DA-CIL: Towards Domain Adaptive Class-Incremental 3D Object Detection

Ziyuan Zhao\textsuperscript{1,2,*}
zhaoy_ziyuan@i2r.a-star.edu.sg
Mingxi Xu\textsuperscript{1,3}
XU0002XI@e.ntu.edu.sg
Peisheng Qian\textsuperscript{1}
qian_peisehmg@i2r.a-star.edu.sg
Ramanpreet Singh Pahwa\textsuperscript{1,2}
ramanpreet_pahwa@i2r.a-star.edu.sg
Richard Chang\textsuperscript{1}
richard_chang@i2r.a-star.edu.sg

\textsuperscript{1}Institute for Infocomm Research (I\textsuperscript{2}R),
Agency for Science, Technology and Research (A*STAR), Singapore
\textsuperscript{2}Artificial Intelligence, Analytics And Informatics (AI\textsuperscript{3}),
Agency for Science, Technology and Research (A*STAR), Singapore
\textsuperscript{3}Nanyang Technological University, Singapore

Abstract

Deep learning has achieved notable success in 3D object detection with the advent of large-scale point cloud datasets. However, severe performance degradation in the past trained classes, \textit{i.e.}, catastrophic forgetting, still remains a critical issue for real-world deployment when the number of classes is unknown or may vary. Moreover, existing 3D class-incremental detection methods are developed for the single-domain scenario, which fail when encountering domain shift caused by different datasets, varying environments, etc. In this paper, we identify the unexplored yet valuable scenario, \textit{i.e.}, class-incremental learning under domain shift, and propose a novel 3D domain adaptive class-incremental object detection framework, DA-CIL, in which we design a novel dual-domain copy-paste augmentation method to construct multiple augmented domains for diversifying training distributions, thereby facilitating gradual domain adaptation. Then, multi-level consistency is explored to facilitate dual-teacher knowledge distillation from different domains for domain adaptive class-incremental learning. Extensive experiments on various datasets demonstrate the effectiveness of the proposed method over baselines in the domain adaptive class-incremental learning scenario.

1 Introduction

With the widespread use of 3D point clouds, deep learning-based 3D object detection has received considerable attention. Many efforts \cite{17, 18, 19, 20, 21, 22} have been devoted to the field, achieving remarkable success in object recognition and localization from 3D point clouds or voxelized data. Nevertheless, the \textit{catastrophic forgetting} problem seriously limits the deployment of existing detection models in dynamic real-world environments, where novel classes are encountered over time. In the class-incremental scenario, the performance of existing models on old classes tends to decrease substantially when trained on...
novel classes. In recent years, class-incremental learning has been extensively investigated for various 2D vision tasks from different perspectives, including regularization-based methods [10], replay-based methods [13, 24] and parameter-isolation methods [15, 25]. However, class-incremental learning (CIL), particularly class-incremental 3D object detection remains underexplored [6, 33].

On the other hand, deep learning models are also required to rapidly adapt to varying environmental conditions, such as locations, surroundings, and weather. Existing class-incremental 3D object detection methods assume that the data is from the same domain with the same distribution. In the presence of domain shift, deep learning models trained on one domain (i.e., source domain) always suffer from tremendous performance degradation when evaluated on another domain (i.e., target domain). Moreover, multiple datasets are encouraged to be used to alleviate data scarcity in a cross-domain scenario, inducing the notorious domain gap across datasets by various factors, such as geometric mismatch (Fig. 1). A series of unsupervised domain adaptation (UDA) approaches have emerged to address the domain shift in various computer vision tasks [2, 4, 9, 28]. However, these methods are not tailored to meet the needs of class-incremental object detection from 3D point clouds.

In summary, existing CIL and UDA methods are designed for addressing different challenges separately, ignoring that the catastrophic forgetting and domain shift problems concurrently exist in 3D object detection on point clouds of real-world environments. This motivates us to investigate a practical yet challenging class-incremental scenario, i.e., class-incremental learning under domain shift, and introduce a new CIL paradigm called domain adaptive class-incremental learning (DA-CIL). To the best of our knowledge, we are the first to study domain adaptation in class-incremental learning for 3D object detection. As shown in Fig. 2, we aim to leverage labeled old classes on the source domain and labeled new classes on the target domain to close the domain shift and adapt to new classes without forgetting old classes on the target domain in DA-CIL. We propose a novel DA-CIL framework for 3D object detection from point clouds. First, we propose a dual-domain copy-paste (DuDo-CP) augmentation method to generate cross-domain point clouds and in-domain point clouds for ground-truth augmentation and gradual domain adaptation to narrow the distribution shift across domains. Second, we employ a dual-teacher network to facilitate knowledge transfer from different domains. The cross-domain teacher trained with cross-domain point clouds is used to generate pseudo labels of old classes in the target domain and distill cross-domain knowledge of old classes to the student model. The in-domain teacher based on self-ensembling [27] is used to transfer in-domain knowledge of new classes to the student model. Third, we construct multi-level consistency (MLC) regularization from different aspects to better facilitate the dual-domain teacher-student training process. Our ap-
Figure 2: The difference between CIL and DA-CIL. (a) CIL is designed for adapting new classes without forgetting old classes in the same domain. (b) Domain shift always exists between source and target domains. (c) DA-CIL is designed to close the domain shift and adapt new classes across different domains.

Our main contributions can be summarized as follows:

• We identify a new CIL scenario where domain shift occurs when adapting new classes across domains, and formulate a new CIL paradigm to enable domain adaptive class-incremental learning for 3D object detection.

• Extensive experiments and analysis demonstrate that our approach can achieve superior performance over existing methods under different incremental learning scenarios.

2 Related Work

Point cloud object detection. 3D object detection from point clouds has drawn great attention in recent years. PointNet [20] is a pioneering work to detect 3D objects using only point cloud data, which extracts global information and point-wise features for 3D object detection. PointNet++ [22] advances PointNet by extracting local information in neighborhoods of individual points at multiple scales. More recently, Qi et al. propose a voting-based framework, VoteNet [21], in which high-quality object bounding boxes are generated by directly voting for object centers, achieving promising performance on various indoor benchmarks such as ScanNet [3] with geometric information alone. Therefore, we use the modified VoteNet [33] as our detection backbone.

Class-incremental learning. Class-incremental learning aims to alleviate the catastrophic forgetting effect and enable deep learning models to continuously learn new classes while preserving the learned knowledge from old classes. Most of the existing class-incremental learning methods were developed for 2D planar datasets, which can be divided into three categories [5]: 1) Regularization-based methods use knowledge distillation [8] to enforce consistency between either the data (e.g., rebalancing [10]) or parameters (e.g., EWC [12]). 2) Replay-based methods such as iCaRL [24] and GEM [13] store exemplars from old classes.
Figure 3: Overview of the proposed DA-CIL framework: First, cross-domain intermediate point clouds and in-domain augmented point clouds are generated by dual-domain copy-paste augmentation. Then, a dual-teacher network is constructed to distill knowledge from these domains for domain adaptive class-incremental learning.

in a replay buffer or reproduce samples of previous tasks with a trained generator. 3) Parameter isolation methods can add new branches for new classes while freezing model parameters learned from old classes, for instance, PackNet[15] and HAT [25]. In comparison, only a few methods explore 3D class-incremental scenarios, especially for class-incremental 3D object detection [6, 33]. For example, SDCoT [33] is a co-teaching method that alleviates catastrophic forgetting of old classes via a static teacher, and consistently learns the underlying knowledge from new data via a dynamic teacher. However, these methods focus on the single-domain scenario and cannot handle the domain shift issue.

Unsupervised domain adaptation. Domain shift has been a long-standing problem for computer vision tasks, especially in the absence of target labels [29]. Considerable efforts have been devoted to studying unsupervised domain adaptation (UDA) for minimizing the cross-domain discrepancy from different perspectives, including adversarial learning [9, 11, 28], self-ensemble learning [14, 35, 36], and self-training [31, 37]. Nevertheless, limited studies have been proposed for UDA in 3D object detection. Wang et al. [30] proposed to align the object size across domains to close the size-level domain gap. Luo et al. [14] proposed a Multi-Level Consistency Network (MLC-Net) based on the mean teacher paradigm [27] for addressing the geometric-level domain gap. These UDA methods require samples from both source and target domains for domain alignment, thereby having limited use on data-scarce scenarios. Besides, the same classes are required for both domains in conventional UDA, limiting the practical use for class-incremental scenarios. Recent works [4, 23] suggest that augmentation techniques [7, 16, 32] can not only relieve data scarcity but also remedy data-level gaps across domains. Inspired by the success of Copy-Paste augmentation [7], we intend to advance it on cross-domain scenarios to diversify training data distributions for domain adaptive class-incremental learning.

3 Methodology

Let $C = C_b \cup C_n$ denote the whole set of classes, where $C_b$ is the set of base (old) classes and $C_n$ is the disjoint set of novel (new) classes. In the conventional class-incremental object detection, $C_b$ and $C_n$ have great amounts of training samples from the same domain with annotated object classes and bounding boxes. Differently, in our DA-CIL setting, $C_b$ and $C_n$
Figure 4: An illustration of Dual-Domain Copy-Paste, which first extracts objects from the source domain to populate the ground-truth database. Then objects are copy-pasted into point clouds from the source domain (In-domain CP) and the target domain (Cross-domain CP) to generate the augmented domain and the intermediate domain, respectively.

are from different domains with distribution shift. We define the set of base classes from the source domain as $C^s_b$, and the set of novel classes from the target domain as $C^t_n$. Given a well-trained base model $\Phi_B$ on $C^s_b$, we aim to learn a domain adaptive class-incremental model $\Phi_{B\cup N}$ using $C^t_n$, which can detect both base classes $C^s_b$ and novel classes $C^t_n$ in the target domain. In this regard, we propose a unified DA-CIL framework to address both catastrophic forgetting and domain shift problems simultaneously.

### 3.1 DA-CIL Architecture

Fig. 3 depicts an overview of the proposed DA-CIL architecture. In DA-CIL, two teacher-student networks are constructed for dual-domain knowledge distillation to achieve class-incremental learning and domain adaptation. More specifically, we first generate cross-domain intermediate point clouds and in-domain augmented point clouds using the proposed dual-domain copy-paste (DuDo-CP) augmentation (see Sec. 3.2) to diversify training distributions, thereby improving the domain adaptive class-incremental performance. Then, we construct a dual-teacher network, which consists of one student model $\Phi_{B\cup N}$, one cross-domain teacher $\Phi_B$, and one in-domain teacher $\Phi^t_{B\cup N}$. The well-trained cross-domain teacher on intermediate point clouds is applied to regularize the student model for closing domain shift and generate pseudo labels for mixed label creation. At the same time, the transformed and original in-domain augmented point clouds are passed to the student model and in-domain teacher model, respectively, for multi-level consistency learning (see Sec. 3.3). It is noted that the in-domain teacher is updated with the exponential moving average (EMA) weights of the student model, i.e., $\Phi^t_{B\cup N} = \alpha \Phi_{B\cup N}^{t-1} + (1 - \alpha) \Phi_{B\cup N}$, where $t$ is the training iteration and $\alpha$ is the EMA decay rate. Finally, the student model is trained under mixed label supervision with dual-teacher knowledge transfer to achieve domain adaptive class-incremental learning (see Sec. 3.4).

### 3.2 Dual-Domain Copy-Paste

To relieve data scarcity and reduce the domain gap at the data level, we extensively leverage copy-paste (CP) augmentation techniques for creating cross-domain and in-domain point
clouds. As shown in Fig. 4, we generate a ground-truth database containing objects in $C_b$, and randomly select objects in the ground-truth database to paste in point clouds from source and target domains, forming in-domain source point clouds $X_{in(source)}$ and cross-domain point clouds $X_{cross}$, respectively. Similarly, $X_{in(target)}$ can be constructed by pasting $C_n$ objects into point clouds in the target domain.

**Cross-domain CP.** To mitigate the serious domain shift between the source domain $X_s$ and the target domain $X_t$ caused by point cloud geometric mismatch (see Fig. 1), we construct the intermediate domain $X_{cross}$. More specifically, we explicitly model $X_{cross}$ that includes source objects in the target context. In $X_{cross}$, the distribution of context around source objects becomes consistent with $X_t$. Meanwhile, local geometries of objects remain unchanged. In $X_{cross}$, data distribution has shifted from $X_s$ to the combination of source objects in target surroundings, which is between $X_t$ and $X_s$ as shown in Fig. 2. Therefore, $X_{cross}$ serves as a bridge between $X_s$ and $X_t$ to facilitate the gradual transfer of knowledge [4]. Let $D_s^n \in X_s$, $(i = 1, 2, 3, \ldots)$ denotes objects in the source domain $X_s$. We copy-paste $D_s^n$ into $X_t$ and construct an intermediate domain $X_{cross}$, formulated as:

$$X_{cross} = X_t + \sum_{n=1}^{N} T(D_s^n),$$

where $N$ refers to the number of objects, and $T$ refers to object transformations, including random size and orientation augmentations. During training, $X_{cross}$ is first created to fine-tune the base model pre-trained on $X_s$. Subsequently, $X_{cross}$ is fed to the cross-domain teacher and the student in Fig. 3 for distillation of knowledge in old classes.

**In-domain CP.** To tackle data scarcity in $X_s$ and $X_t$, we perform in-domain copy-paste, which diversifies object appearances and their surrounding environment. Specifically, we create an augmented source domain $X_{in(source)}$ and an augmented target domain $X_{in(target)}$, calculated as:

$$X_{in(source)} = X_s + \sum_{n=1}^{N} T(D_s^n); \quad X_{in(target)} = X_t + \sum_{n=1}^{N} T(D_t^n),$$

where $D_t^n \in X_t$, $(i = 1, 2, 3, \ldots)$ denotes objects in the target domain. The approach complements the cross-domain copy-paste and enhances the model performance in individual domains. During training, $X_{in(source)}$, and $X_{in(target)}$ are constructed for base model pre-training and dual-teacher knowledge distillation, respectively.

### 3.3 Multi-Level Consistency

To facilitate the dual-teacher knowledge transfer, we propose multi-level consistency regularization from two aspects. First, we propose statistics-level consistency to tackle the mismatch in batch normalization statistics. Second, bounding box-level consistency (i.e., center-level, class-level and size-level consistency [34]) is applied for intra-domain knowledge distillation.

**Statistics-Level Consistency.** In our architecture, the inputs of the student and teacher models are different, resulting in batch statistics mismatch and an unstable training process. The mismatched batch statistics, such as parameters in batch normalization (BN) layers (mean $\mu$ and variance $\sigma$) could degrade the model performance [1]. In this regard, we propose to share the BN parameters of the in-domain teacher with the cross-domain teacher and the student network for closing the distribution gap caused by statistics mismatch and stabilizing the training process.
Bounding Box-Level Consistency. Let \( B_s = \{ b_1^s, b_2^s, \ldots, b_N^s \} \) and \( B_{in} = \{ b_1^d, b_2^d, \ldots, b_M^d \} \) be the proposals generated by the student and the in-domain teacher, respectively, where \( N \) and \( M \) refer to the respective number of proposals from student and in-domain teacher. Our bounding box-level consistency includes 3 components: 1) **center-level consistency loss** calculates the distance between centers of bounding boxes proposed by the student and the in-domain teacher, shown in Eq. 3, 2) **class-level consistency loss** computes the Kullback-Leibler (KL) Divergence between class proposals from the student and the in-domain teacher, as shown in Eq. 4, and 3) **size-level consistency loss** is the mean squared error (MSE) of widths, lengths, and heights of paired bounding boxes calculated in Eq. 5,

\[
L_{\text{center}} = \text{dist}(\text{Pair}_N) + \text{dist}(\text{Pair}_M),
\]

\[
L_{\text{class}} = \frac{1}{M} \sum_{i=1}^{M} D_{KL}(p_i^s, p_i^d), \quad (b_i^s, b_i^d) \in \text{Pair}_M, (6)
\]

\[
L_{\text{size}} = \frac{1}{M} \sum_{i=1}^{M} \text{MSE}(s_i^s, s_i^d),
\]

where \( \text{dist} \) refers to the square of the Euclidean distance between the centers. \( \text{Pair}_M \) is a paired bounding box set. Each pair contains an in-domain teacher proposal \( b_i^d \) and the closest student proposal \( b_i^s \) to \( b_i^d \) based on the minimum Euclidean distance between their centers and same to \( \text{Pair}_N \) \([\text{3}4]\). \( p_i^s \) is the probability of \( b_i^s \) belonging to each class. \( s_i^s \) contains the length, width and height of \( b_i^s \), and the same setting applies for \( b_i^d \).

### 3.4 Dual-Teacher Training Strategies

We implement the modified VoteNet \([\text{3}3]\) as our detection backbone for both teacher and student models, in which the indices of sampled points and votes from the student model are stored and reused in the cross-domain teacher for class-incremental learning. The student model learns the underlying knowledge in base classes from a cross-domain teacher (i.e., a frozen model pre-trained on the intermediate domain \( X_{\text{cross}} \) and augmented source domain \( X_{\text{in(target)}} \)) via a distillation loss \( L_{\text{dis}} \), which is the square of Euclidean distance between the classification logits of different bounding boxes. Meanwhile, the EMA in-domain teacher helps student model capture structure and semantic invariant information in objects with a consistency loss \( L_{\text{con}} \), shown as:

\[
L_{\text{con}} = L_{\text{center}} + \lambda_{\text{class}} L_{\text{class}} + \lambda_{\text{size}} L_{\text{size}},
\]

where \( \lambda_{\text{class}} \) and \( \lambda_{\text{size}} \) are both set to 1, empirically.

The 3D bounding boxes generated by the cross-domain teacher for the base classes in the target domain \( X_{\text{in(target)}} \) are utilized as pseudo labels which can be combined with labels of the novel classes to form the mixed labels for supervised learning. More specifically, the mixed labels are transformed by the same augmentation step that is applied on \( X_{\text{in(target)}} \) to compute a supervised loss \( L_{\text{sup}} \), following multi-task loss in VoteNet \([\text{2}1]\). The final loss \( L \) used for updating the student model is calculated as:

\[
L = \lambda_{\text{sup}} L_{\text{sup}} + \lambda_{\text{dis}} L_{\text{dis}} + \lambda_{\text{con}} L_{\text{con}},
\]

where \( \lambda_{\text{sup}}, \lambda_{\text{dis}}, \) and \( \lambda_{\text{con}} \) are set to 10, 1, and 10, empirically.
4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate our method on 3D object detection datasets, ScanNet [3] and SUN RGB-D [26]. ScanNet is an RGB-D video dataset containing 1,513 scans with object bounding boxes derived from point-level segmentation labels. SUN RGB-D is captured by 4 different sensors and contains 10,335 RGB-D images with 64,595 3D bounding boxes including orientations. To simulate the domain adaptive class-incremental scenario, ScanNet was employed as the source domain and 5 categories (bathtub, bed, bookshelf, chair, desk) were selected as base classes $C_b$. SUN RGB-D was employed as the target domain, in which $C_b$ refers to the same 5 base classes. 5 additional categories (dresser, nightstand, sofa, table, toilet) in SUN RGB-D were selected as novel classes in target domain, i.e., $C_n$. For evaluation, we adopted the common metric [21], mAP@IoU=0.25, i.e., mean average precision of predicted bounding boxes that overlap ground-truth with an amount of 0.25 or higher. Higher values indicate better performance.

4.2 Implementation Details

For DuDo-CP, we randomly pasted 1 or 2 $C_b$ object(s) to each scene in target domain $X_t$ to form cross-domain point clouds $X_{cross}$. The $C_b$ objects were randomly positioned on the left and right half along the x-axis of the point cloud to avoid collision. To address data scarcity, random scaling between (0.9, 1.1) and random rotation of ±10 degrees were applied to the objects. We first conducted base training on $C_b$ for 150 epochs and fine-tuned using $X_{in(source)}$ and $X_{cross}$ sequentially for 5 epochs each. We used the Adam optimizer with a batch size of 8 and an initial learning rate of 0.001, decayed at epoch 80 and 120 by 0.1. Then, we trained the dual-teacher network for 100 epochs. The teacher and student networks were initialized using the base-train model weights. Following [27], the EMA decay rate was set to 0.999. In each batch, 6 point clouds were selected from $X_{in(target)}$ and 2 point clouds were selected from $X_{cross}$ for training. The remaining settings were the same as base training.

4.3 Baselines

We compared our results with the following baselines. VoteNet [21] was used as the base model in all experiments. Following [33], we modified the VoteNet for aligning proposals from teacher and student models to facilitate class-incremental learning without leveraging color information. The model size is 10.9MB with 0.94M parameters. The inference time for a single 3D point cloud is around 0.2370s. It was trained on $C_b$ and $C_t$ to measure the domain gap in base classes. We then conducted joint training to evaluate the domain gap in all classes. For class-incremental learning, we evaluated 2 naive transfer learning baselines Freeze-and-add, and Fine-tune. To be specific, Freeze-and-add adds a new classifier for novel classes on the pre-trained base model with frozen weights. Fine-tune is similar to freeze-and-add except that all model parameters excluding the old classifier are not frozen. We implemented one recent popular self-ensembling method, MT [27] for unsupervised domain adaptation, and pseudo labels of base classes generated by the base model were introduced for adapting MT to class-incremental scenarios. We also used SDCoT [33], the co-teaching architecture as a strong baseline for class-incremental learning.
Table 1: 3D object detection performance mAP@0.25 on the SUN RGB-D validation set.

<table>
<thead>
<tr>
<th>Method</th>
<th>$C_b^s$</th>
<th>$C_b^t$</th>
<th>$C_n^t$</th>
<th>Base</th>
<th>Novel</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base only</td>
<td>✓</td>
<td>✓</td>
<td>√</td>
<td>26.60</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Base train</td>
<td>×</td>
<td>✓</td>
<td>√</td>
<td>57.63</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Class-Incremental Learning (CIL)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freeze-and-add</td>
<td>×</td>
<td>✓</td>
<td>√</td>
<td>54.24</td>
<td>10.61</td>
<td>32.42</td>
</tr>
<tr>
<td>Fine-tune</td>
<td>√</td>
<td>✓</td>
<td>✓</td>
<td>3.48</td>
<td>54.10</td>
<td>28.79</td>
</tr>
<tr>
<td>MT [27]</td>
<td>✓</td>
<td>√</td>
<td>✓</td>
<td>49.63</td>
<td>61.21</td>
<td>55.42</td>
</tr>
<tr>
<td>SDCoT [33]</td>
<td>✓</td>
<td>✓</td>
<td>√</td>
<td>52.04</td>
<td>59.48</td>
<td>55.76</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>53.81</td>
<td>59.50</td>
<td>56.66</td>
</tr>
<tr>
<td>CIL under Domain Shift</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freeze-and-add</td>
<td>✓</td>
<td>×</td>
<td>√</td>
<td>23.31</td>
<td>2.04</td>
<td>12.68</td>
</tr>
<tr>
<td>Fine-tune</td>
<td>✓</td>
<td>×</td>
<td>√</td>
<td>1.73</td>
<td>58.24</td>
<td>29.99</td>
</tr>
<tr>
<td>MT [27]</td>
<td>✓</td>
<td>×</td>
<td>√</td>
<td>33.84</td>
<td>59.03</td>
<td>46.43</td>
</tr>
<tr>
<td>SDCoT [33]</td>
<td>✓</td>
<td>×</td>
<td>√</td>
<td>29.41</td>
<td>57.74</td>
<td>43.57</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>36.41</td>
<td>58.80</td>
<td>47.60</td>
</tr>
<tr>
<td>Base &amp; Novel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joint train</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>6.80</td>
<td>52.52</td>
<td>29.66</td>
</tr>
<tr>
<td>Joint train</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>58.11</td>
<td>59.27</td>
<td>58.69</td>
</tr>
</tbody>
</table>

Table 2: Per-class object detection performance AP@0.25 on SUN RGB-D validation set, under the domain adaptive class incremental setting.

<table>
<thead>
<tr>
<th>Base classes</th>
<th>Novel classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>bathtub</td>
</tr>
<tr>
<td>SDCoT</td>
<td>41.82</td>
</tr>
<tr>
<td>Ours</td>
<td>52.62</td>
</tr>
</tbody>
</table>

4.4 Experimental Results

As shown in Table 1, the base train result with $C_b^s$ is significantly worse than that with $C_b^t$, which confirms the severity of the domain gap. SDCoT cannot well handle the domain shift problem, while MT improves the results, which demonstrate that self-ensemble learning can help domain adaptation in the class-incremental scenarios. Next, our method has a slight edge over SDCoT in the class-incremental learning (CIL) scenario, which proves the effectiveness of our approach in a single domain. Furthermore, our method outperforms SDCoT by a clear margin in CIL under domain shift. There is a 4% performance improvement in all classes (7% on base classes), indicating the effectiveness of our DA-CIL framework. In addition, for the CIL scenario, mAP@0.5 is 30.36% and 32.16% for SDCoT and our method, respectively. For CIL under domain shift scenario, mAP@0.5 is 21.19% and 22.39% for SDCoT and our method, respectively. Besides, it is well noted that our results (36.41%) have greatly improved from the lower bound (26.60%) on base classes. Finally, the domain gap is confirmed between two results of joint training on $C_b^s \cup C_b^t$, and $C_b^s \cup C_n^t$ respectively. We further investigate model performance in each class under the domain adaptive class-incremental setting, as presented in Table 2. Our method achieves better results in most classes, especially for large objects such as bathtubs and beds. They have relatively fixed contexts, which can benefit more from the augmented object-context combinations. On the other hand, objects like chairs receive slight performance degradation because they are small and can be erroneously copy-pasted with other objects. Visualizations of object detection results on SUN RGB-D validation data are provided in the supplementary material.

4.5 Ablation Study

To evaluate the effectiveness of the proposed augmentation method, we implemented two popular augmentation techniques for comparison, i.e., Mix3D [16] and CutMix [32]. Mix3D
combines 2 point clouds into a mixed point cloud via concatenation, which results in excessive overlaps in the mixed point clouds. CutMix randomly replaces patches of point clouds with patches from other point clouds, which unnecessarily introduces extra context from the source domain. As reported in Table 3, our method outperforms the two augmentation techniques. To evaluate the influence of the number of augmented data, we reported the model performance with different maximum numbers of augmented objects for cross-domain copy-paste, i.e., the mAP@0.25 scores are 47.60% (1-2 objects), 47.08% (1-3 objects), and 46.86% (1-4 object). We can see that our model is not sensitive to changes.

To evaluate the effectiveness of different components in our method for CIL under domain shift, we perform ablation studies by removing components in our approach. The results are demonstrated in Table 4. Without cross-domain copy-paste, the model achieves a lower result in base classes, which proves the effectiveness of cross-domain copy-paste augmentation in base class recognition. Similarly, it can be inferred from results without in-domain augmentation that in-domain copy-paste can enhance the model performance in novel classes. Moreover, statistics-level consistency provides around 0.8% performance increment in all classes. Besides, by adding in-domain CP in SDCoT, the mAP@0.25 is increased to 46.60%, and the score is increased to 46.86% by adding both in-domain CP and cross-domain CP, which demonstrate the effectiveness of augmentation methods against domain shift. In summary, the results indicate the effectiveness of each component in our architecture.

5 Conclusion

In this work, we identify and explore a novel domain adaptive class-incremental learning paradigm for 3D object detection. To achieve both incremental learning and domain adaptation, we propose a novel 3D object detection framework, DA-CIL, in which we design a novel dual-domain copy-paste augmentation method to address both in-domain data scarcity and cross-domain distribution shift. We further enhance the dual-teacher knowledge distillation with multi-level consistency between different domains. Extensive experimental results and analysis demonstrate the superior performance of our method over baselines under the challenging class-incremental scenario. In our future work, we will explore more scenarios with different domain shifts, such as geography-to-geography, day-night, and simulation-to-reality to further verify the effectiveness of the proposed method.

Acknowledgment

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AI3 HTPO Seed Fund no. C211118008.
References


