDCoT w/o L_{con}		51.37	61.02	56.20	42.77	67.56	50.21	32.26	27.28	31.76
16	SDCoT [27]		53.13	60.64	56.89	44.35	67.62	51.33	37.81	**43.13**	38.34
17	**SDCoT++**		**55.12**	**60.92**	**58.02**	**45.35**	**69.43**	**52.57**	**40.02**	43.11	**40.33**
18	Joint training		57.92	59.98	58.95	54.37	69.63	58.95	55.39	91.00	58.95
19	Base training	CAGroup3D	62.23	-	-	60.09	-	-	63.84	-	-
20	Fine-tuning		4.37	67.69	36.03	4.45	70.91	24.39	17.24	1.94	15.71
21	SDCoT [27]		56.80	67.06	61.93	49.57	71.11	56.03	41.25	**46.47**	41.76
22	**SDCoT++**		**58.01**	**67.31**	**62.66**	**52.73**	**71.47**	**58.35**	**43.37**	43.69	**43.40**
23	Joint training		65.40	68.20	66.80	63.32	74.92	66.80	63.90	92.91	66.80

TABLE II: *Batch incremental* 3D object detection performance (mAP@0.25) in the **ScanNet val** set with various backbones. No. 1, 10, and 19 are the base models trained only on C_{base} using different backbones. No. 2-8, No. 11-17, and No. 20-22 are incremental learning methods initialized by the corresponding base model and trained on C_{novel}. No. 9, 18, and 23 represent the corresponding model jointly trained on all classes.

	Method	Model	$\|C_{novel}\|$ = 9			$\|C_{novel}\|$ = 4			$\|C_{novel}\|$ = 1		
			Base	Novel	All	Base	Novel	All	Base	Novel	All
1	Base training	VoteNet	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}		51.21	53.58	52.39	48.45	69.82	53.19	48.60	30.07	47.57
6	SDCoT w/o L_{con}		53.31	51.22	52.26	48.54	67.52	52.76	49.31	30.52	48.26
7	SDCoT [27]		**53.75**	54.91	54.33	49.50	70.85	54.25	52.01	31.71	50.89
8	**SDCoT++**		53.41	**56.81**	**55.11**	**50.89**	**71.18**	**55.40**	**54.08**	**33.46**	**52.94**
9	Joint training		58.90	54.13	56.51	53.16	68.23	56.51	57.83	34.16	56.51
10	Base training	3DETR	64.80	-	-	59.73	-	-	65.37	-	-
11	Freeze and add		60.99	4.07	32.53	55.59	1.81	43.64	60.46	0.14	57.11
12	Fine-tuning		0.22	56.45	28.33	0.30	64.19	14.50	0.07	29.16	1.69
13	SDCoT w/o L_{dis} & L_{con}		63.20	59.86	61.53	56.43	72.46	59.99	62.67	35.25	61.15
14	SDCoT w/o L_{dis}		68.08	55.13	61.61	60.73	69.06	62.57	64.59	36.70	63.04
15	SDCoT w/o L_{con}		66.02	55.22	60.62	59.73	66.72	61.28	65.26	27.50	63.16
16	SDCoT [27]		66.79	61.25	64.02	59.96	71.97	62.63	65.45	**40.81**	64.08
17	**SDCoT++**		**69.11**	**61.76**	**65.44**	**61.57**	**72.99**	**64.11**	**66.56**	38.25	**64.98**
18	Joint training		68.70	62.02	64.94	63.02	73.55	64.94	66.43	39.60	64.94
19	Base training	CAGroup3D	74.25	-	-	70.02	-	-	71.38	-	-
20	Fine-tuning		2.73	63.63	33.18	1.43	73.72	17.49	1.10	39.76	3.25
21	SDCoT [27]		71.04	**65.57**	68.31	66.04	75.72	68.19	67.66	**43.88**	66.34
22	**SDCoT++**		**72.35**	65.22	**68.79**	**66.81**	**77.59**	**69.21**	**68.57**	43.29	**67.16**
23	Joint training		77.91	72.33	75.12	73.31	81.46	75.12	75.80	63.60	75.12

- "*fine-tuning*": updates all parameters of the base model (excluding the old classifier) adds a new classifier for C_{novel} with D_{novel}.
- "*SDCoT and its variants*" includes SDCoT [27] and its three variants with each omitting a different component: 1) Without the distillation loss ($\mathcal{L}_{dis}$); 2) Without the consistency loss ($\mathcal{L}_{con}$); 3) Without both of them. When the consistency loss is omitted, the dynamic teacher is entirely removed; when the distillation loss is omitted, the static teacher is only utilized for generating pseudo labels.
- "*Joint training*": the model is fully supervised for all classes in all scenes, which serves as a reference of the upper-bound performance.

D. Quantitative Results

Batch Incremental Learning. Table I and II demonstrate the batch incremental 3D object detection results performed under 3 base-novel splits on SUN RGB-D and ScanNet, respectively. In both tables, VoteNet serves as the backbone model for experiments No. 1-9, 3DETR is used for experiments No. 10-18, and CAGroup3D is adopted for experiments No. 19-23. Experiment No. 1 is the base training on C_{base}. Experiments No. 2-8 list the results when C_{novel} is added in one batch, and experiment No. 9 is an upper-bound jointly trained on $C_{base} \cup C_{novel}$. Results are arranged in a similar manner for experiments No. 10-18 and No. 19-23. As presented in the tables, straightforward solutions including "freeze and add" and "fine-tuning" exhibit significant performance degradation for either novel or base classes across all splits on both datasets. The "freeze and add" approach yields inferior results for C_{novel} despite generally maintaining the performance on C_{base}. Conversely, "fine-tuning" the model on C_{novel} results in catastrophic forgetting of C_{base}.

Incorporating pseudo labels with ground-truth labels (No. 4 and 13 in Table I and II) notably aids the incremental model in retaining knowledge from previous classes. Moreover, adding the distillation loss (No. 6 versus 4 and 15 versus 13 in Table I and II) leads to noticeable improvements on the base classes across various settings compared to only using pseudo labels alone, indicating that the distillation loss effectively leverages additional knowledge from the static teacher. Notably, the performance with $\mathcal{L}_{dis}$ also exceeds that without $\mathcal{L}_{dis}$ for novel classes in most settings, suggesting that the distillation loss helps to minimize confusion caused by background regions in the incremental model. The addition of the consistency loss (No. 5 compared to No. 4 and 14 compared to No. 13 in Table I and II) leads to consistent and notable improvements on the novel classes for VoteNet and 3DETR, respectively. These improvements highlight the effectiveness of the dynamic teacher in assimilating the underlying knowledge from new data. Comparing No. 13-15, it is noted that the superior performance of No. 13 over No. 14 and 15 under the 'Novel' split can be attributed to the different roles of the loss functions and backbone architecture. The inclusion of the consistency loss (L_{con}) in No. 14 helps transfer knowledge but shows limited gains in novel classes due to 3DETR's strong retention of base-class knowledge. No. 15, which uses the distillation loss (L_{dis}), improves base-class performance but sacrifices novel class results. The observation supports the discussion on balancing performance for base and novel classes at the end of Section III-B. Combining the three losses (No. 7 and 16 in Table I and II), our SDCoT [27] demonstrates superior performance on both base and novel classes compared to its three variants, showcasing its capability to adapt to new knowledge while preserving existing knowledge. Finally, our SDCoT++ outperforms all baselines under all base-novel splits for both datasets with both backbone models. Our approach consistently enhances detection performance for base classes, which effectively alleviates the forgetting problem by providing better pseudo labels through incorporating pseudo labels from the dynamic teacher and calibrating class probabilities. Moreover, it also lifts performance in novel classes under some settings, *e.g.* VoteNet-ScanNet-C_{novel} = 9, which can be attributed to the removal of pseudo labels overlapping ground-truth labels for novel classes during mixed label generation. It is worth noting that the experiments with 3DETR as the backbone show more significant improvements overall, bringing performance closer to the oracle baseline. This enhanced performance with 3DETR underscores the effectiveness of our approach in leveraging transformer-based architectures for complex class-incremental learning tasks. In addition, the observation that joint training performs worse than base training for base classes with VoteNet is attributed to the presence of novel classes acting as distractions. For instance, classes like "desk" and "table" share similar features and spatial characteristics, which can confuse the model during joint training. This overlap complicates class distinction, resulting in degraded performance compared to base training, where such distractions are absent.

To validate generality of our approach with more advanced detectors, we incorporate experiments using CAGroup3D [23]. It can be observed that the proposed SDCoT [27] significantly outperforms the fine-tuning baseline under all settings, demonstrating its capability to retain knowledge of base classes. Moreover, SDCoT++ consistently improves performance on base classes across both datasets and splits, contributing to better overall performance. It also maintains or enhances detection performance for novel classes, *e.g.* CAGroup3D-ScanNet-C_{novel} = 4.

Sequential Incremental Learning. Table III and IV present the average precision (AP) for each class when novel classes are incrementally introduced in a sequential manner for class-incremental learning. We evaluate our method using two consecutive subsets of novel classes on SUN RGB-D and ScanNet, respectively. The incremental model first adapts to the initial subset of novel classes using the base model. The model after adaptation is considered the new base model, which is used to initialize the incremental model trained on the next subset of novel classes. In SUN RGB-D, SDCoT++ outperforms SDCoT at two consecutive incremental learning steps, regardless of the choice of backbones, as shown in Table III. The advantage of SDCoT++ is also seen in Table IV, which confirms the validity of our SDCoT++ under the sequential incremental learning scenario.

TABLE III: *Sequential incremental* object detection performance (AP@0.25) of each class in the **SUN RGB-D val** set. The model sequentially learns 5 novel classes in 2 batches. B[1-5] represents base training on 5 base classes. +N[6,7,8] and +N[9, 10] denote incremental training on novel classes. B[1-10] represents joint training on all classes.

	Model	bath	bed	bookshelf	chair	desk	dresser	nightstand	sofa	table	toilet	mAP
B[1-5]	VoteNet	74.71	85.53	32.75	73.02	24.96	-	-	-	-	-	58.19
+ N[6,7,8] SDCoT [27]		**51.57**	84.04	23.83	62.83	**16.94**	**26.04**	57.34	59.75	-	-	47.79
+ N[6,7,8] SDCoT++		49.67	**84.52**	**29.46**	**65.87**	15.27	25.99	**59.99**	**61.62**	-	-	**49.05**
+ N[9, 10] SDCoT [27]		36.59	79.60	10.35	60.12	**15.16**	12.80	35.15	56.51	**46.95**	88.08	44.13
+ N[9, 10] SDCoT++		**38.91**	**82.47**	**13.75**	**61.34**	14.28	**15.35**	**38.59**	**59.26**	44.93	**89.32**	**45.82**
B[1-10]		78.49	84.31	32.62	73.73	25.44	30.90	58.11	64.15	50.48	90.36	58.86
B[1-5]	3DETR	73.97	84.71	30.19	75.09	23.93	-	-	-	-	-	57.58
+ N[6,7,8] SDCoT [27]		71.01	**82.44**	**28.28**	**73.38**	21.94	30.99	59.75	64.38	-	-	54.02
+ N[6,7,8] SDCoT++		**71.93**	82.34	28.06	72.50	**24.04**	**31.26**	**60.81**	**65.40**	-	-	**54.54**
+ N[9, 10] SDCoT [27]		46.61	80.89	15.29	61.41	15.04	**14.51**	37.71	59.22	46.57	89.17	46.64
+ N[9, 10] SDCoT++		**48.78**	**83.84**	**17.59**	**62.81**	**17.91**	14.09	**38.64**	**59.45**	**47.20**	**89.39**	**47.97**
B[1-10]		69.80	84.60	28.50	72.40	34.30	29.60	61.40	65.30	52.60	91.00	58.95

TABLE IV: *Sequential incremental* object detection performance (AP@0.25) of each class in the **ScanNet val** set. The model sequentially learns 5 novel classes in 2 batches. B[1-14] represents base training on 5 base classes. +N[15,16] and +N[17, 18] denote incremental training on novel classes. B[1-18] represents joint training on all classes.

	Model	bath	bed	bkshlf	cbnt	chair	cntr	crtn	desk	door	ofurn	pctr	refri	shwr	sink	sofa	table	toil	wind	mAP
B[1-14]	VoteNet	75.93	84.17	47.86	35.73	87.09	51.50	44.02	68.67	45.52	41.47	6.86	44.08	60.13	50.97	-	-	-	-	53.14
+ N[15, 16] SDCoT [27]		49.10	84.28	39.24	**30.70**	**86.16**	**39.16**	**40.29**	**58.86**	**35.09**	33.60	**2.66**	**41.51**	28.72	**50.02**	86.65	56.66	-	-	47.67
+ N[15, 16] SDCoT++		**61.67**	**84.81**	**44.38**	29.18	85.51	27.42	32.79	57.89	33.26	**33.89**	2.11	39.02	**52.29**	39.02	**87.17**	**56.98**	-	-	**47.96**
+ N[17, 18] SDCoT [27]		39.31	83.22	**37.60**	18.62	82.04	0.39	30.76	**36.78**	21.57	30.48	0.11	**33.38**	**27.48**	**19.70**	84.32	**57.18**	**95.34**	**37.73**	40.89
+ N[17, 18] SDCoT++		**50.08**	**84.25**	35.09	**22.20**	**84.02**	**2.02**	**31.55**	33.61	**30.32**	**30.52**	**1.39**	31.95	26.13	16.92	**86.58**	56.89	93.74	36.09	**41.85**
B[1-18]		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
B[1-14]	3DETR	78.88	79.68	39.56	49.82	87.67	57.65	45.59	71.52	48.48	47.67	9.65	53.56	69.71	68.52	-	-	-	-	57.71
+ N[15, 16] SDCoT [27]		71.81	78.79	**27.14**	**40.78**	**87.59**	**52.53**	32.03	52.54	**40.15**	44.79	**9.75**	51.50	61.84	60.00	**86.30**	**66.27**	-	-	53.99
+ N[15, 16] SDCoT++		**74.91**	**79.62**	25.64	35.80	87.58	52.01	**34.95**	**58.27**	40.12	**46.27**	9.25	**53.97**	**63.67**	**65.38**	86.15	65.11	-	-	**54.92**
+ N[17, 18] SDCoT [27]		71.40	**78.91**	20.96	**37.44**	**88.31**	47.05	31.91	45.82	37.75	43.05	**9.67**	46.60	53.02	55.64	84.89	61.43	**96.10**	**37.08**	52.61
+ N[17, 18] SDCoT++		**73.89**	78.73	**21.73**	32.14	87.99	**49.57**	**34.71**	**51.38**	**39.03**	**45.96**	8.28	**48.50**	**54.65**	**62.88**	**85.50**	**62.88**	95.32	35.54	**53.82**
B[1-18]		92.20	83.60	56.40	49.40	90.90	55.90	58.30	79.20	52.40	53.00	15.20	57.60	67.60	70.60	89.80	67.60	97.20	39.60	65.36

Furthermore, when using VoteNet as backbone, comparing batch incremental and sequential incremental experiments, we attain a 45.82% mAP across all classes in SUN RGB-D after sequentially adding 5 novel classes in two batches, as shown in the final entry of the fifth row in Table III. The result is lower than the 57.87% mAP obtained by adding the 5 classes all at once, which is reported in the $|C_{novel} = 5, All|$ column in the eighth row of Table I. A similar trend is observed on ScanNet, indicating that the sequential incremental learning setting is more challenging than batch incremental learning and merits further investigation by future research. Examining the performance of each base class individually in Table III and IV, we notice that the classes experiencing significant performance drops during sequential incremental learning are where the model tends to have weaker detection capabilities in the initial base training stage (base training). Although there is a performance decline in sequential incremental learning compared to batch incremental learning, it does not lead to severe catastrophic forgetting, as observed with fine-tuning.

E. Ablation Studies

We present ablation studies of SDCoT++ in Table V, VI, and VII to investigate the effects of the two components introduced in this paper: consolidated pseudo labels and class probability calibration, which enhance the pseudo label generation, across different incremental settings. In all the three tables, SDCoT with consolidated pseudo labels (SDCoT+) consistently outperforms the original SDCoT with both backbones and under all splits. This validates the effectiveness of using consolidated pseudo labels from both teachers for class incremental 3D object detection. Moreover, introducing class probability calibration (SDCoT++) further enhances the model. In Table. V and VI, the performance gain is consistent for base classes while its effect on novel classes is less stable, especially in the extreme case where $C_{novel} = 1$. In the challenging sequential incremental learning scenario, the additional probability calibration brings performance enhancement to SDCoT++ in every incremental stage compared to SDCoT+, as demonstrated in Table. VII.

We conduct additional experiments to assess the impact of different hyperparameters, particularly on the thresholds for objectness scores (τ_o) and classification probabilities (τ_c) as incremental learning is sensitive to these choices. As presented in Table IX and Table X, we vary τ_o and τ_c in steps of 0.05 to observe their effects on model performance. The results show that the optimal objectness threshold ($\tau_o = 0.95$) and classification probability threshold ($\tau_c = 0.9$) produced the best overall performance (mAP@0.25). While higher thresholds help filter out incorrect pseudo-labels, overly strict thresholds can lead to a performance drop, indicating that careful balancing of these parameters is crucial for maximizing performance.

F. Strategies for Distillation Loss Configuration

We explore the impacts of different configurations of the distillation loss focusing on various distillation targets, such as classification logits and bounding box regression

TABLE V: Ablation studies in the *batch incremental* setting. The results (mAP@0.25) are reported in the **SUN RGB-D val** set. **SDCoT+** refers to SDCoT with mixed pseudo labels proposed in Section III-D. **SDCoT++** refers to SDCoT with mixed pseudo labels and calibrated probabilities proposed in Section III-D and Section III-D, respectively.

	Method	Model	$\|C_{novel}\|$ = 5			$\|C_{novel}\|$ = 3			$\|C_{novel}\|$ = 1		
			Base	Novel	All	Base	Novel	All	Base	Novel	All
1	SDCoT [27]	VoteNet	53.61	60.80	57.21	44.48	**67.41**	51.36	36.81	**42.69**	37.40
2	SDCoT+	VoteNet	53.81	61.26	57.54	44.53	67.36	51.38	38.41	37.79	38.35
3	**SDCoT++**	VoteNet	**53.95**	**61.78**	**57.87**	**44.88**	67.33	**51.62**	**38.44**	41.72	**38.77**
4	SDCoT [27]	3DETR	53.13	60.64	56.89	44.35	67.62	51.33	37.81	43.13	38.34
5	SDCoT+	3DETR	54.34	**60.99**	57.67	45.19	69.08	52.36	39.97	43.01	40.27
6	**SDCoT++**	3DETR	**55.12**	60.92	**58.02**	**45.35**	**69.43**	**52.57**	**40.02**	**43.11**	**40.33**

TABLE VI: Ablation studies in the *batch incremental* setting. The results (mAP@0.25) are reported in the **ScanNet val** set. **SDCoT+** refers to SDCoT with mixed pseudo labels proposed in Section III-D. **SDCoT++** refers to SDCoT with mixed pseudo labels and calibrated probabilities proposed in Section III-D and Section III-D, respectively.

	Method	Model	$\|C_{novel}\|$ = 9			$\|C_{novel}\|$ = 4			$\|C_{novel}\|$ = 1		
			Base	Novel	All	Base	Novel	All	Base	Novel	All
1	SDCoT [27]	VoteNet	53.75	54.91	54.33	49.50	70.85	54.25	52.01	31.71	50.89
2	SDCoT+	VoteNet	**53.97**	56.8	**55.38**	49.85	**71.96**	54.76	53.16	31.26	51.94
3	**SDCoT++**	VoteNet	53.41	**56.81**	55.11	**50.89**	71.18	**55.40**	**54.08**	**33.46**	**52.94**
4	SDCoT [27]	3DETR	66.79	61.25	64.02	59.96	71.97	63.22	65.45	**40.81**	64.08
5	SDCoT+	3DETR	68.88	61.60	65.21	**61.86**	71.42	63.99	66.14	38.29	64.59
6	**SDCoT++**	3DETR	**69.11**	**61.76**	**65.44**	61.57	**72.99**	**64.11**	**66.56**	38.25	**64.99**

TABLE VII: Ablation studies in the *sequential incremental* object detection performance (AP@0.25) of each class in the **SUN RGB-D val** set. The model sequentially learns 5 novel classes in 2 batches. B[1-5] represents base training on 5 base classes. +N[6,7,8] and +N[9, 10] denote incremental training on novel classes. B[1-10] represents Joint training on all classes. **SDCoT+** refers to SDCoT with mixed pseudo labels proposed in Section III-D. **SDCoT++** refers to SDCoT with mixed pseudo labels and calibrated probabilities proposed in Section III-D and Section III-D, respectively.

	Model	bath	bed	bookshelf	chair	desk	dresser	nightstand	sofa	table	toilet	mAP
B[1-5]	VoteNet	74.71	85.53	32.75	73.02	24.96	-	-	-	-	-	58.19
+ N[6,7,8] SDCoT [27]	VoteNet	**51.57**	84.04	23.83	62.83	**16.94**	**26.04**	57.34	59.75	-	-	47.79
+ N[6,7,8] SDCoT+	VoteNet	48.36	82.16	22.76	67.04	17.48	24.34	60.13	61.41	-	-	47.96
+ N[6,7,8] SDCoT++	VoteNet	49.67	**84.52**	**29.46**	**65.87**	15.27	25.99	**59.99**	**61.62**	-	-	**49.05**
+ N[9, 10] SDCoT [27]	VoteNet	36.59	79.60	10.35	60.12	**15.16**	12.80	35.15	56.51	**46.95**	88.08	44.13
+ N[9, 10] SDCoT+	VoteNet	38.57	82.35	13.62	57.24	14.84	14.79	41.86	54.35	48.82	86.74	45.33
+ N[9, 10] SDCoT++	VoteNet	**38.91**	**82.47**	**13.75**	**61.34**	14.28	**15.35**	**38.59**	**59.26**	44.93	**89.32**	**45.82**
B[1-10]	VoteNet	78.49	84.31	32.62	73.73	25.44	30.90	58.11	64.15	50.48	90.36	58.86
B[1-5]	3DETR	73.97	84.71	30.19	75.09	23.93	-	-	-	-	-	57.58
+ N[6,7,8] SDCoT [27]	3DETR	71.01	**82.44**	**28.28**	**73.38**	21.94	30.99	59.75	64.38	-	-	54.02
+ N[6,7,8] SDCoT+	3DETR	70.15	82.81	28.82	70.83	21.18	32.79	61.26	63.93	-	-	53.97
+ N[6,7,8] SDCoT++	3DETR	**71.93**	82.34	28.06	72.50	**24.04**	**31.26**	**60.81**	**65.40**	-	-	**54.54**
+ N[9, 10] SDCoT [27]	3DETR	46.61	80.89	15.29	61.41	15.04	**14.51**	37.71	59.22	46.57	89.17	46.64
+ N[9, 10] SDCoT+	3DETR	48.54	82.62	15.33	62.27	18.39	14.87	37.72	58.23	49.12	88.16	47.53
+ N[9, 10] SDCoT++	3DETR	**48.78**	**83.84**	**17.59**	**62.81**	**17.91**	14.09	**38.64**	**59.45**	**47.20**	**89.39**	**47.97**
B[1-10]	3DETR	69.80	84.60	28.50	72.40	34.30	29.60	61.40	65.30	52.60	91.00	58.95

TABLE VIII: Impacts of distillation targets on object detection performance of SDCoT++. The results are reported on SUN RGB-D dataset with $|C_{novel}| = 5$.

Class	Center	Size	Base	Novel	All
			50.95	60.41	55.68
✓	✓		53.01	61.29	57.15
✓		✓	53.55	61.25	57.40
✓	✓	✓	53.19	61.33	57.26
✓			**53.95**	**61.78**	**57.87**

TABLE IX: Impact of objectness threshold τ_o on incremental learning performance mAP@0.25 (ScanNet val, VoteNet, $|C_{novel}| = 9$, $\tau_c = 0.9$).

τ_o	**Base**	**Novel**	**All**
0.80	40.81	49.37	45.09
0.85	44.23	51.94	48.09
0.90	49.20	55.13	50.17
0.95	**53.41**	**56.81**	**55.11**

values (including *center* and *size*). Additionally, we examine alternative loss functions and assess the influence of these variations on the performance of the model. We adopt VoteNet as our backbone for the study.

Distillation Targets. Table VIII presents the outcomes of utilizing various distillation targets for calculating the final distillation loss. Specifically, we compute the mean square error between the outputs related to size and center from Φ_B and $\Phi_{B\cup N}$, in addition to the class-aware distillation loss we originally employed. The table illustrates that class-aware distillation effectively enhances the detection performance by preserving base class knowledge, while size-aware and center-aware distillation do not enhance the extraction of valuable

TABLE X: Impact of classification probability threshold τ_c on incremental learning performance mAP@0.25 (ScanNet val, VoteNet, $|C_{novel}| = 9$, $\tau_o = 0.95$).

τ_c	Base	Novel	All
0.70	49.50	47.44	48.47
0.75	49.59	47.66	48.63
0.80	50.44	50.38	50.41
0.85	52.06	52.41	52.24
0.90	**53.41**	**56.81**	**55.11**
0.95	50.99	50.74	50.87

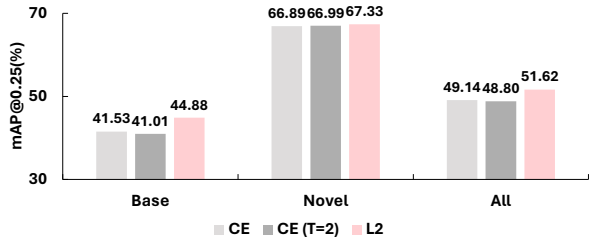

Fig. 5: Impacts of different distillation loss functions on object detection performance of SDCoT++. The results are reported on SUN RGB-D dataset with $|C_{novel}| = 3$. **CE**: The cross entropy loss. **CE (T=2)**: Cross entropy with temperature $T = 2$. **L2**: The mean square error.

information from previous knowledge. In some instances, they even slightly detract from the performance on the base classes within the specified scenario. Therefore, our strategy focuses solely on distilling knowledge from the classification logits.

Distillation Losses. To assess the impact of various loss functions, we substitute the L2 norm loss in Eq. 4 with a cross-entropy loss and a knowledge distillation loss (a cross-entropy loss with temperature) [62]. According to Figure 5, the L2 norm loss emerges as the more effective option for class-incremental 3D object detection, suggesting that it better facilitates the retention of previously learned knowledge.

G. Applicability to the Outdoor Dataset

The experimental results for the outdoor KITTI dataset are shown in Table XI. Both SDCoT and SDCoT++ significantly outperform the fine-tuning baseline, demonstrating the effectiveness of our method in the outdoor scenario. SDCoT++, outperforms SDCoT [27], achieving better performance across the base, novel, and all categories. Specifically, SDCoT++ attains higher mAP on base classes and overall than SDCoT. The improvement of SDCoT++ over SDCoT indicates the applicability of mixed pseudo labels and probability calibration even with a smaller number of base classes (3) and varied class-wise object occurrences in the dataset, as detailed in Table III of the supplementary material. However, there remains a noticeable gap between the upper-bound performance (joint training) and our methods, suggesting room for further enhancement in adapting our approach to outdoor scenarios.

V. Conclusion

This research addresses the novel and practical challenge of class-incremental 3D object detection. We introduced SDCoT++ that leverages static-dynamic co-teaching via enhancing pseudo labels to incrementally learn new object classes without revisiting past training data and validated the efficacy of our SDCoT++ across various class-incremental 3D object detection scenarios using the SUN RGB-D and ScanNet datasets. Our SDCoT++ significantly mitigates the issue of catastrophic forgetting and enhances the ability of the model to adapt to new classes. Our findings aim to inspire further investigations into this pertinent area.

TABLE XI: Batch incremental 3D object detection performance mAP on the KITTI dataset *val* split.

Method	Base	Novel	All
Base training	81.49	-	-
Fine-tuning	5.49	25.75	13.60
SDCoT [27]	61.27	22.50	45.77
SDCoT++	**62.56**	**22.52**	**46.55**
Joint training	80.55	32.74	61.43

VI. Acknowledgement

This extension is primarily conducted at SUTD and A*STAR, with primary support from the Agency for Science, Technology and Research (A*STAR) under its MTC Programmatic Funds (Grant No. M23L7b0021). It is partially supported by the National Research Foundation Singapore and DSO National Laboratories under its AI Singapore Programme (AISG Award No: AISG2-RP-2020-016), and the Tier 2 grant MOET2EP20120-0011 from the Singapore Ministry of Education.

References

[1] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3d object detection in point clouds," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[2] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2906–2917.

[3] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3d object detection with pointformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 7463–7472.

[4] Z. Liu, Z. Zhang, Y. Cao, H. Hu, and X. Tong, "Group-free 3d object detection via transformers," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2929–2938.

[5] D. Rukhovich, A. Vorontsova, and A. Konushin, "Fcaf3d: Fully convolutional anchor-free 3d object detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 477–493.

[6] Y. Shen, Z. Geng, Y. Yuan, Y. Lin, Z. Liu, C. Wang, H. Hu, N. Zheng, and B. Guo, "V-DETR: DETR with vertex relative position encoding for 3d object detection," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=NDkpxG94sF

[7] J. Guo, X. Xing, W. Quan, D.-M. Yan, Q. Gu, Y. Liu, and X. Zhang, "Efficient center voting for object detection and 6d pose estimation in 3d point cloud," *IEEE Transactions on Image Processing*, vol. 30, pp. 5072–5084, 2021.

[8] X. He, Z. Wang, J. Lin, K. Nai, J. Yuan, and Z. Li, "Do-sa&r: Distant object augmented set abstraction and regression for point-based 3d object detection," *IEEE Transactions on Image Processing*, vol. 32, pp. 5852–5864, 2023.

[9] Q. Cai, Y. Pan, T. Yao, and T. Mei, "3d cascade rcnn: High quality object detection in point clouds," *IEEE Transactions on Image Processing*, vol. 31, pp. 5706–5719, 2022.

[10] M. Feng, S. Z. Gilani, Y. Wang, L. Zhang, and A. Mian, "Relation graph network for 3d object detection in point clouds," *IEEE Transactions on Image Processing*, vol. 30, pp. 92–107, 2021.

[11] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[12] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[13] Y. Liu, B. Schiele, and Q. Sun, "Rmm: Reinforced memory management for class-incremental learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3478–3490, 2021.

[14] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to prompt for continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 139–149.

[15] D.-W. Zhou, F.-Y. Wang, H.-J. Ye, L. Ma, S. Pu, and D.-C. Zhan, "Forward compatible few-shot class-incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 9046–9056.

[16] M. Kang, J. Park, and B. Han, "Class-incremental learning by knowledge distillation with adaptive feature consolidation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 16 071–16 080.

[17] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental learning of object detectors without catastrophic forgetting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3400–3409.

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

[19] K. J. Joseph, J. Rajasegaran, S. Khan, F. S. Khan, and V. N. Balasubramanian, "Incremental object detection via meta-learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 9209–9216, 2022.

[20] T. Feng, M. Wang, and H. Yuan, "Overcoming catastrophic forgetting in incremental object detection via elastic response distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9427–9436.

[21] L. Wang, X. Zhang, K. Yang, L. Yu, C. Li, L. HONG, S. Zhang, Z. Li, Y. Zhong, and J. Zhu, "Memory replay with data compression for continual learning," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=a7H7OucbWaU

[22] N. Dong, Y. Zhang, M. Ding, and G. H. Lee, "Incremental-detr: Incremental few-shot object detection via self-supervised learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, 2023, pp. 543–551.

[23] H. Wang, L. Ding, S. Dong, S. Shi, A. Li, J. Li, Z. Li, and L. Wang, "Cagroup3d: Class-aware grouping for 3d object detection on point clouds," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 29 975–29 988.

[24] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 567–576.

[25] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.

[26] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[27] N. Zhao and G. H. Lee, "Static-dynamic co-teaching for class-incremental 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 3436–3445.

[28] Y. Zhang, J. Lu, and J. Zhou, "Objects are different: Flexible monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 3289–3298.

[29] X. Shi, Q. Ye, X. Chen, C. Chen, Z. Chen, and T.-K. Kim, "Geometry-based distance decomposition for monocular 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 15 172–15 181.

[30] T. Wang, X. Zhu, J. Pang, and D. Lin, "Fcos3d: Fully convolutional one-stage monocular 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, October 2021, pp. 913–922.

[31] D. Rukhovich, A. Vorontsova, and A. Konushin, "Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2397–2406.

[32] K.-C. Huang, T.-H. Wu, H.-T. Su, and W. H. Hsu, "Monodtr: Monocular 3d object detection with depth-aware transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 4012–4021.

[33] D.-P. Fan, Z. Lin, Z. Zhang, M. Zhu, and M.-M. Cheng, "Rethinking rgb-d salient object detection: Models, data sets, and large-scale benchmarks," *IEEE Transactions on neural networks and learning systems*, vol. 32, no. 5, pp. 2075–2089, 2020.

[34] R. Cong, Q. Lin, C. Zhang, C. Li, X. Cao, Q. Huang, and Y. Zhao, "Cir-net: Cross-modality interaction and refinement for rgb-d salient object detection," *IEEE Transactions on Image Processing*, vol. 31, pp. 6800–6815, 2022.

[35] W. Ji, J. Li, S. Yu, M. Zhang, Y. Piao, S. Yao, Q. Bi, K. Ma, Y. Zheng, H. Lu *et al.*, "Calibrated rgb-d salient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9471–9481.

[36] N. Huang, Q. Jiao, Q. Zhang, and J. Han, "Middle-level feature fusion for lightweight rgb-d salient object detection," *IEEE Transactions on Image processing*, vol. 31, pp. 6621–6634, 2022.

[37] Z. Wu, G. Allibert, F. Meriaudeau, C. Ma, and C. Demonceaux, "Hidanet: Rgb-d salient object detection via hierarchical depth awareness," *IEEE Transactions on Image Processing*, vol. 32, pp. 2160–2173, 2023.

[38] X. Qian, L. Yu-kun, W. Jing, W. Zhoutao, Z. Yiming, X. Kai, and W. Jun, "Mlcvnet: Multi-level context votenet for 3d object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[39] Z. Zhang, B. Sun, H. Yang, and Q. Huang, "H3dnet: 3d object detection using hybrid geometric primitives," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*. Springer, 2020, pp. 311–329.

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

[41] D. Rukhovich, A. Vorontsova, and A. Konushin, "Tr3d: Towards real-time indoor 3d object detection," in *2023 IEEE International Conference on Image Processing (ICIP)*, 2023, pp. 281–285.

[42] Y. Zhu, L. Hui, Y. Shen, and J. Xie, "Spgroup3d: Superpoint grouping network for indoor 3d object detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, pp. 7811–7819, Mar. 2024. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/28616

[43] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[44] Z. Wang, Y.-L. Li, X. Chen, H. Zhao, and S. Wang, "Uni3detr: Unified 3d detection transformer," *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[45] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, "Class-incremental learning: Survey and performance evaluation on image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5513–5533, 2023.

[46] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[47] K. Li, J. Wan, and S. Yu, "Ckdf: Cascaded knowledge distillation framework for robust incremental learning," *IEEE Transactions on Image Processing*, vol. 31, pp. 3825–3837, 2022.

[48] Z. Ji, Z. Hou, X. Liu, Y. Pang, and X. Li, "Memorizing complementation network for few-shot class-incremental learning," *IEEE Transactions on Image Processing*, vol. 32, pp. 937–948, 2023.

[49] Z. Ji, J. Li, Q. Wang, and Z. Zhang, "Complementary calibration: Boosting general continual learning with collaborative distillation and self-supervision," *IEEE Transactions on Image Processing*, vol. 32, pp. 657–667, 2023.

[50] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, and J. Choi, "Rainbow memory: Continual learning with a memory of diverse samples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8218–8227.

[51] M. De Lange and T. Tuytelaars, "Continual prototype evolution: Learning online from non-stationary data streams," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 8250–8259.

[52] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Advances in Neural Information Processing Systems*, 2017, pp. 2990–2999.

[53] L. Wang, K. Yang, C. Li, L. Hong, Z. Li, and J. Zhu, "Ordisco: Effective and efficient usage of incremental unlabeled data for semi-supervised continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5383–5392.

[54] D.-W. Zhou, Q.-W. Wang, H.-J. Ye, and D.-C. Zhan, "A model or 603 exemplars: Towards memory-efficient class-incremental learning," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=S07feAlQHgM

[55] H. Zhao, H. Wang, Y. Fu, F. Wu, and X. Li, "Memory-efficient class-incremental learning for image classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 10, pp. 5966–5977, 2021.

[56] X. Li, Y. Zhou, T. Wu, R. Socher, and C. Xiong, "Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3925–3934.

[57] S. Yan, J. Xie, and X. He, "Der: Dynamically expandable representation for class incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3014–3023.

[58] F.-Y. Wang, D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Foster: Feature boosting and compression for class-incremental learning," in *European conference on computer vision*. Springer, 2022, pp. 398–414.

[59] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

[60] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy *et al.*, "Dualprompt: Complementary prompting for rehearsal-free continual learning," in *European Conference on Computer Vision*. Springer, 2022, pp. 631–648.

[61] A. Douillard, A. Ramé, G. Couairon, and M. Cord, "Dytox: Transformers for continual learning with dynamic token expansion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9285–9295.

[62] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[63] Z. Yang, C. Zhang, R. Li, Y. Xu, and G. Lin, "Efficient few-shot object detection via knowledge inheritance," *IEEE Transactions on Image Processing*, vol. 32, pp. 321–334, 2023.

[64] Y. Liu, B. Schiele, A. Vedaldi, and C. Rupprecht, "Continual detection transformer for incremental object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 23 799–23 808.

[65] Z. Cheng, C. Chen, Z. Zhao, P. Qian, X. Li, and X. Yang, "Coco-teach: A contrastive co-teaching network for incremental 3d object detection," in *2023 IEEE International Conference on Image Processing (ICIP)*, 2023, pp. 1990–1994.

[66] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[67] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 408–18 419, 2021.

[68] Y. Wang, H. Chen, Q. Heng, W. Hou, Y. Fan, , Z. Wu, J. Wang, M. Savvides, T. Shinozaki, B. Raj, B. Schiele, and X. Xie, "Freematch: Self-adaptive thresholding for semi-supervised learning," *International Conference on Learning Representations (ICLR)*, 2023.

[69] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1607–1616.

[70] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in neural information processing systems*, 2017, pp. 1195–1204.

[71] N. Zhao, T.-S. Chua, and G. H. Lee, "Sess: Self-ensembling semi-supervised 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 079–11 087.

[72] J. Ren, C. Yu, s. sheng, X. Ma, H. Zhao, S. Yi, and h. Li, "Balanced meta-softmax for long-tailed visual recognition," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 4175–4186.

[73] H. Wang, L. Liu, B. Zhang, J. Zhang, W. Zhang, Z. Gan, Y. Wang, C. Wang, and H. Wang, "Calibrated teacher for sparsely annotated object detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, pp. 2519–2527, Jun. 2023. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/25349

[74] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 529–10 538.

[75] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7