THE UNIVERSITY OF SYDNEY

MASTER THESIS

Cross-Domain Point Cloud Recognition with Deep Learning

Author:
Zicheng WANG

Supervisor:
A/Prof. Luping ZHOU

Co-Supervisor:
Dr. Dong YUAN

A thesis submitted in fulfillment of the requirements for the degree of Master of Philosophy

in the

School of Electrical and Information Engineering
Faculty of Engineering

June 1, 2023

Abstract of thesis entitled

Cross-Domain Point Cloud Recognition with Deep Learning

Submitted by

Zicheng WANG

for the degree of Master of Philosophy

at The University of Sydney

in June, 2023

Point cloud recognition using deep learning methods has attracted increasing research interest recently due to its great potential in real-world applications such as autonomous driving, robotics, etc. However, point clouds of similar objects often exhibit notable geometric variations due to the difference in capturing devices or environmental changes. This leads to significant performance degradation when a learnt point cloud recognition model is applied to a new scenario, which is also known as the domain adaptation issue.

In this thesis, we first provide a comprehensive literature review of deep learning on visual recognition, unsupervised domain adaptation, open-set unsupervised domain adaptation, self-supervised learning and knowledge transfer to introduce the background of the thesis. Then, an introduction to the problem setting and the commonly used benchmark datasets is provided for a better understanding of the task. Next, a point-level domain adaptive point sampling (DAPS) strategy is proposed to tackle the domain gap in cross-domain point cloud recognition. In addition, an instance-level domain adaptive cloud sampling (DACS) strategy is proposed to learn additional target-specific information for better recognition performance on the target domain. Moreover, we further propose a two-stage open-set domain adaptive sampling (OS-DAS)

strategy to learn an open-set recognition model in a coarse-to-fine manner to tackle the open-set unsupervised domain adaptation issue. Finally, we list some potential research directions for cross-domain point cloud recognition.

Cross-Domain Point Cloud Recognition with Deep Learning

by

Zicheng WANG
B.E. *Xidian University*

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy

at

University of Sydney
June, 2023

Declaration

I, Zicheng WANG, declare that this thesis titled, "Cross-Domain Point Cloud Recognition with Deep Learning", which is submitted in fulfilment of the requirements for the Degree of Master of Philosophy, represents my own work except where due acknowledgement has been made. I further declared that it has not been previously included in a thesis, dissertation, or report submitted to this University or to any other institution for a degree, diploma or other qualifications.

Author Name

Signed: _____

Date: June 1, 2023

For Mama and Papa

Acknowledgements

First, I would like to thank my supervisors Prof. Dong Xu and A/Prof. Luping Zhou for all the support in my research. With their patient guidance, I learnt many useful research techniques during my MPhil study period, which helped me to become an independent researcher. I would like to especially thank A/Prof. Luping Zhou, she is an excellent mentor, but also a respectable elder. It is her care and enlightenment that saved me in my darkest hour.

I would also sincerely thank A/Prof. Wen Li for his useful discussion during my MPhil study period. His valuable suggestions helped me gain a better understanding of my research area and solve many difficult problems in this area.

Next, I would like to sincerely thank my family. As a student who changed majors, it was their unconditional support that made me firm in my decision to leave my previous major, allowing me to really engage in the research I am interested in.

Furthermore, I wish to thank all my colleagues. They have provided great help in both my studies and my life during my MPhil study period. It is with their company that I am so determined to continue my research life.

In addition, I would like to thank my girlfriend who has not yet appeared, because I firmly believe that she is waiting for me in the near future, I can overcome all the difficulties in my study life.

Last but not least, I would also like to thank fate for allowing me to experience everything I have experienced, hone my will, exercise my ability, and let me understand the state of mind of the prodigal son, the determination to overcome obstacles, and the courage to remain calm. I am grateful for all the opportunities and challenges that have befallen me, and I look forward to the day when my life will be in harmony.

Zicheng WANG
University of Sydney
June 1, 2023

List of Publications

JOURNALS:

[1] **Zicheng Wang**, Wen Li, and Dong Xu, "Domain Adaptive Sampling for Cross-Domain Point Cloud Recognition", *IEEE Transactions on Circuits and Systems for Video Technology*. (Under review)

CONFERENCES:

[1] **Zicheng Wang**, Zhen Zhao, Xiaoxia Xing, Dong Xu, Xiangyu Kong and Luping Zhou, "Conflict-Based Cross-View Consistency for Semi-Supervised Semantic Segmentation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2023*. (Accepted)

[2] **Zicheng Wang** and Luping Zhou, "Perturbation is What You Need in Cross-Domain Point Cloud Recognition", in *Proceedings of the IEEE/CVF International Conference on Computer Vision 2023*. (Under review)

Contents

List of Figures

List of Tables

List of Abbreviations

DAS	**D**omain **A**daptive **S**ampling
DAPS	**D**omain **A**daptive **P**oint **S**ampling
DACS	**D**omain **A**daptive **C**loud **S**ampling
DG	**D**omain **G**eneralisation
DNN	**D**eep **N**eural **N**etwork
FC	**F**ully **C**onnected
FPS	**F**arthest **P**oint **S**ampling
GPU	**G**raphics **P**rocessing **U**nit
GRL	**G**radient **R**everse **L**ayer
G2L	**G**lobal **to** **L**ocal
ILSVRC-2012	**I**mageNet **L**arge **S**cale **V**isual **R**ecognition **C**hallenge-2012
K	**K**ey
Kin.	**Kin**ect
k-NN	k-**N**earest **N**eighbor
M	**M**odelNet-10
MLP	**M**ulti-**L**ayer **P**erceptron
OS-DAS	**O**pen **S**et-**D**omain **A**daptive **S**ampling
OS-DAPS	**O**pen **S**et-**D**omain **A**daptive **P**oint **S**ampling
OS-DACS	**O**pen **S**et-**D**omain **A**daptive **C**loud **S**ampling
OS-UDA	**O**pen **S**et-**U**nsupervised **D**omain **A**daptation
Q	**Q**uery
ReLU	**Re**ctified **L**inear **U**nit
RPS	**R**andom **P**oint **S**ampling
RS	**R**eal**S**ense
S	**S**hapeNet-10
S*	**S**canNet-10
SA	**S**et **A**bstraction
SEM	**S**tandard **E**rror of the **M**ean

SOTA	**S**tate-**O**f-**T**he-**A**rt
SPL	**S**elf-**P**aced **L**earning
SSL	**S**elf-**S**upervised **L**earning
Syn.	**Syn**thetic
T-S	**T**eacher-**S**tudent
UDA	**U**nsupervised **D**omain **A**daptation
V	**V**alue

Chapter 1

Introduction

In this chapter, we first give a short introduction to the unsupervised domain adaptation techniques for cross-domain point cloud recognition. Then, we present the main challenges of the existing cross-domain point cloud recognition methods and briefly discuss the motivations of our proposed approaches. Finally, we will give the outline of this thesis as well as our contributions.

1.1 Problem Statement

With the huge demand for real-world applications like autonomous driving and robotics, point cloud recognition has received increasing research interest from both academia and industry [43, 70, 16, 59, 83, 98]. Although a variety of methods have been proposed for point cloud recognition, the recognition performance is often unsatisfactory in practical applications due to the huge data variance and limited model generalisation ability. These issues often lead to significant performance degradation of the learnt point cloud recognition model when applied to a new scenario, which is also known as the unsupervised domain adaptation problem for point cloud recognition.

Fig. 1.1 illustrates the unsupervised domain adaptation problem for point cloud recognition. In this problem, we are given a labelled source domain and an unlabeled target domain. Our goal is to learn a point

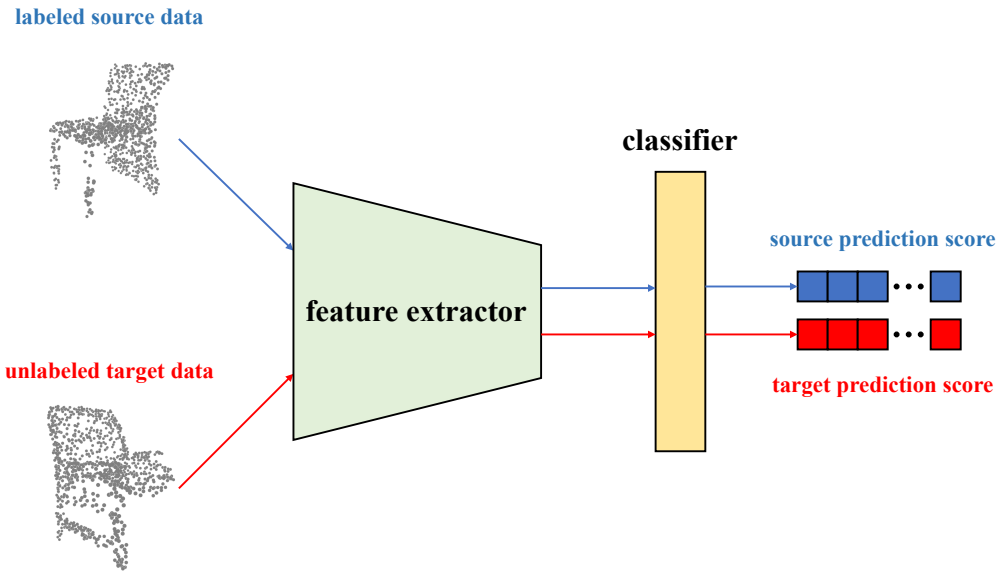

Figure 1.1: Illustration of the pipeline of unsupervised domain adaptation methods. We are given a labelled source domain and an unlabeled target domain. Our goal is to learn a point cloud classification model, including a feature extractor and a classifier, using data from both the source domain and the target domain, as well as the labels from the source domain and can achieve a good classification performance on the target domain without accessing the labels from the target domain.

cloud classification model that performs well on the target domain without accessing the labels of the samples from the target domain. The classification model consists of a feature extractor to extract the global feature of each sample and a classifier to distinguish which class the sample belongs to. The key to solving the unsupervised domain adaptation problem is to reduce the domain discrepancy, as illustrated in Fig. 1.2. We only have labels of the samples from the source domain to train the classifier, but we have data from both domains to train the feature extractor. A general idea is to learn a feature extractor that extracts domain-invariant global features of the samples from both domains, thereby, the classifier trained on the labelled samples from the source domain is able to correctly distinguish the samples not only from the source domain but also the samples from the target domain.

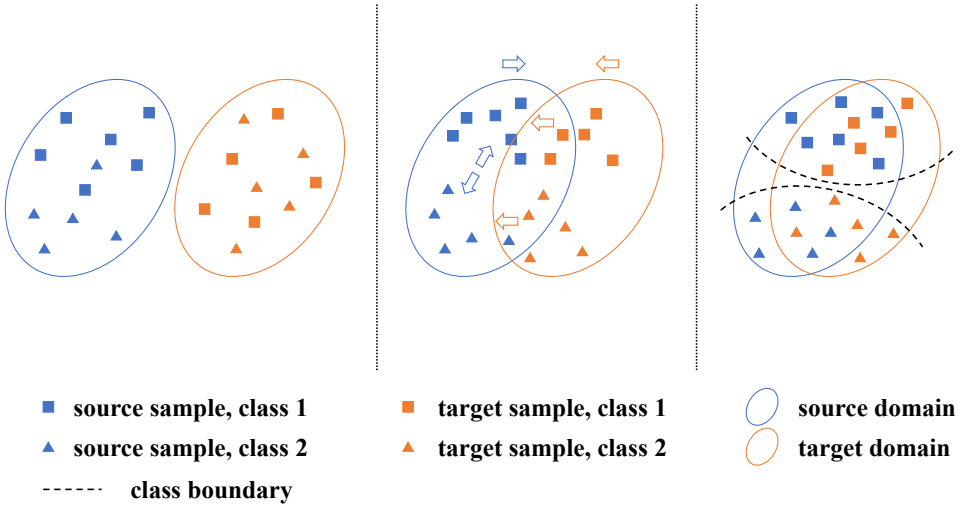

Figure 1.2: Illustration of the pipeline of unsupervised domain adaptation methods.

1.2 Challenges and Motivations

As discussed in Sec. 1.1, unsupervised domain adaptation techniques aim to extract domain-invariant features of samples from different domains. The key problem is the data discrepancy of the point clouds from different domains. First, the 3D capturing devices are diverse. Even for similar objects, the point clouds collected by different devices are often quite different. For example, the point clouds collected with CAD are often dense, while those scanned by depth cameras are usually sparse, leading to considerable domain gaps. Second, environmental changes can also affect the acquired point clouds. Even for the same object, the point clouds scanned by the depth cameras from different angles are quite different. Finally, the objects from the same semantic category might exhibit different geometric appearances, which also leads to variations in point clouds. This data divergence will severely influence the cross-domain recognition performance. Therefore, the main challenge of unsupervised domain adaptation on point clouds is to reduce such data divergence and align samples from different domains. The existing methods for solving the cross-domain point cloud recognition problem can be divided into two categories. The first category contains methods

that are based on adversarial learning methods [61, 79]. However, it is hard to balance both local feature alignment and global feature alignment well through adversarial training. Therefore, most of the recent works focus on constructing suitable self-supervised tasks to help the model extract domain invariant features [1, 50, 101, 22, 69]. However, the point distribution of samples from different datasets is different. If the network is designed to extract the relationship between all the points within each point cloud, it may suffer performance degradation due to the influence of the domain-specific noise or structure. Therefore, we propose a new domain adaptive point sampling (DAPS) strategy to sample domain-invariant point cloud structures useful for semantic domain alignment.

Apart from the challenge of extracting domain-invariant features, another challenge is how to better adapt the model to the target domain. Intuitively, each dataset has its unique distribution. In the unsupervised point clouds domain adaptation problems, even if the feature divergence is reduced, the feature extracted with the learnt model can hardly be optimal for recognition on the target domain, as the feature might lack important target-specific information. To address this challenge, most of the existing works [101, 22, 69] adopt pseudo-labelling methods with a self-paced learning paradigm (SPL) to gradually select confidently predicted target samples as pseudo-labelled samples to fine-tune the learnt model. However, the model may suffer from incorrect pseudo-labels, which may perturb the learnt domain-invariant features. Therefore, we propose a new domain adaptive cloud sampling strategy (DACS) method to learn target-specific information without disturbing the learnt domain-invariant features. Specifically, we design a set of lightweight adapters as add-ons to the learnt model. Then, we fix the learnt model and gradually use the predictions of the confidently predicted samples as the pseudo labels to train the adapters only. In this way, the adapter-based model would be gradually drawn away from the source domain and move close to the target domain, while preventing the vanilla model from being disturbed.

Last but not least, existing cross-domain point cloud recognition approaches only aim at solving the close-set domain adaptation problem, *i.e.*, the source domain and the target domain share the same categories. However, in practical applications, it is unrealistic that the source domain and the target domain share the same classes and it is inevitable that some unknown classes may exist in the target domain, this is also known as the open-set domain adaptation problem [67, 37, 45]. It becomes a great challenge to distinguish those samples from the unknown classes. Currently, open-set unsupervised domain adaptation on point cloud recognition remains an untouched field. Therefore, it is desirable to develop an unsupervised domain adaptation framework to tackle the open-set cross-domain point cloud recognition problem. To this end, we propose a two-stage open-set domain adaptive sampling strategy (OPDAS), which first learns a coarse open-set recognition model by treating all target samples as belonging to the target-specific category and then fine-tunes the recognition model with selected pseudo-labelled target samples using our newly proposed entropy-based pseudo-label selection algorithm to get a fine recognition performance.

In summary, this thesis focuses on developing new cross-domain point cloud recognition strategies to address three main challenges, including (a) How to decrease the data divergence between different domains; (2) How to better adapt the model to the target domain; and (3) How to tackle the open-set cross-domain point cloud recognition issue.

1.3 Thesis Outline and Contributions

The rest parts of this thesis are organised into six chapters. The main contents of these chapters are summarised as follows:

Chapter 2. Literature Review. In this chapter, we give a comprehensive literature review on the background and development of deep learning on point clouds. We also discuss existing works related to the three methods proposed in this thesis.

Chapter 3. Background Introduction of Cross-Domain Point Cloud Recognition. In this chapter, we detail the problem statement and the currently widely used benchmark datasets for cross-domain point cloud recognition.

Chapter 4. Point-Level Domain Adaptive Point Sampling for Cross-Domain Point Cloud Recognition. In this chapter, we propose a new point-level domain adaptive point sampling strategy (DAPS) to sample out domain-invariant structures based on geometry consistency to reduce the data divergence between different domains for better cross-domain point cloud recognition performance. We validate the effectiveness of our DAPS strategy on two benchmark datasets, *i.e.*, PointDA-10 dataset [61] and GraspNetPC-10 dataset [69].

- The contributions in this part are included in:

> **Zicheng Wang**, Wen Li, and Dong Xu, "Domain Adaptive Sampling for Cross-Domain Point Cloud Recognition", *IEEE Transactions on Circuits and Systems for Video Technology*. (Under review)

Chapter 5. Instance-Level Domain Adaptive Cloud Sampling for Cross-Domain Point Cloud Recognition. In this chapter, we propose a new domain adaptive cloud sampling strategy (DACS) to learn target-specific information in addition to the learnt domain-invariant features where we train a set of lightweight adapters as add-ons to the original learnt model without modifying the parameters of the learnt model. We also evaluate the effectiveness of our DACS strategy on the PointDA-10 dataset and GraspNetPC-10 dataset.

- The contributions in this part are included in:

> **Zicheng Wang**, Wen Li, and Dong Xu, "Domain Adaptive Sampling for Cross-Domain Point Cloud Recognition", *IEEE Transactions on Circuits and Systems for Video Technology*. (Under review)

Chapter 6. Open-Set Domain Adaptive Sampling for Open-Set Cross-Domain Point Cloud Recognition. In this chapter, we propose a new

open-set domain adaptive sampling strategy (OS-DAS) to distinguish whether the samples from the target domain belong to the source-known classes or the target-specific class. We come up with a two-stage method to learn a recognition model in a coarse-to-fine manner to tackle the open-set cross-domain point cloud recognition problem and then we verify the effectiveness of our OS-DACS strategy on the PointDA-10 dataset.

Chapter 7. Conclusion and Future Work. In this chapter, we present conclusions and the contributions of this thesis. We also discuss possible research directions in future work.

Chapter 2

Literature Review

In this chapter, we will first present the background of the deep learning methods for visual recognition. Then we will introduce the unsupervised domain adaptation (UDA) problem and the relative works. In addition, we will introduce the self-supervised learning (SSL) problem and the relative works. Finally, we will present relevant research on knowledge transfer.

2.1 Deep Learning on Visual Recognition

2.1.1 Deep Learning on 2D Images Recognition

Deep learning-based methods have shown great success in various fields [43, 13, 14, 25, 42, 100], which is mainly due to the proposal of deep neural networks (DNNs) like the ResNet [32]. The convolutional neural network (CNN) [40, 39, 71, 73, 32, 8, 7, 9, 10, 80, 99, 29] has dominated the 2D feature extractor for many years due to its great potential in increasing the parameter of the model and preventing the model from the over-fitting problem, leading to great recognition performance.

More recently, the Transformer [77] architecture was proposed to capture long-range relations, which inspires tremendous work to use the attention mechanism to solve the vision problems [5, 20, 48, 88, 93, 100, 95] and achieves better recognition performance than CNN in various scenarios.

2.1.2 Deep Learning on 3D Point Clouds Recognition

3D vision has gained increasing research interest as 3D data can contain more spatial information than 2D images, which inspires a huge amount of real-world applications like autonomous driving and robotics [70, 16, 98, 46, 49, 65, 86]. Among the different representations of 3D data, the point cloud is the closest to the actual object, leading to little information loss, and the point cloud can be easily converted to other formats by projection or voxelization, etc [51, 54, 92]. Moreover, the point cloud can be directly scanned from real objects. Therefore, most of the research on 3D data has been focused on point clouds.

Early research on point clouds mainly focused on extracting features of point clouds by first pre-processing the point clouds into voxels and then using 3D convolutional neural networks to extract features from the voxels [52, 58]. However, voxelization results in a large loss of information and is computationally intensive.

The PointNet [57] is a pioneer work that directly uses deep neural networks to operate on unordered point sets. PointNet uses a set of multi-layer perceptions (MLPs) to extract features for each point and then uses a max-pooling operation to aggregate the point features to obtain the global feature of the point set. Moreover, considering that the input points are unordered, PointNet uses a symmetry function, *i.e.*, a transformation matrix to multiply the inputs or the features, to make the model invariant to input permutations.

Based on PointNet, Qi et al. [59] further enhance the model and propose PointNet++. PointNet++ improves the local representation of the point clouds by proposing a set abstraction (SA) layer to encode local features for point clouds. In particular, each SA layer contains a sampling layer, a grouping layer and a PointNet layer. First, a set of key points are sampled using the farthest point sampling (FPS) algorithm where the key points can cover the whole point sets. Second, the neighbour points around each key point are sampled using the ball query algorithm, which can find the points within a given radius around the key

point. Third, the feature of each group is extracted using a mini PointNet. In this way, the SA layer can aggregate the features of the point sets to some down-sampled key points which can contain rich local representations, thus contributing to the learning of the global representation.

Inspired by PointNet++, Wang et al. [83] propose DGCNN, which extends the PointNet++ by using an EdgeConv layer to take the place of the SA layer and omit the down-sampling operation to obtain richer local representations. Specifically, each EdgeConv layer can be divided into a grouping operation and an aggregation operation. The grouping operation aims at finding the k-nearest neighbour (k-NN) of each point node and then calculating the shifted features as the edge features. The aggregation operation applies a channel-wise aggregation operation on the edge features to update the feature of each point node. With a couple of EdgeConv layers, the information can pass through the edges of the point sets, thus enabling each point to contain more semantic information. Similar to PointNet and PointNet++, DGCNN also utilises a max-pooling operation to aggregate the feature of each point to get the global feature. DGCNN has shown its great power in feature extraction and has been used in various works.

Recently, inspired by Transformer, Zhao et al. [98] verify that the attention mechanism can also be applied to the point clouds and they improve the PointNet++ with PointTransformer. By simply introducing an attention module after each SA layer in the PointNet++, PointTransformer achieves excellent performance on various downstream tasks like classification and segmentation, etc, and inspires various following works to explore the application of the Transformer in point clouds [60, 92]. But relatively, the most widely used feature extractor in cross-domain point cloud recognition problems is DGCNN due to its great potential and not easy to overfit.

2.1.3 Weakness

The success of the deep learning methods is mainly due to the huge amount of fully annotated datasets. However, it usually takes great

effort to collect precisely labelled data for training the deep learning models, which limits the practical application of the deep learning models. Moreover, it is difficult for a deep learning model trained with one dataset to perform well on another dataset, mainly due to domain gaps between different datasets, *e.g.*, diverse capturing devices, environmental changes, different object shapes, etc. These problems often lead to significant performance degradation of a learnt model when applied to a new scenario, which is also known as the unsupervised domain adaptation (UDA) problem.

2.2 Unsupervised Domain Adaptation

In the UDA problem, we are given a source domain and a target domain. All of the samples from the source domain are fully annotated while all of the samples from the target domain are unlabelled. There is a domain gap between the source domain and the target domain. Our goal is to learn a model with the labelled source data and the unlabelled target data that performs well on the target domain. As currently, there are far more studies focusing on 2D UDA problems than the 3D UDA problem, and these 2D UDA methods can also give us good insights on the 3D domain, so in this part, we first introduce some classic 2D UDA methods, and then introduce 3D UDA methods currently available. Finally, as there are also some works focusing on domain generalisation on point clouds, which can also provide us with good insights, we also give a brief introduction to the current research about domain generalisation on point clouds.

2.2.1 Unsupervised Domain Adaptation in the 2D Domain

Various methods have been proposed to tackle the UDA problem in the 2D domain. Currently, the 2D UDA methods can be divided into two main categories, *i.e.*, the adversarial-based methods [26, 75, 96, 97, 66, 27, 38, 33] and the prototype-based methods [17, 94].

Adversarial-based UDA methods in the 2D domain

Most adversarial-based UDA methods aim at minimizing the domain gap in the feature space by exploiting the adversarial training framework to directly extract domain-invariant features. Yaroslav et al. [26] proposed DANN, a pioneer work that proposes to use an adversarial training framework to extract domain-invariant features. In particular, DANN trains a domain discriminator in an adversarial manner. On one hand, DANN encourages the discriminator to distinguish whether the features extracted by the feature extractor are from the source domain or the target domain. On the other hand, DANN enforces the feature extractor to extract features of samples from both domains that cannot be distinguished by the discriminator. In this way, the model learnt will recognise the features extracted from the samples belonging to different domains and can achieve good recognition performance on the target domain. MCD [66] points out that the current UDA methods do not consider task-specific decision boundaries during the adaptation process, and most of them only align the features from different domains, ignoring the specific characteristics of each category. To tackle the issue, MCD iteratively trains two classifiers. In the first step, MCD fixes the feature extractor and trains two classifiers and enforces the predictions of the two classifiers to be different. In the second step, MCD fixes the two classifiers and trains the feature extractor and enforces the predictions of the two classifiers to be similar. To sum up, MCD uses two classifiers to replace the discriminator and makes the task-specific decision boundaries robust.

Prototype-based UDA methods in the 2D domain

Recently, most state-of-the-art (SOTA) 2D UDA methods are based on prototype alignment, which share similar ideas with contrastive-based methods and pseudo-labelling-based methods. CAT [17] utilises a teacher-student (T-S) framework and encourages the features extracted by the teacher model from the source data and the target data to update the corresponding prototypes. It should be noticed that the teacher model is

updated by the student model to output stable features. After generating prototypes for the source domain and the target domain, CAT first aligns the features extracted by the student model from the source data with the corresponding source prototypes according to the given ground truth labels and also aligns the features extracted from the target data with the corresponding target prototypes according to the pseudo-labels generated by the teacher model. Finally, CAT aligns the source prototypes with the corresponding target prototypes to minimise the domain gap. These methods utilise the source data as beacons and use pseudo-labelling for domain alignment, which shows great potential and inspires various works like PCS [94] and PACL [17].

Weakness

The above-mentioned methods mainly focus on 2D images to tackle the cross-domain recognition problem, but most of these methods can hardly be adopted to tackle other modalities of the input data directly, *e.g.*, the point cloud. The main reason is that the domain gaps between different domains of data of different modalities have different characteristics, for example, the domain gaps between different domains of 2D images may lie in the global feature space while the domain gaps between different domains of 3D point clouds may probably lie in the local feature space. Therefore, such 2D cross-domain recognition methods may have poor performance on 3D cross-domain recognition tasks, but they can indeed provide us with good insights.

2.2.2 Unsupervised Domain Adaptation in the 3D Domain

While various studies have focused on 3D visual recognition, only a few works focus on the cross-domain point cloud recognition problem [85, 35]. In this chapter, we introduce all of the current 3D UDA methods in detail and also analyse the weakness of these methods. Currently, the 3D UDA methods can be divided into two categories, *i.e.*, the adversarial-based methods [61, 79], and the self-supervised learning (SSL)-based methods [1, 50, 101, 69, 22, 4].

Adversarial-Based UDA Methods in the 3D Domain

Inspired by the 2D UDA methods, Qin et al. [61] proposed PointDAN, which is the first pioneering work to tackle the point cloud UDA problem. PointDAN analyses the difference between point clouds and 2D images in detail, *i.e.*, the point cloud contains much more depth information than 2D images, and the local structures of the point cloud can also contain rich semantic information. Therefore, different from most 2D UDA methods that focus on global feature alignment, it is usually difficult to align the global representations of point clouds from different domains, but it is easy to align the local structures of different point clouds from different domains. To this end, PointDAN proposes to align multi-scale features to minimise the domain discrepancy. In particular, on one hand, PointDAN adopts the idea from MCD and uses the adversarial training framework for global feature alignment, on the other hand, PointDAN also proposes a new concept named the self-adaptive node, which is used for local geometric alignment. The illuminating findings influenced many subsequent studies on 3D UDA problems.

Inspired by PointDAN, Wang et al. [79] proposed DSDAN, which first comes up with a two-view network and uses adversarial training to align the local features of each view with a cross-view consistency loss. Then, DSDAN further proposes to use a pseudo-labelling method to align global features from different domains, indicating the effectiveness of multi-scale feature alignment.

However, although the above-mentioned works [61, 79] have verified the effectiveness of multi-scale feature alignment, it is hard to balance local feature alignment and global feature alignment well through adversarial training. The main reason is that the adversarial training is unstable and it is difficult to guarantee that the alignment is semantically meaningful [69].

Self-Supervised Learning-Based UDA Methods in the 3D Domain

In recent years, various works have been devoted to searching for suitable self-supervised tasks to help models extract domain-invariant features.

DefRec [1] is the first work that focuses on designing self-supervised learning (SSL) sub-tasks to tackle the 3D UDA problem. Similar to DRCN [27], DefRec trains a shared feature extractor for both the source data and the target data. Then, a classification head is trained to generate proposals with labelled source data, and an SSL head is trained to reconstruct the initial input point clouds with the extracted high-level semantic features. Different from DRCN, DefRec reduces the risk of the feature extractor being unable to effectively extract meaningful semantic information on the target domain by allowing reconstruction supervision in both source and target domains. Moreover, DefRec adopts several data augmentations to make sure the feature extractor and different task heads, *i.e.*, the classification head and the SSL head, are robust to generate accurate predictions. A similar idea is also adopted in [50]. However, the reconstruction task can only assist the model to extract low-level geometry features, which may not be that useful for high-level semantic tasks like recognition.

GAST [101] proposes two fancy SSL tasks to assist the domain-invariant semantic feature extraction. The first one is the rotation angle prediction task that uses the features extracted by the shared feature extractor to predict the rotation angle of the augmented input data. The second one is the distortion location prediction task which uses the global feature extracted to predict the distortion location of the input. Different from DefRec which only uses a reconstruction task to reconstruct the input using the high-level learnt feature, the two sub-tasks proposed by GAST can enable the feature extractor to focus on both the local features and the global features. In addition, GAST also adopts a self-paced learning (SPL) paradigm, which makes full use of target samples by selecting confidently predicted target samples as pseudo-labels to fine-tune the model, which guarantees the semantic meaning of the global features learnt by the feature extractor from target samples. However, GAST will

introduce several times training costs and lead to high requirements on the GPU.

GAI [69] proposes to encode the geometry information of the input point clouds into a latent space that can maintain both local information and global information. In particular, GAI adopts the idea of the implicit function by learning an implicit representation and predicting the distance from those discrete points to the surface of a given point cloud. Therefore, the learnt implicit representation can contain rich local geometry information as well as global representations that are suitable for downstream tasks. However, such a self-supervised task cannot handle noisy data like the samples from the ScanNet very well. Therefore, GAI performs poorly in adapting the model learnt using the synthetic data to the noisy data.

GLRV [22] proposes a global scaling-up-down prediction task to predict the scale of the augmented input using the generated global semantics. At the same time, GLRV proposes a local 3D-2D-3D projection-reconstruction task to train the encoder so that it can learn potential global and local representations of the input point clouds. However, incorporating the pseudo-labelling into the training process will make the training unstable and also requires rigorous parameter adjustments.

MLSP [44] proposes a point cardinality estimation sub-task to learn potential information from the basic local structures of a point cloud, a position estimation sub-task to learn an overall geometry of a point cloud, and a normal estimation task to reduce the influence of the noise. In this way, MLSP will learn robust target-specific information. However, as mentioned before, such geometry information is not suitable for high-level classification tasks.

More recently, SD [4] proposes a self-training-based consistency regularization framework to align the prediction of each augmented point cloud with the prediction of the original point cloud, thereby improving the robustness of the network to deal with the domain gap caused by noises. In addition, SD further utilises graph neural networks to improve the reliability of predictions and improve the performance of the

self-training framework. However, such an operation will only mitigate the influence of predictable noises and can hardly reduce the domain gap caused by the different shapes of the objects.

Despite the shortcomings of the above-mentioned methods, extensive studies have demonstrated the feasibility of using SSL sub-tasks to assist the domain-invariant feature extraction in point cloud UDA problems.

Weakness

Most of the above-mentioned 3D cross-domain recognition methods take the whole point sets as input and treat each point with equal contribution. However, the point distributions of samples from different datasets are different. For example, the point distribution of the ScanNet data is relatively sparse and the samples are accompanied by much noise. At the same time, the shapes of samples from different datasets may be different. Therefore, if the network is designed to extract the relationship between all the points, it may suffer performance degradation due to the influence of the domain-specific noise or structure. Therefore, it is vital to search for a domain-invariant structure that is useful for semantic domain alignment.

2.2.3 Domain Generalisation in the 3D Domain

Currently, there are also some works focusing on domain generalisation (DG) on point clouds, which aim at training a model with given labelled source data that can achieve good recognition performance on any given target dataset [35, 85]. It can be seen that DG is a more critical problem than the UDA problem.

MetaSets [35] is the first work that studies the DG problem in the 3D domain. In particular, MetaSets introduces three data augmentations to simulate the scanned data by the given synthetic data. Then, MetaSets adopts a meta-learning framework to enable the model to fit different augmentations. In this way, MetaSets can efficiently improve the robustness and the generalisation ability of the model. However, such a

method can only adapt the model from synthetic domains to scanned domains, and can hardly tackle the domain gap caused by the different shapes of the objects.

PDG [85] is another work that focuses on the generalisation ability of the 3D model. In particular, PDG aligns the local features of each point cloud to a set of calculated part-template features to smooth the local features for better generalization ability. However, an extra dictionary containing the calculated part-template features, an extra cross-attention module and an extra searching operation are required during inference, which will introduce extra parameters and increase computational complexity. Moreover, problems that existed in MetaSets also persist in PDG.

Weakness

Current 3D DG methods adopt all kinds of strong data augmentations to improve the robustness and the generalisation ability of the model. However, most of these methods only focus on adapting the model from synthetic domains to scanned domains, while the cross-domain recognition performance of the model from scanned domains to synthetic domains is usually very poor. However, these methods can still provide us with good insights on cross-domain point cloud recognition.

2.3 Open-Set Unsupervised Domain Adaptation

In real-world scenarios, it is hard to make sure that the semantic categories from the source domain and the target domain are exactly the same. In most cases, the target domain may contain some categories that do not exist in the source domain, making domain alignment more difficult, which is also known as the open-set unsupervised domain adaptation (OS-UDA) problem. Currently, most of the research focuses on the 2D OS-UDA problem and there is no research focusing on the 3D OS-UDA problem yet [24, 36, 45]. Saito et al. [67] first define the OS-UDA problem by considering that the target domain contains not only all of the semantic classes visible in the source domain but also some semantic categories that are specific to the target domain. In this work,

Saito adversarial trains a feature extractor and a classifier to obtain a decision boundary for the source and the target samples. In particular, during the training process, the classifier aims at classifying all target samples as belonging to the unknown category while the feature extractor aims at reducing the probability that the target sample be classified to the unknown category. How to separate the shared classes and the target-specific class and how to align the target samples with the source samples belonging to the shared classes are two critical problems in the OS-UDA problem, and only a handful of works tackle this problem.

2.4 Self-Supervised Learning

Self-supervised learning (SSL) is a powerful tool that assists feature extraction and has attracted great attention. SSL methods aim at training a model using unlabelled training data and ensuring that the learnt model can effectively extract the underlying semantic features of the input data, while the learnt knowledge can be transferred to multiple downstream tasks. As the size of the 2D datasets is usually much larger than the 3D datasets, 2D SSL methods are not identical to 3D SSL methods, but 2D SSL methods can also provide us with great insights.

2.4.1 Self-Supervised Learning in the 2D Domain

The self-supervised methods dominating the 2D domain can be divided into two broad categories: contrastive learning-based methods [11, 31, 28, 12] and reconstruction-based methods [30, 89, 2]. The former approaches aim to search for the relationships between samples and the latter approaches aim at finding the intrinsic relationship within the given input.

A representative work of contrastive learning-based SSL methods is SimCLR [11], proposed by Chen et al., which treats two different augmentations of the same input as positive samples and treats all of the samples within a minibatch as negative samples. Then, SimCLR aims at minimizing the cosine similarity between the positive samples and maximizing the cosine similarity between the negative samples. In this

way, the model is encouraged to extract similar features for similar samples and extract dissimilar features for dissimilar samples. SimCLR is further improved by MoCo [31], which utilises a memory bank to save the negative samples, reducing the heavy reliance on the GPU memory. Contrastive learning is further extended by BYOL [28] and SimSiam [12], which use a non-linear mapping operation to omit the negative samples and simplify the training. Contrastive learning encourages the model to treat each input as an isolated sample and extract deep fundamental information from each sample.

MAE [30] is the representative work of reconstruction-based SSL methods, aiming at searching for relations between local geometries. MAE first divides each input into several equal-sized patches. Then, MAE randomly masks a set of patches by replacing the masked patches with noise. After that, MAE feeds the masked patches into a Transformer architecture to encode the high-level semantics for the masked patches and then decode the semantics to reconstruct the initial unmasked input. MAE utilises the Transformer to capture long-range relationships within each input to find the relationships between patches for high-level semantic information.

Due to the large amount of training data, these 2D SSL methods have shown great success in model pre-training to extract high-level semantic features without acquiring labels.

2.4.2 Self-Supervised Learning in the 3D Domain

The size of the point cloud datasets is usually much smaller than that of the image datasets, mainly because point clouds are not easy to obtain [69]. Therefore, it is hard to train a point cloud feature extractor directly using contrastive learning-based SSL methods or MAE-like SSL methods. In earlier years, most of the works aim at pre-training a point cloud model using reconstruction, but these methods can only acquire low-level geometry information, which may not be suitable for high-level downstream tasks.

Sauder et al. [68] adopt the idea from reconstruction and then propose RS, which aims at using the reconstruction method to find the semantic relationship between different parts within a given point cloud. In particular, RS first divides the initial input point cloud into several voxels according to the coordinates of the points and then randomly rearranges the voxels. Finally, RS trains a model to predict the initial location of each voxel, which is able to effectively learn the semantics of the local structures of the input.

Inspired by contrastive learning-based methods in the 2D domain, Rao et al. [63, 64] propose PointGLR, which performs contrastive learning between the features of the local patches and the global representation. In particular, PointGLR uses a PointNet++ model to extract both the local features for each patch and the global feature for the whole point cloud. Then PointGLR treats the patches and the corresponding whole point clouds as positive pairs. Similarly, PointGLR treats the patches from different point clouds within a minibatch as negative pairs. Finally, PointGLR performs a local-to-global contrastive alignment with InfoNCE loss [31]. To ensure that the model does learn semantic features and prevent the model from stepping into the trivial solution, PointGLR also uses a reconstruction sub-task and a norm estimation sub-task to preserve the semantics of the learnt model. By dividing the point cloud into different patches, the number of training samples can be increased. Moreover, PointGLR can encourage the model to capture the local semantic information efficiently.

There are some other 3D SSL methods like PointBERT [92] and PointMAE [54]. However, these methods rely heavily on the transformer architecture, while the size of the training data is limited in the 3D domain, and the transformer can easily overfit training data, leading to poor generalisation ability.

2.5 Knowledge Transfer

With the development of large pre-trained models like CLIP and MAE, etc. [30, 62, 6, 53, 18], it is efficient to fine-tune the pre-trained model

to get excellent performance for various downstream tasks. Some studies further propose to only train a set of light-weighted modules named Adapters without modifying the learnt parameters, which can also achieve great performance and can boost the fine-tuning efficiency. More importantly, training the adapters does not require many samples as the amount of the parameters of the adapters is very small [19, 34, 72, 56].

However, most of the adapter-based works are based on the Transformer architecture. Li et al. [41] proposed task-specific adapters attached to the convolutional operation to transfer the knowledge learnt from one domain to another, which achieves great success and verifies that the adapter architecture can also be used in other kinds of network backbones. However, how to effectively use the adapter structure for knowledge transfer in the point cloud model is still an untouched field.

2.6 Summary

In this chapter, we introduced the background of deep learning methods and the unsupervised domain adaptation (UDA) problem. Moreover, we introduced some related 2D UDA methods and 3D UDA methods in detail. In addition, we also gave a brief introduction to some self-supervised learning methods in both the 2D domain and the 3D domain that may contribute to model learning. Finally, we also briefly introduced some related works on knowledge transfer. It should be noticed that while there are various works in these research fields, we only present some most representative works in this section.

As discussed above, since the annotation and acquisition of 3D data are more difficult than that of 2D data, the size of 3D datasets is often small, making it difficult to train various complex 3D models. In addition, as a 3D model learnt on one dataset can hardly generalise well on a new dataset, it is necessary to study the 3D UDA problem. Moreover, as mentioned above, the UDA problem is not that realistic compared to the open-set UDA (OS-UDA) problem, and there is no research focusing on the 3D OS-UDA problem so far. Therefore, in this thesis, we first

come up with two 3D UDA methods for domain alignment, and then we further propose a new method to tackle the 3D OS-UDA problem.

Specifically, in Chapter 3, we detail the problem statement and the currently widely used benchmark datasets for cross-domain point cloud recognition.

In Chapter 4, we propose a new domain adaptive point sampling (DAPS) method to sample out the representative points within each point cloud so as to extract domain-invariant features and preserve geometric consistency. In this way, we can shorten the domain gap from the data level.

In Chapter 5, we propose a new domain adaptive cloud sampling (DACS) strategy to gradually learn target-specific information with the selected confidently predicted samples from the target domain using the self-paced learning (SPL) paradigm. As a result, the model can have a better recognition performance on the target domain.

In Chapter 6, we propose a new open-set domain adaptive sampling (OS-DAS) strategy to first learn a coarse recognition model with our proposed open-set domain adaptive point sampling (OS-DAPS) strategy and then refine the model with our proposed open-set domain adaptive cloud sampling (OS-DACS) method.

Chapter 3

Background Introduction of Cross-Domain Point Cloud Recognition

In this chapter, we detail the background of the cross-domain point cloud recognition task, including the problem statement and the current widely-used benchmark datasets for cross-domain point cloud recognition.

3.1 Problem Statement

In this section, we briefly introduce the problem statement for the cross-domain point cloud recognition task, including the close-set unsupervised domain adaptation on point clouds and the open-set unsupervised domain adaptation on point clouds.

3.1.1 Close-Set Unsupervised Domain Adaptation on Point Clouds

In the task of unsupervised domain adaptation (UDA) on point clouds, we are given a labelled source domain and an unlabelled target domain. Our goal is to learn a point cloud classification model that performs well on the target domain without accessing the target labels.

Formally, we denote the source domain as $\mathcal{S} = \{\mathcal{P}_n^s, y_n^s\}_{n=1}^{N_s}$, which consists of N_s point clouds $\mathcal{P}^s$ with their corresponding category labels

y_n^s, $n = \{1, 2, \cdots, N_s\}$. Similarly, the target domain $\mathcal{T} = \{\mathcal{P}_n^t, y_n^t\}_{n=1}^{N_t}$ consists of N_t samples $\mathcal{P}_n^t$ and the labels y_n^t are only available during testing, $n \in \{1, 2, \cdots, N_t\}$. Each point cloud $\mathcal{P}_n^s$ or $\mathcal{P}_n^t$ consists of M three-dimensional points, (x, y, z), indicating the spatial coordinates of the points. Let us assume there are C categories in total, *i.e.* $y_n^s \in \{1, 2, \cdots, C\}$ and $y_n^t \in \{1, 2, \cdots, C\}$ for any samples from the source domain or the target domain, respectively. For better presentation, below we may ignore the subscript s and t when it is unnecessary to distinguish whether the point clouds come from the source domain or the target domain. We also denote the point cloud recognition model as Φ. In the classification tasks, the model usually consists of a feature extractor Φ_{f} that operates on the raw input of the point sets and outputs the global features of the point cloud (*i.e.* g), as well as a classifier Φ_{cls} that uses the feature g as an input and outputs the classification probability of all C classes, which can be formulated as follows:

$$\Phi(\mathcal{P}) = \Phi_{\mathrm{cls}}(\Phi_{\mathrm{f}}(\mathcal{P})) \tag{3.1}$$

The current deep learning-based recognition models often learn Φ_{cls} and Φ_{f} jointly by using the end-to-end training strategy. However, as we only have the labels for the source data, the feature extractor Φ_{f} and the classifier Φ_{cls} would be inevitably biased towards the source domain, leading to poor performance on the target domain during testing. Therefore, our task is to leverage the unlabelled target samples, such that we can use Φ_{f} to extract domain-invariant features and employ Φ_{cls} to acquire better recognition performance on the target domain.

3.1.2 Open-Set Unsupervised Domain Adaptation on Point Clouds

In the task of open-set unsupervised domain adaptation (OS-UDA) on point clouds, we are given a labelled source domain and an unlabelled target domain. The categories existing in the target domain are not only all the categories in the source domain but also the categories that are

specific to the target domain. Our goal is to learn a point cloud classification model that can not only recognise the target samples from the shared classes of the two domains, and also recognise the samples belonging to the target-specific classes.

The problem statement of 3D open-set unsupervised domain adaptation is almost the same as the problem statement of 3D close-set unsupervised domain adaptation, the only difference is that in the 3D OS-UDA task, there are C categories in the source domain and $C+1$ categories in the target domain, *i.e.* $y_n^s \in \{1, 2, \cdots, C\}$ and $y_n^t \in \{1, 2, \cdots, C+1\}$. It should be mentioned that all of the samples belonging to the classes that are unique in the target domain can be divided into the $C+1$th class.

The current deep learning-based recognition models often learn Φ_{cls} and Φ_{f} jointly by using the end-to-end training strategy. However, as we only have the labels for the source data, the feature extractor Φ_{f} and the classifier Φ_{cls} would be inevitably biased towards the source domain and the categories belonging to the source domain, leading to poor performance on recognising the samples from the target domain or distinguish the samples from target-specific class during testing. Therefore, our task is to leverage the unlabelled target samples, such that we can use Φ_{f} to extract domain-invariant features and employ Φ_{cls} to acquire better recognition performance on the target domain.

3.2 Datasets Introduction

In this section, we briefly introduce the two benchmark datasets used for the cross-domain point cloud recognition task, *i.e.*, PointDA-10 [61] and GraspNetPC-10 [69].

The PointDA-10 dataset consists of three subsets, ModelNet-10 (M), ShapeNet-10 (S) and ScanNet-10 (S*), which are sampled from three widely-used datasets, ModelNet [78], ShapeNet [87] and ScanNet [15], respectively. The three subsets share the same 10 categories. In particular,

ModelNet-10 consists of 4,183 training samples and 856 testing samples, where these samples are generated with CAD by uniformly sampling from synthetic 3D models. ShapeNet-10 consists of 17,387 training samples and 2,492 testing samples, where the samples are also generated with CAD, but the shape of the samples in ShapeNet-10 also exhibits more variations than those in ModelNet-10. ScanNet-10 consists of 6,110 training samples and 1,769 testing samples, and all of the samples are scanned from real-world indoor scenarios with RGB-D cameras. The samples in ScanNet-10 are usually sparse with some missing parts caused by noise and occlusion.

The GraspNetPC-10 dataset is created from GraspNet [23], which focuses on two challenge scenarios in cross-domain point cloud recognition, *i.e.*, the sim-to-real scenario and the real-to-real scenario. The synthetic samples in GraspNetPC-10 are re-projected from rendered synthetic scenes while the real depth scanned samples in GraspNetPC-10 are captured by two different depth cameras, *i.e.*, Kinect2 and Intel Realsense, consisting of two domains of real-world point clouds. The subsets also share the same 10 categories. In particular, the synthetic domain (Syn.) contains 12,000 training samples. The Kinect real-world domain (Kin.) contains 10,973 training samples and 2,560 testing samples while the Realsense real-world domain (RS) contains 10,698 training samples and 2,560 testing samples.

When performing the unsupervised domain adaptation task, we use the training set of one subset as the source domain, and the training set of another subset as the target domain, which leads to six domain adaptation scenarios in the PointDA-10 dataset, and four domain adaptation scenarios in GraspNetPC-10 dataset. Note that the labelled training set from the target domain is not used, while the testing set of the target subset is used for performance evaluation.

When performing the open-set unsupervised domain adaptation task, we use the training set of one subset as the source domain, and the training set of another subset as the target domain, which leads to six domain adaptation scenarios in the PointDA-10 dataset. Note that all of the three sub-datasets share the same 10 categories, in this work, we only keep the

samples belonging to the former 5 categories from the source domain and omit the remaining samples. In addition, we keep all of the samples from the target domain and we treat all of the samples belonging to the latter 5 categories from the target domain as the samples belonging to the target-specific class, *i.e.*, the number of the shared classes is 5 in this work and the labels of the samples belonging to the latter five categories from the target domain are set as 5 (class labels start counting from 0). Also note that the training labels from the target domain are not used, while the testing set of the target subset is used for performance evaluation.

Chapter 4

Point-Level Domain Adaptive Point Sampling for Cross-Domain Point Cloud Recognition

In this chapter, we propose a new domain adaptive point sampling (DAPS) strategy to enhance the domain-invariant representations of point clouds by progressively focusing on representative points within each point cloud based on geometric consistency. We validate our DAPS strategy on the benchmark datasets, *i.e.*, PointDA-10 and GraspNetPC-10, and demonstrate the effectiveness of our method.

4.1 Motivations and Contributions

Due to the great success in 3D visual recognition [57, 59, 83, 98, 74], researchers are now focusing on 3D UDA problems. Previous works [61, 79] focus on using the adversarial learning strategy for local and global feature alignment to increase the generalisation ability of the model. However, it is hard to balance local feature alignment and global feature alignment well with adversarial training [22].

Most recent works focus on searching for proper self-supervised tasks, like reconstruction [1, 50, 22], rotation prediction [101], and surface estimation [69], to help the model extract semantic meaningful features

from both domains. Most of these works take the whole point sets as the input and treat each point with equal contribution. However, the point distribution of samples from different datasets is different. For example, the point distribution of the ScanNet data is relatively sparse and the samples are accompanied by much noise. Moreover, the shapes of the samples from different domains are quite different. If the network is designed to extract the relationship between all the points, it could cause performance degradation, due to the influence of the domain-specific noise or structure.

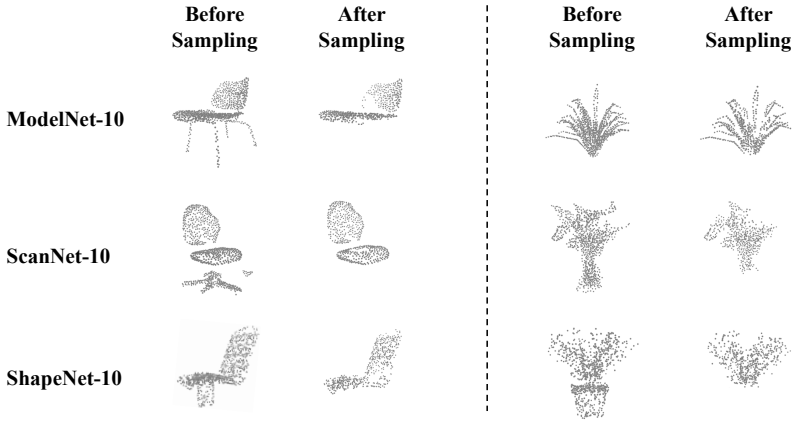

Figure 4.1: The visualization of the sampled points after using our domain adaptive point sampling method on different samples.

Therefore, we aim at finding a domain-invariant structure that is useful for semantic domain alignment. To this end, we propose a domain adaptive point sampling (DAPS) strategy to reduce data divergence by paying more attention to the domain-invariant structures within each point cloud. Intuitively, the model is prone to distinguish those commonly-seen geometric structures from those more unique structures, and these commonly-seen geometric structures are mostly shared across different domains. For this purpose, on one hand, we need to sample out the domain-invariant structure, *i.e.*, the representative points, on the other hand, the representative points should be able to represent the intrinsic feature of the whole point cloud. Therefore, for each point cloud, we identify these representative points based on the similarity between a set of local features from small regions and the global feature from the

whole point cloud. With the DAPS strategy, we are able to focus our attention on the intrinsic part of each sample that contributes most to the semantic meaning that it contains. As shown in Figure 4.1, the styles of chair legs are often diverse and the chair legs are also easy to lose in the scanned point clouds. With the DAPS strategy, we are able to focus our attention on the intrinsic part of each chair that contributes most to recognizing which category it belongs to.

The main contributions in this chapter can be summarised as follows: (1) We propose a new representative point sampling method to sample out the representative points within each point cloud so as to produce domain-invariant features and preserve geometric consistency. (2) We use our proposed domain adaptive point sampling method to train the recognition network for unsupervised domain adaptation on point clouds. Comprehensive experiments on benchmark datasets have demonstrated the effectiveness of our newly proposed domain adaptive point sampling approach.

4.2 Methodology

In this section, we will introduce our newly proposed domain adaptive point sampling strategy (DAPS) in detail. Here we adopt DGCNN [83] as our backbone, and the pipeline of our DAPS is shown in Figure 4.2. In particular, we apply the domain adaptive point sampling strategy within each point cloud based on the geometric consistency to sample out the representative points for each point cloud in a coarse to fine fashion, as shown in Figure 4.2 (a). In particular, we come up with two variations of the backbone, *i.e.*, expand-DGCNN and DAPS-DGCNN, as shown in Figure 4.2 (b), where the expand-DGCNN is the vanilla DGCNN combined with a classifier, a global-to-local (G2L) consistency module and a reconstruction module, and the DAPS-DGCNN is the combination of the expand-DGCNN module and our point sampling module.

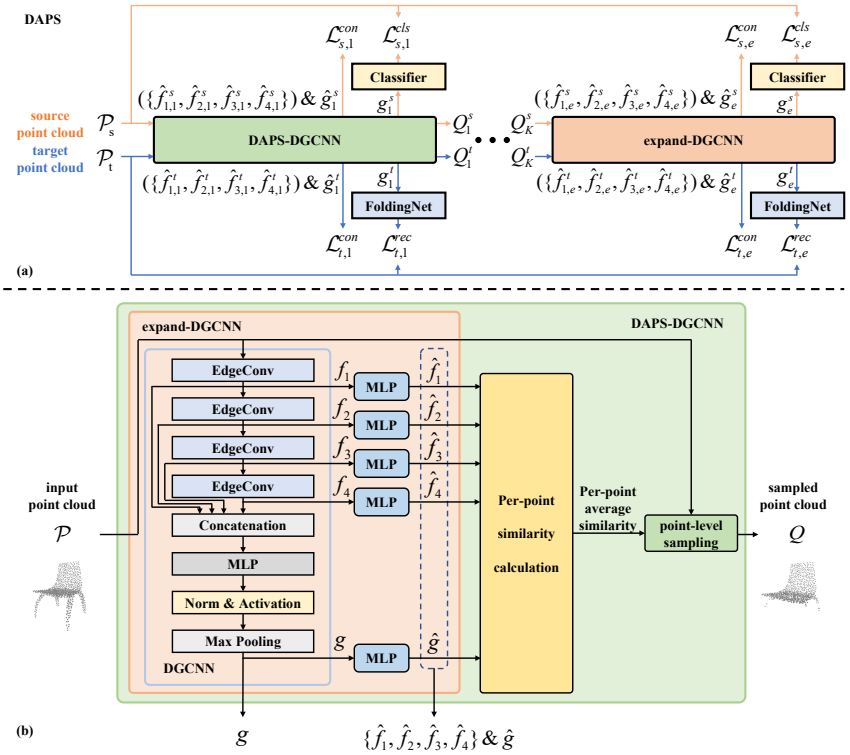

Figure 4.2: Overview of our domain adaptive point sampling (DAPS) module. (a) Our DAPS method selects the representative points that can preserve geometric consistency according to the per-point similarity between the local feature and the global feature, which is combined with three losses, *i.e.*, a global-to-local (G2L) consistency loss, a classification loss determined on the source samples only and a reconstruction loss determined on the target samples only. It consists of K DAPS-DGCNN modules and an expand-DGCNN module. (b) The detailed network structure of our proposed DAPS-DGCNN, expand-DGCNN and vanilla DGCNN.

4.2.1 Representation Learning through Domain Adaptive Point Sampling

Our approach builds upon the existing work, DGCNN. It consists of L EdgeConv layers ($L = 4$ in this work) to aggregate the features from a certain local region to each point, followed by a pooling operation to produce the global feature for classification. In particular, given a point cloud $\mathcal{P}$, we represent the local feature of each point at each EdgeConv layer as $f_{i,l,n}$, where i is the index of a point in each point cloud, l is the index of the layer of the neural network where $l = 1, \cdots, L$, and n is the

index of the point cloud. Moreover, the global feature of the n-th sample after the pooling operation is denoted by g_n.

Domain Adaptive Point Sampling

When learning the feature representation for 3D point clouds, we design the DAPS strategy to identify the representative points within each point cloud and expect the network to pay more attention to these representative points than the noisy points. In Figure 4.2 (b) we illustrate our DAPS module. We assume that the model is prone to distinguish those commonly-seen geometric structures, which are mostly shared across domains, from those unique structures. As the model has the potential to capture both the local feature and the global feature, the model could easily extract semantic features from these commonly-seen geometric structures, which we call the representative points. Therefore, we further assume that if the local feature of a point is able to infer the overall semantic meaning of the object, it is likely that the corresponding point is the representative one. Therefore, we calculate the similarities between the global feature g_n and the local features $f_{i,l,n}$ of each point at different layers, and then select the points with the average similarity over all layers higher than a defined threshold as the representative points.

As the feature dimensions of the global feature and the local features at different layers are different, we introduce a simple multi-layer perceptron (MLP) network as the mapping module for the features and transform different features into a common feature space. Let us denote Φ^l_{map} as the MLP network at the l-th layer, and Φ^g_{map} as the MLP network for the global feature. The similarity between the feature of each point and the global feature in the common space can be calculated as:

$$\text{sim}(f_{i,l,n}, g_n) = \frac{\Phi^l_{\text{map}}(f_{i,l,n}) \cdot \Phi^g_{\text{map}}(g_n)}{||\Phi^l_{\text{map}}(f_{i,l,n})|| \times ||\Phi^g_{\text{map}}(g_n)||} \tag{4.1}$$

Then we summarise the similarities from different layers to produce the average similarity value for domain adaptive point sampling, *i.e.*

$$\mathrm{S}_{i,n} = \frac{1}{L}\sum_{l=1}^{L} \mathrm{sim}(f_{i,l,n},\, g_n) \tag{4.2}$$

Only if $\mathrm{S}_{i,n}$ is larger than a pre-defined threshold ϵ, we will treat the i-th point from the n-th point cloud as a representative point, otherwise, we will treat the point as noise.

Learning Feature Mappings

To learn meaningful feature mapping Φ^{l}_{map} for $l = 1, \cdots, L$ and Φ^{g}_{map}, we introduce a global-to-local (G2L) consistency loss for training these MLP networks, which forces the network to extract local features from each point to be similar with the global feature from the same point cloud and dissimilar with the global features from other point clouds. Here we formulate the G2L consistency loss [63] as below:

$$\ell^{\mathrm{con}}(f_{i,l,n},\, g_n) = -\log\frac{\Phi^{l}_{\mathrm{map}}(f_{i,l,n}) \cdot \Phi^{g}_{\mathrm{map}}(g_n)}{\sum_m \Phi^{l}_{\mathrm{map}}(f_{i,l,n}) \cdot \Phi^{g}_{\mathrm{map}}(g_m)} \tag{4.3}$$

and

$$\mathcal{L}^{\mathrm{con}} = \frac{\frac{1}{N_s}\sum_{i,l,n} \ell^{\mathrm{con}}_{\mathrm{s}}(f_{i,l,n},\, g_n) + \frac{1}{N_t}\sum_{i,l,n} \ell^{\mathrm{con}}_{\mathrm{t}}(f_{i,l,n},\, g_n)}{ML} \tag{4.4}$$

where $\ell^{\mathrm{con}}_{\mathrm{s}}(f_{i,l,n},\, g_n)$ and $\ell^{\mathrm{con}}_{\mathrm{t}}(f_{i,l,n},\, g_n)$ are the consistency losses defined for the point clouds from the source domain and the target domain, respectively, M is the total number of points within each point cloud, while N_s and N_t represent for the number of samples in the source domain or the target domain, respectively. Note that we perform the G2L consistency alignment on both the source domain and the target domain. With the G2L consistency loss, we can train the mapping networks to calculate the similarity between the local features and the global features, thus enhancing the reliability of our DAPS. Moreover, we can encourage the network to learn an object from only a part of it, thus increasing the generalisation ability.

Semantic Preserving

To ensure that our method can preserve meaningful semantics when extracting the features, we further impose a classification loss for labelled data from the source domain and a reconstruction loss for unlabelled data from the target domain when training our model.

In particular, as the semantic labels are available for the samples from the source domain, we employ the cross-entropy loss as the classification loss.

$$\mathcal{L}^{\text{cls}} = -\frac{1}{N_s}\sum_{n=1}^{N_s} \ell^{ce}(\Phi(\mathcal{P}_n^s), y_n^s) \tag{4.5}$$

where ℓ^{ce} denotes the cross entropy loss, $\mathcal{P}_n^s$ and y_n^s stand for the n-th sample in the source domain and the corresponding label.

For the target domain, since the semantic labels are not available, we, therefore, use the reconstruction loss to train our model. In particular, we follow the design in FoldingNet [91] to build a reconstruction network Φ_{rec}, and use the global feature g_n to recover the whole point sets. Given a target point cloud $\mathcal{P}_n^t$, let us denote the reconstructed point cloud as $\hat{\mathcal{P}}_n^t$, then the reconstruction loss is defined as:

$$\mathcal{L}^{\text{rec}} = \frac{1}{N_t}\sum_{n=1}^{N_t} \ell^{rec}(\mathcal{P}_n^t, \hat{\mathcal{P}}_n^t) \tag{4.6}$$

where $\ell^{rec}(\mathcal{P}_n^t, \hat{\mathcal{P}}_n^t) = \sum_{y\in\hat{\mathcal{P}}_n^t} \min_{x\in\mathcal{P}_n^t} ||x-y||_2 + \sum_{x\in\mathcal{P}_n^t} \min_{y\in\hat{\mathcal{P}}_n^t} ||x-y||_2$ is the Chamfer distance [21] between $\mathcal{P}_n^t$ and $\hat{\mathcal{P}}_n^t$.

Progressive Representation Learning

When learning the model, we jointly consider the G2L consistency loss, the cross entropy loss and the reconstruction loss to learn the model. The total loss can be written as follows,

$$\mathcal{L}^{\text{DAPS}} = \lambda_1\mathcal{L}^{\text{cls}} + \lambda_2\mathcal{L}^{\text{con}} + \lambda_3\mathcal{L}^{\text{rec}} \tag{4.7}$$

λ_1, λ_2 and λ_3 are the trade-off parameters.

However, if we filter out too many points in each point cloud, the global feature of the point cloud would vary a lot. Some of the representative points might be treated as noisy points, and the remained points may be too few to provide useful semantic information. Therefore, we design a progressive learning strategy to simultaneously perform point sampling and representation learning by only filtering out a small part of noisy points within each DAPS-DGCNN training stage, and we perform K DAPS-DGCNN training stages in total. As introduced above, for the first stage of DAPS-DGCNN training, we first feed the complete point cloud into the model and perform the first round of point sampling based on the similarity calculated by Eq. (4.2). Then we perform the next DAPS-DGCNN training stage by using the sampled representative points as the input of the model. After K ($K = 2$ in this work) DAPS-DGCNN training stages, we use the sampled representative points as the input to finally train an expand-DGCNN. Note that for each feed-forward process, we calculate the loss $\mathcal{L}^{\text{DAPS}}$ from the DAPS-DGCNN and sum all losses from all K DAPS-DGCNN of feed-forward processes together with the loss $\mathcal{L}^{\text{expand}}$, which is calculated in the same way as Eq. (6.1), for the last expand-DGCNN training stage for back propagation. The final loss $\mathcal{L}^{\text{prl}}$ of our progressive representation learning method can be calculated as:

$$\mathcal{L}^{\text{prl}} = \sum_{k=1}^{K} \mathcal{L}_k^{\text{DAPS}} + \mathcal{L}^{\text{expand}} \tag{4.8}$$

where k denotes the k-th DAPS-DGCNN feed-forward process for each point cloud. Note here no weight coefficients of $\mathcal{L}^{\text{DAPS}}$ or $\mathcal{L}^{\text{expand}}$ are required as we treat our DAPS-DGCNN and expand-DGCNN with equal contribution. With the progressive training strategy, we can make the gradient backwards propagate multiple times through those representative points and make the network focus more on learning the features of these representative points and eventually learn more useful representations from these representative points.

4.2.2 Experiments

In this section, we first list the implementation details of our DAPS strategy. We further evaluate our DAPS strategy on the two benchmark datasets. In addition, we analyse the effectiveness of different modules in our DAPS strategy. Finally, we present some visualization results to verify the effectiveness of our DAPS strategy.

Implementation Details

Here, we follow the previous works [1, 101, 69] and adopt DGCNN as the backbone. We use two-layer MLPs with a hidden layer dimension of 512 to map the output features of each EdgeConv layer as well as the global features to the same dimension of 512. Our methods are trained on the server with four NVIDIA RTX A6000 GPUs, and our implementation is based on the PyTorch framework. During training, we use the Adam optimizer together with an epoch-wise cosine annealing learning rate scheduler in which we set the learning rate to 0.001 and the weight decay as 0.00005. We set the hyper-parameters λ_1, λ_2 and λ_3 to 0.5, 0.5 and 0.5, respectively. Moreover, we set the number of DAPS-DGCNN training stages (*i.e.*, K) as 2 and the sampling ratio as 0.9, which means we perform domain adaptive sampling for twice and use the sampled points to train the expand-DGCNN. We train our DAPS module for 100 epochs with a batch size of 18.

Experimental Results

We compare our proposed approach with the recent state-of-the-art (SOTA) point cloud-based unsupervised domain adaptation methods, including PointDAN [61], RS [68], DefRec with PCM [1], GAST [101] and the latest GAI [69] and GLRV [22]. Moreover, we include the popular 2D UDA method DANN [26] for a fair comparison. In addition, we also include the results of supervised learning methods that directly train the model with only labelled source data for comparison (denoted as "w/o DA"). The Oracle method [83] that trains the model by using labelled target data is also listed for comparison (denoted as "Oracle"). Moreover, as most of the recent 3D UDA methods are based on a self-paced learning

Table 4.1: The classification accuracies (mean ± SEM) of different methods over 3 rounds of experiments on the GraspNetPC-10 dataset. The numbers in the brackets denote the year of the compared methods.

Methods	Syn. → Kin.	Syn. → RS	Kin. → RS	RS → Kin.	Avg.
Oracle (2019) [83]	97.2 ± 0.8	95.6 ± 0.4	95.6 ± 0.3	97.2 ± 0.4	96.4
w/o DA	61.3 ± 1.0	54.4 ± 0.9	53.4 ± 1.3	68.5 ± 0.5	59.4
Baseline	80.3 ± 2.3	66.3 ± 2.0	65.5 ± 0.5	73.0 ± 1.8	71.3
DANN (2016) [26]	78.6 ± 0.3	70.3± 0.5	46.1 ± 2.2	67.9 ± 0.3	65.7
PointDAN (2019) [61]	77.0 ± 0.2	72.5 ± 0.3	65.9 ± 1.2	82.3 ± 0.5	74.4
RS (2019) [68]	67.3 ± 0.4	58.6 ± 0.8	55.7 ± 1.5	69.6 ± 0.4	62.8
DefRec+PCM (2021) [1]	80.7 ± 0.1	70.5 ± 0.4	65.1 ± 0.3	77.7 ± 1.2	73.5
GAST (2021) [101]	81.3 ± 1.8	72.3 ± 0.8	61.3 ± 0.9	80.1 ± 0.5	73.8
GAI (2022) [69]	94.6 ± 0.4	80.5 ± 0.2	76.8 ± 0.4	85.9 ± 0.3	84.4
DAPS	**97.0** ± 0.2	79.6 ± 0.8	**79.1** ± 0.7	**95.7** ± 0.7	**87.8**

paradigm, we construct another baseline by first training the model with labelled source data which is then fine-tuned with the pseudo-labelled target data based on the self-paced learning paradigm (denoted as "Baseline"). It should be noticed that for a fair comparison, our DAPS is also implemented based on a self-paced learning paradigm.

We first validate our method on the newly proposed GraspNetPC-10 dataset. For each method, we run the experiments on each domain adaptation scenario for 3 rounds. The recognition accuracy and standard error of the mean (SEM) of each method on each scenario are reported in Table 4.1. Here we use 'Syn.', 'Kin.' and 'RS' to denote the synthetic domain, the Kinect real-world domain, and the Realsense real-world domain, respectively. For example, Syn. → Kin. means we use the synthetic domain as the source domain and the Kinect real-world domain as the target domain. We observe that our proposed method achieves the state-of-the-art (SOTA) results on GraspNetPC-10, exceeding the current SOTA method GAI by a notable margin of 3.4%. Specifically, when using the Realsense real-world domain (RS) as the source domain and using the Kinect real-world domain (Kin.) as the target domain, our proposed DAPS strategy surpasses the SOTA method GAI by 9.8%, demonstrating the effectiveness of our proposed approach.

We further validate our method on the widely used PointDA-10

Table 4.2: The classification accuracies (mean ± SEM) of different methods over 3 rounds of experiments on the PointDA-10 dataset. The numbers in the brackets denote the year of the compared methods.

Methods	M → S	M → S*	S → M	S → S*	S* → M	S* → S	Avg.
Oracle (2019) [83]	93.9 ± 0.2	78.4 ± 0.6	96.2 ± 0.1	78.4 ± 0.6	96.2 ± 0.1	93.9 ± 0.2	89.5
w/o DA	83.3 ± 0.7	43.8 ± 2.3	75.5 ± 1.8	42.5 ± 1.4	63.8 ± 3.9	64.2 ± 0.8	62.2
Baseline	85.4 ± 0.3	56.9 ± 0.2	76.5 ± 0.2	53.5 ± 0.7	76.9 ± 0.4	73.6 ± 1.3	70.5
DANN (2016) [26]	74.8 ± 2.8	42.1 ± 0.6	57.5 ± 0.4	50.9 ± 1.0	43.7 ± 2.9	71.6 ± 1.0	56.8
PointDAN (2019) [61]	83.9 ± 0.3	44.8 ± 1.4	63.3 ± 1.1	45.7 ± 0.7	43.6 ± 2.0	56.4 ± 1.5	56.3
RS (2019) [68]	79.9 ± 0.8	46.7 ± 4.8	75.2 ± 2.0	51.4 ± 3.9	71.8 ± 2.3	71.2 ± 2.8	66.0
DefRec+PCM (2021) [1]	81.7 ± 0.6	51.8 ± 0.3	78.6 ± 0.7	54.5 ± 0.3	73.7 ± 1.6	71.1 ± 1.4	68.6
GAST (2021) [101]	84.8 ± 0.1	59.8 ± 0.2	80.8 ± 0.6	56.7 ± 0.2	81.1 ± 0.8	74.9 ± 0.5	73.0
GLRV (2022) [22]	85.4 ± 0.4	60.4 ± 0.4	78.8 ± 0.6	**57.7** ± 0.4	77.8 ± 1.1	76.2 ± 0.6	72.7
GAI (2022) [69]	86.2 ± 0.2	58.6 ± 0.1	81.4 ± 0.4	56.9 ± 0.2	81.5 ± 0.5	74.4 ± 0.6	73.2
MLSP (2022) [44]	85.7 ± 0.6	59.4 ± 1.3	**82.3** ± 0.9	57.3 ± 0.7	**82.2** ± 0.5	76.4 ± 0.5	73.8
DAPS	**86.9** ± 0.5	**59.7** ± 0.5	78.7 ± 1.2	55.5 ± 1.1	82.0 ± 2.0	**80.5** ± 0.7	**73.9**

dataset. We follow a similar setting to the experiments of the GraspNetPC-10 dataset by running the experiments on each domain adaptation scenario for 3 rounds. The recognition accuracy and standard error of the mean (SEM) of each method on each scenario are reported in Table 4.2. Here we use 'M', 'S' and 'S*' to denote the ModelNet-10, ShapeNet-10, and ScanNet-10 subsets, respectively. For example, M → S means we use ModelNet-10 as the source domain and ShapeNet-10 as the target domain. We observe that our proposed method achieves the SOTA results on PointDA-10. With the help of our DAPS method, we achieve an average accuracy of 73.9%. The results clearly demonstrate the effectiveness of our proposed approach.

Specifically, when using ScanNet-10 (*i.e.*, S*) as the source domain, the results of our DAPS strategy are boosted significantly. A possible explanation is as follows. When compared to ModelNet-10 and ShapeNet-10, the point clouds in ScanNet-10 are quite sparse and often with missed parts, which leads to considerable variation within each point cloud from the same class. With our proposed DAPS module, the model is able to focus on the representative points, which reduces the divergence between ScanNet-10 and the other two subsets, and leads to better performance.

Table 4.3: The classification accuracies when using different losses with or without the sampling strategy in DAPS-DGCNN training on the PointDA-10 dataset.

con	rec	sampling	Avg.
	✓		68.5
✓			68.9
✓	✓		69.0
	✓	✓	69.7
✓		✓	70.5
✓	✓	✓	70.8

Ablation Study

In this section, we analyse the effectiveness of the detailed modules designed for our domain adaptive point sampling (DAPS) on the PointDA-10 dataset. In DAPS, three losses are used to learn the feature mappings and preserve the feature semantics. As the cross entropy loss is compulsory for training the model, we now investigate the individual contributions of the G2L consistency loss and the reconstruction loss, as well as the contribution of our sampling method. Here we discard the SPL paradigm and only focus on the performance of our DAPS module.

We conduct the experiments for 3 rounds on 6 domain adaptation scenarios. In Table 4.3, we report the mean accuracies over 6 cases, where the "con" represents the G2L consistency loss, "rec" represents the reconstruction loss and "sampling" represents our representative points sampling strategy. The SEM results are not included for simplicity. We observe that the alternative methods by using either G2L consistency loss or the reconstruction loss can contribute to domain alignment as these two methods both enable the model to learn domain-invariant information in a self-supervised learning fashion. Our domain adaptive point sampling process can further help the model to learn domain-invariant features for achieving better recognition results, with an improvement of more than 1%. If we only apply representative point sampling and the reconstruction loss, the sampling process would be random, as there is no supervision for training the mapping functions in MLPs. Therefore, the model would only learn the global semantic meaning from the part

Table 4.4: The classification accuracies (mean ± SEM) of different sampling methods over 3 rounds of experiments on the PointDA-10 dataset.

Methods	M → S	M → S*	S → M	S → S*	S* → M	S* → S	Avg.
Oracle	93.9 ± 0.2	78.4 ± 0.6	96.2 ± 0.1	78.4 ± 0.6	96.2 ± 0.1	93.9 ± 0.2	89.5
w/o DA	83.3 ± 0.7	43.8 ± 2.3	75.5 ± 1.8	42.5 ± 1.4	63.8 ± 3.9	64.2 ± 0.8	62.2
FPS	79.6 ± 1.0	45.5 ± 0.9	76.7 ± 0.7	43.1 ± 0.4	65.7 ± 1.2	62.3 ± 0.7	62.1
RPS	80.5 ± 0.8	48.3 ± 2.2	74.3 ± 0.5	49.9 ± 1.3	62.7 ± 1.4	66.5 ± 1.7	63.7
DAPS	**84.6** ± 0.9	**59.2** ± 0.4	**77.1** ± 0.6	**56.0** ± 0.8	**73.1** ± 0.8	**76.2** ± 0.9	**70.8**

missing point cloud, namely, only the robustness would be improved. However, if we apply the G2L consistency loss, we can filter out the noisy points for domain-invariant feature learning, thus the recognition accuracy is improved by 1.6%, which verifies the effectiveness of our sampling module. Moreover, if we apply all of the three losses, the semantic meaning would be best preserved, thus leading to an average accuracy of 70.8%. The experiments demonstrate that it is effective to learn the feature mappings and preserve the semantic meaning during the domain adaptive point sampling process.

We then verify the effectiveness of our DAPS method on the PointDA-10 dataset by comparing our domain adaptive point sampling (DAPS) method with the farthest point sampling (FPS) method and random point sampling (RPS) method on the PointDA-10 dataset. Note that we perform DAPS twice with a sampling ratio of 0.9, meaning we use around 81% points for our extend-DGCNN training. Therefore, we set the sampling ratio of FPS and RPS as 0.81. We also include the results of supervised learning methods that directly train the model with only labelled source data for comparison (denoted as "w/o DA"). The Oracle method that trains the model using labelled target data is also listed for comparison (denoted as "Oracle"). For each method, we run the experiments on each domain adaptation scenario for 3 rounds. The recognition accuracy and standard error of the mean (SEM) of each method on each scenario are reported in Table 4.4. Note that all of the experiments are conducted without pseudo-labelled target data fine-tuning.

We can see that our DAPS outperforms FPS and RPS by a large margin on all 6 domain adaptation scenarios. The main reason is that when performing farthest point sampling or random point sampling, not only

Table 4.5: The overall classification accuracies when using different DAPS-DGCNN training stages on the PointDA-10 dataset.

# DAPS-DGCNN training stages (*i.e.*, K)	Avg.
w/o DAPS-DGCNN training	69.1
$K = 1$	69.7
$K = 2$	70.8
$K = 3$	70.3
$K = 4$	69.9

the points that contain domain-specific information will be sampled out, but some points that contain domain-invariant information which is useful for cross-domain recognition will also be sampled out, leading to the recognition performance decrease. Our DAPS method will only sample out those points containing domain-specific information and those points containing domain-invariant information will remain, thus improving the cross-domain recognition performance of our model. Moreover, it can be inferred from the table that the performance of the FPS method and the w/o DA method is not much different, the main reason is that the points in the PointDA-10 dataset are obtained in order using the FPS method. Sampling 1024 points or 830 points through the FPS method will not have a large impact on the overall geometry of the point cloud. On the contrary, the RPS method will outperform the FPS method, the main reason is that the RPS method is prone to miss the points distributing on the sparse parts like the chair legs, thus the remained points after sampling will be more likely distributed on the cushion of the chairs, reducing the domain gap in a degree, but still, RPS method is far worse than our DAPS method.

We finally verify the importance of our progressive representation learning strategy on the PointDA-10 dataset. We conduct the experiments by varying the number of DAPS-DGCNN training stages K with a fixed sampling ratio 0.9, *i.e.*, we sort the similarity between the feature of each point and the global feature according to Eq. (4.2) and only keep the top 90% points as the representative points. The results are listed in Table 4.5, we report the mean accuracies when training DAPS without adopting the SPL strategy over 6 cases and we also discard the SEM

results. We can see that when we only train the last expand-DGCNN without using DAPS, *i.e.*, "w/o DAPS-DGCNN training", the overall accuracy is only 69.1%. After applying our DAPS strategy, we can see even when we only perform DAPS once, (namely, we only filter out 10% noisy points), we can achieve a performance improvement of 0.6%. When $K = 2$, the model can achieve the best performance, with the mean accuracy of 70.8%, where around 20% points are filtered. When we filter out too many points, (*i.e.*, the number of training iterations is larger than 2), the performance drops, as the remained points are too few to contain useful semantic information for recognition.

Visualization Results

In this section, we illustrate some visualizations of whether applying our domain adaptive sampling (DAPS) method on point cloud samples from different datasets in PointDA-10 dataset *i.e.*, ModelNet-10, ScanNet-10 and ShapeNet-10, as shown in Figure 4.3.

We observe that the shape of the legs of most chair samples from the ModelNet-10 dataset varies a lot. For example, some chairs have legs that look like octopuses, and some other chairs have integrated legs, as shown in Figure 4.3 (a). But most point clouds of the chairs from the ShapeNet-10 dataset have four legs, as shown in Figure 4.3 (b). Moreover, most of the chair samples from the ScanNet-10 have some legs missing, and some of them even may have no legs visible, as shown in Figure 4.3 (c). These multiple forms of chair legs may confuse the model severely. However, the design of the cushions and backrests of the chair samples is similar across different datasets, and they are easy to classify from other categories. Our DAPS method can successfully sample out the points distributed on the cushions or backrests and drop the points distributed on the legs, thus the remained representative points share a similar geometric structure across different datasets and are easy to classify. Thus improving the cross-domain recognition performance.

The category lamp has a similar question, where the design of the lamp holder of different samples from different datasets is quite different, for example, most of the lamp holders from the ModelNet-10

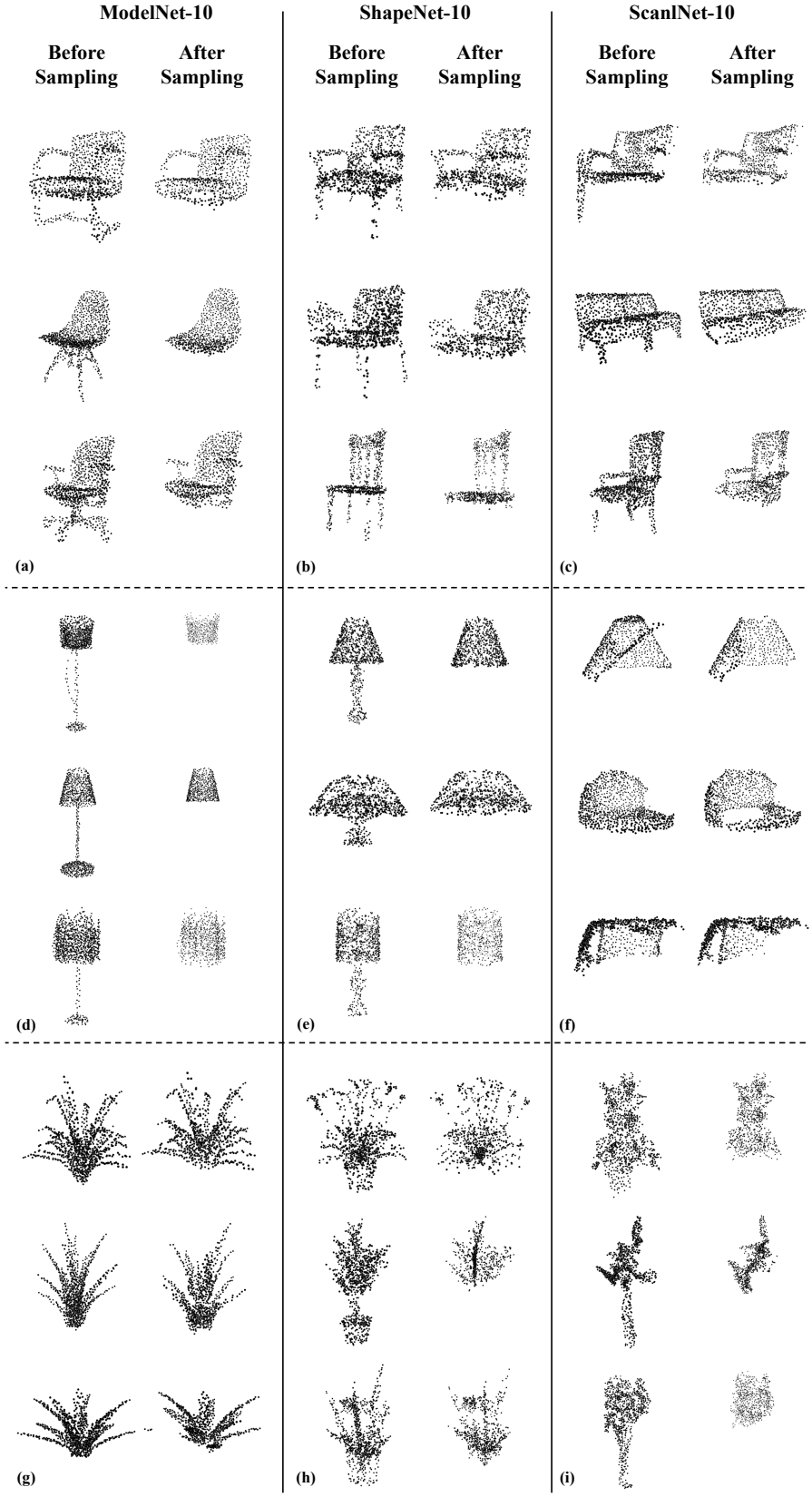

Figure 4.3: The visualization of the point cloud with or without using our domain adaptive point sampling (DAPS) method on different samples.

are quite simple (see Figure 4.3 (d)) while some lamp holders from the ShapeNet-10 have complex structure (see Figure 4.3 (e)), and most of the samples from the ScanNet-10 may miss parts of all of the lamp holders due to the occlusion or noise (see Figure 4.3 (f)). However, the shape of the lampshades is usually curved. Our DAPS method can focus on the domain-invariant structure of lampshades and sample out the point distributed on the lampshades for classification.

Moreover, we also observe that some samples of plant from ModelNet-10 have no vase (see Figure 4.3 (g)), while most point clouds of plants from ShapeNet-10 have complete vases (see Figure 4.3 (h)) and plant point clouds from ScanNet-10 usually have parts of the vase missing (see Figure 4.3 (i)). The existence of the vase may cause the attention of the model to be different. However, we also observe that the shape of the vase is also curved and can be indistinguishable from the lampshade, which may also degrade the recognition performance of the model. Our DAPS method can filter out those points distributed on the vase and focus on the representative points on the branches and leaves of the plants, which can also represent the plants and are easy to classify from samples belonging to other categories.

4.3 Summary

In this chapter, we have proposed a new domain adaptive point sampling (DAPS) strategy for cross-domain point cloud recognition. Our DAPS strategy selects representative points within each point cloud and uses the sampled points to learn robust domain-invariant representations based on a new progressive learning pipeline. Comprehensive experiments on two benchmark datasets have demonstrated the effectiveness of our newly proposed DAPS strategy. However, as our DAPS method can only extract domain-invariant features, the lack of target-specific information might influence the cross-domain recognition performance.

Chapter 5

Instance-Level Domain Adaptive Cloud Sampling for Cross-Domain Point Cloud Recognition

In this chapter, we propose an instance-level domain adaptive cloud sampling (DACS) strategy which can be combined with the point-level domain adaptive point sampling (DAPS) strategy to learn target-specific information. Our proposed DACS strategy is based on the self-paced learning (SPL) paradigm, where we select a set of pseudo-labelled target point clouds to train our designed light-weighted adapters without modifying the learnt domain-invariant representation. We validate our DACS method on the benchmark datasets, *i.e.*, PointDA-10 and GraspNetPC-10, and demonstrate the effectiveness of our method.

5.1 Motivations and Contributions

In Chapter 4, we have proposed the DAPS strategy to extract domain-invariant representations. However, even though the point-level divergence is reduced, the feature extracted with the learnt model can hardly be optimal for point cloud recognition on the target domain, as the domain-invariant features might lack important target-specific information. Most of the works focusing on the cross-domain point cloud recognition task

ignored the issue and only a few recent works [101, 69, 22] noticed the problem. However, these works only use a pseudo-labelling method to fine-tune the learnt mode, while the training is unstable and may suffer from the confirmation bias problem, *i.e.*, the model would be misled by the wrongly selected pseudo-labels and degrade the recognition performance. Moreover, the learnt domain-invariant features may also be disturbed [90, 84, 82, 47].

Adapters have been widely used in various knowledge transfer tasks in the 2D or the NLP domain [34, 81, 55, 41, 3]. Most of the works use a set of light-weighted adapters for fine-tuning the large pre-trained model to various downstream tasks [34, 81]. More recently, some works proposed to learn a set of task-specific adapters based on the pre-trained task-agnostic model to filter useful knowledge from the learnt model to new tasks or domains [41, 3]. These works have shown a great transfer ability to learn task-specific information with adapters based on the learnt task-agnostic information. The adapter structure has the potential to learn target-specific information without modifying the learnt domain-invariant information. However, how to effectively integrate adapter modules into 3D feature extractors to solve the 3D UDA problem is still unknown. Therefore, in this work, we aim to design a proper adapter architecture that is suitable for cross-domain 3D point cloud recognition by learning target-specific information based on the learnt domain-invariant information.

Therefore, based on the model learnt in the DAPS stage, we further propose a domain adaptive cloud sampling (DACS) strategy to gradually learn target-specific information with the selected confidently predicted samples from the target domain using a self-paced learning (SPL) paradigm. Specifically, we design a set of light-weighted adapters as add-ons to the initial learnt model. Then, we fix the learnt model and gradually use the predictions of the confidently predicted samples as the pseudo-labels to train the adapters. In this way, the adapter-based model would be gradually drawn away from the source domain, and move close to the target domain, while preventing the vanilla model from being disturbed. In this way, the model will learn potential target-specific

information in addition to the domain-invariant features learnt by the DAPS model.

The main contributions in this chapter can be summarised as follows: (1) We propose a new adapter architecture to learn target-specific information. (2) We propose a new domain adaptive cloud sampling (DACS) method based on the adapter architecture to train the recognition network for unsupervised domain adaptation on point clouds. (3) Our proposed DACS method can be combined with the domain adaptive point sampling (DAPS) strategy for better cross-domain point cloud recognition. Comprehensive experiments on benchmark datasets have demonstrated the effectiveness of our newly proposed domain adaptive cloud sampling method.

5.2 Methodology

In this section, we will introduce our newly proposed domain adaptive cloud sampling (DACS) method in detail. In particular, our DACS is based on a self-paced learning paradigm (SPL) with several training circles. Each training circle can be further divided into a target pseudo-label selection operation and an adapter training operation, as shown in Figure 5.1 (a). In particular, we come up with a variation of the DGCNN [83], *i.e.*, Adapter-DGCNN for adapter training, (see Figure 5.1 (b)) and adapter-based EdgeConv architecture (see Figure 5.1 (c)).

5.2.1 Target-Specific Information Learning through Domain Adaptive Cloud Sampling

Target Pseudo-Label Selection

The category labels of the target samples are absent, so we use the predictions of the samples to generate pseudo-labels for the target point clouds. Then, we select the pseudo-labelled target point clouds with high prediction scores to fine-tune the model. Given a target point cloud $\mathcal{P}_n^t$, its pseudo-label can be written as $\hat{y}_n^t = \arg\max_c \Phi_{[c]}(\mathcal{P}_n^t)$, where $\Phi_{[c]}(\mathcal{P}_n^t)$ is the c-th dimension of the prediction scores of $\mathcal{P}_n^t$, and $c = \{1, \ldots, C\}$

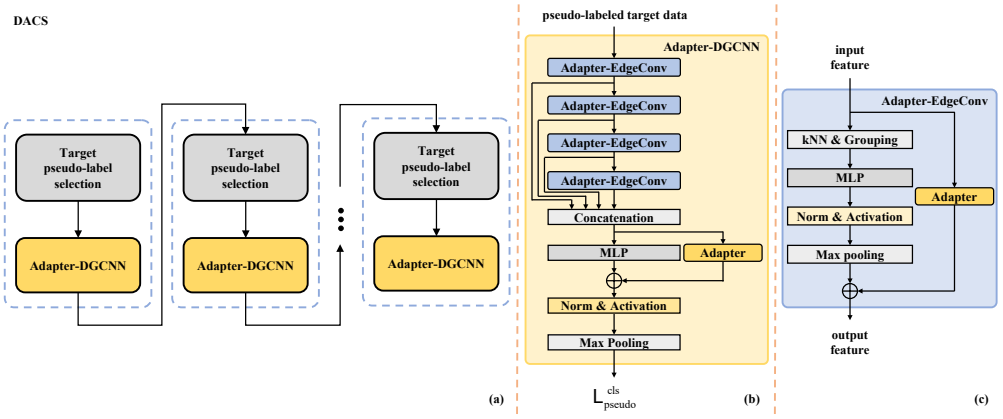

Figure 5.1: Overview of our domain adaptive cloud sampling (DACS) module. (a) Our DACS method is based on a self-paced learning paradigm (SPL). During each training circle in SPL, we select confidently predicted target samples as pseudo-labelled samples and use the pseudo-labelled samples to train the adapter module in adapter-DGCNN so as to better exploit target-specific information. During adapter-DGCNN training, the parameters of other modules except the adapters are fixed. (b) The detailed network structure of our proposed Adapter-DGCNN. (c) The detailed network structure of our proposed Adapter-EdgeConv.

represents the index of the categories. We apply the cross-entropy loss to fine-tune the model, and the classification loss in the target domain can be formulated as below:

$$\tilde{\mathcal{L}}_t^{\text{cls}} = -\frac{1}{|\hat{\mathcal{T}}|} \sum_{(\mathcal{P}_n^t, \hat{y}_n^t) \in \hat{\mathcal{T}}} \ell^{ce}(\Phi(\mathcal{P}_n^t), \hat{y}_n^t) \tag{5.1}$$

where $\hat{\mathcal{T}} = \{(\mathcal{P}_n^t, \hat{y}_n^t) | \max_c \Phi_{[c]}(\mathcal{P}_n^t) > \varepsilon, \forall n = 1, \ldots, N_t\}$ is the set of sampled target point clouds together with their pseudo-labels, and ε is a pre-defined threshold.

In addition, we follow [101] and use an easy-to-hard training strategy and gradually raise the threshold for selecting the confidently predicted samples, *i.e.* we train the network with a fixed ε for each circle

(*i.e.*, a few epochs) and update $\varepsilon = \varepsilon + \Delta$ at the end of each circle. Intuitively, the model will gradually move close to the target domain during the training procedure, and the prediction scores of the accurately predicted target samples will be higher. Therefore, it is beneficial to use a higher threshold in later circles, which can help us to focus on those most reliable pseudo-labelled target samples that can better adapt the model to the target domain [101, 96].

Adapter Training

When using the selected pseudo-labelled target point clouds to fine-tune the model, it is unrealistic to fine-tune the whole model with a large number of parameters if the number of the selected pseudo-labelled target data is too small. Moreover, the inaccurate pseudo-labels can make the model training unstable. We tackle these problems by only training a set of light-weighted adapters without updating the whole learnt model. In this way, the adapters can learn target-specific information in addition to the domain-invariant features learnt in the DAPS stage. Each adapter has a similar network architecture, *i.e.*, an MLP that maps the input features with a dimension of D_1 to a low dimension D_a and another MLP that maps the features from D_a to an output dimension D_2, where $D_1 * D_a + D_a * D_2$ is smaller than $D_1 * D_2$. During training, we only keep the vanilla DGCNN model in expand-DGCNN and introduce the light-weighted adapters to short-circuit each neural unit in the vanilla DGCNN network, including the MLP in each EdgeConv layer and the final MLP operation. Let $\Phi_{\mathrm{f},l}$ denote the l-th EdgeConv layer in vanilla DGCNN and $\Phi_{\mathrm{f},cat}$ be the final mapping MLP operation. Similarly, let f_l represent the output feature of the l-th EdgeConv layer while f_{cat} be the concatenated features from all four EdgeConv layers. We rename EdgeConv with adapters as Adapter-EdgeConv and also rename the adapter for the l-th Adapter-EdgeConv layer as $\Phi_{\mathrm{ad},l}$, we illustrate the architecture of each Adapter-EdgeConv in Figure 5.1 (c). Also, we rename the adapter paralleled with $\Phi_{\mathrm{f},cat}$ as $\Phi_{\mathrm{ad},cat}$, then we can calculate the f_l outputs by each Adapter-EdgeConv and the f_{cat} outputs by $\Phi_{\mathrm{f},cat}$ as:

$$f_l = \Phi_{\mathrm{f},l}(f_{l-1}) + \Phi_{\mathrm{ad},l}(f_{l-1}) \tag{5.2}$$

and

$$f_{cat} = \Phi_{\mathrm{f},cat}(f_{cat}) + \Phi_{\mathrm{ad},cat}(f_{cat}) \tag{5.3}$$

During fine-tuning, we fix the parameters of the vanilla DGCNN and only train the adapters. In this way, the adapters can learn extra target-specific information using the selected pseudo-labelled target data based on the domain-invariant feature learnt with DAPS. Moreover, the hidden layer dimension of the adapter is small, therefore, the amount of tuning parameters is low, which makes it easy to learn the robust adapters using a small number of selected pseudo-labelled target samples.

5.2.2 Experiments

In this section, we first list the implementation details of our DACS method. We further evaluate our DACS on the two benchmark datasets. In addition, we analyse the effectiveness of the detailed modules of our DACS strategy. Finally, we present some visualization results to verify the effectiveness of our DACS strategy.

Implementation Details

Here, we follow the previous works [1, 101, 69] and adopt DGCNN as the backbone. The hidden layer dimension of each adapter is set to 32. Our methods are trained on the server with four NVIDIA RTX A6000 GPUs, and our implementation is based on the PyTorch framework. During training, we use the Adam optimizer together with an epoch-wise cosine annealing learning rate scheduler in which we set the learning rate to 0.001 and the weight decay as 0.00005. We set the initial threshold ε for determining the pseudo-labels as 0.8 and the increasing step Δ as 0.01. With the batch size of 18, we fine-tune the adapters for 10 epochs within one circle except for the first circle, where we train the adapters for 50 epochs as the adapters are randomly initialised in the first circle. The DACS training stage consists of 10 circles in total.

Table 5.1: The classification accuracies (mean ± SEM) of different methods over 3 rounds of experiments on the GraspNetPC-10 dataset. The numbers in the brackets denote the year of the compared methods.

Methods	Syn. → Kin.	Syn. → RS	Kin. → RS	RS → Kin.	Avg.
Oracle (2019) [83]	97.2 ± 0.8	95.6 ± 0.4	95.6 ± 0.3	97.2 ± 0.4	96.4
w/o DA	61.3 ± 1.0	54.4 ± 0.9	53.4 ± 1.3	68.5 ± 0.5	59.4
Baseline	80.3 ± 2.3	66.3 ± 2.0	65.5 ± 0.5	73.0 ± 1.8	71.3
DANN (2016) [26]	78.6 ± 0.3	70.3± 0.5	46.1 ± 2.2	67.9 ± 0.3	65.7
PointDAN (2019) [61]	77.0 ± 0.2	72.5 ± 0.3	65.9 ± 1.2	82.3 ± 0.5	74.4
RS (2019) [68]	67.3 ± 0.4	58.6 ± 0.8	55.7 ± 1.5	69.6 ± 0.4	62.8
DefRec+PCM (2021) [1]	80.7 ± 0.1	70.5 ± 0.4	65.1 ± 0.3	77.7 ± 1.2	73.5
GAST (2021) [101]	81.3 ± 1.8	72.3 ± 0.8	61.3 ± 0.9	80.1 ± 0.5	73.8
GAI (2022) [69]	94.6 ± 0.4	80.5 ± 0.2	76.8 ± 0.4	85.9 ± 0.3	84.4
DACS	86.3 ± 0.9	70.5 ± 0.4	71.5 ± 1.7	80.0 ± 0.8	77.1
DAS	**97.2** ± 0.1	**84.4** ± 1.6	**79.9** ± 0.4	**97.0** ± 0.7	**89.6**

Experimental Results

We compare our proposed DACS approach with the recent SOTA point cloud-based unsupervised domain adaptation methods, including PointDAN [61], RS [68], DefRec with PCM [1], GAST [101] and the latest GAI [69] and GLRV [22]. Moreover, we include the popular 2D UDA method DANN [26] for a fair comparison. In addition, we also include the results of supervised learning methods that directly train the model with only labelled source data for comparison (denoted as "w/o DA"). The Oracle method [83] that trains the model by using labelled target data is also listed for comparison (denoted as "Oracle"). Moreover, as most of the recent 3D UDA methods are based on a self-paced learning paradigm, we construct another baseline by first training the model with labelled source data which is then fine-tuned with the pseudo-labelled target data based on the self-paced learning paradigm (denoted as "Baseline"). It should be noticed that we name the combination of our proposed DAPS strategy and DACS method as domain adaptive sampling (DAS) and our DACS is based on a pre-trained w/o DA model.

We first validate our method on the newly proposed GraspNetPC-10 dataset. For each method, we run the experiments on each domain adaptation scenario for 3 rounds. The recognition accuracy and standard error of the mean (SEM) of each method on each scenario are reported in Table 5.1. Here we use 'Syn.', 'Kin.' and 'RS' to denote the synthetic

domain, the Kinect real-world domain, and the Realsense real-world domain, respectively. For example, Syn. $\rightarrow$ Kin. means we use the synthetic domain as the source domain and the Kinect real-world domain as the target domain. We observe that our proposed DAS method achieves the state-of-the-art (SOTA) results on GraspNetPC-10, including all 4 domain adaptation scenarios and the overall performance, exceeding the current SOTA method GAI by a notable margin of 5.2%. Specifically, when using the Kinect real-world domain (Kin.) as the target domain, our proposed DAS method can approach the upper bound, *i.e.*, the Oracle method, demonstrating the effectiveness of our proposed approach. Moreover, as shown in the table, our DACS method can also exceed the baseline, *i.e.*, the Baseline method, by a large margin.

Table 5.2: The classification accuracies (mean $\pm$ SEM) of different methods over 3 rounds of experiments on the PointDA-10 dataset. The numbers in the brackets denote the year of the compared methods.

Methods	M $\rightarrow$ S	M $\rightarrow$ S*	S $\rightarrow$ M	S $\rightarrow$ S*	S* $\rightarrow$ M	S* $\rightarrow$ S	Avg.
Oracle (2019) [83]	93.9 $\pm$ 0.2	78.4 $\pm$ 0.6	96.2 $\pm$ 0.1	78.4 $\pm$ 0.6	96.2 $\pm$ 0.1	93.9 $\pm$ 0.2	89.5
w/o DA	83.3 $\pm$ 0.7	43.8 $\pm$ 2.3	75.5 $\pm$ 1.8	42.5 $\pm$ 1.4	63.8 $\pm$ 3.9	64.2 $\pm$ 0.8	62.2
Baseline	85.4 $\pm$ 0.3	56.9 $\pm$ 0.2	76.5 $\pm$ 0.2	53.5 $\pm$ 0.7	76.9 $\pm$ 0.4	73.6 $\pm$ 1.3	70.5
DANN (2016) [26]	74.8 $\pm$ 2.8	42.1 $\pm$ 0.6	57.5 $\pm$ 0.4	50.9 $\pm$ 1.0	43.7 $\pm$ 2.9	71.6 $\pm$ 1.0	56.8
PointDAN (2019) [61]	83.9 $\pm$ 0.3	44.8 $\pm$ 1.4	63.3 $\pm$ 1.1	45.7 $\pm$ 0.7	43.6 $\pm$ 2.0	56.4 $\pm$ 1.5	56.3
RS (2019) [68]	79.9 $\pm$ 0.8	46.7 $\pm$ 4.8	75.2 $\pm$ 2.0	51.4 $\pm$ 3.9	71.8 $\pm$ 2.3	71.2 $\pm$ 2.8	66.0
DefRec+PCM (2021) [1]	81.7 $\pm$ 0.6	51.8 $\pm$ 0.3	78.6 $\pm$ 0.7	54.5 $\pm$ 0.3	73.7 $\pm$ 1.6	71.1 $\pm$ 1.4	68.6
GAST (2021) [101]	84.8 $\pm$ 0.1	59.8 $\pm$ 0.2	80.8 $\pm$ 0.6	56.7 $\pm$ 0.2	81.1 $\pm$ 0.8	74.9 $\pm$ 0.5	73.0
GLRV (2022) [22]	85.4 $\pm$ 0.4	60.4 $\pm$ 0.4	78.8 $\pm$ 0.6	57.7 $\pm$ 0.4	77.8 $\pm$ 1.1	76.2 $\pm$ 0.6	72.7
GAI (2022) [69]	86.2 $\pm$ 0.2	58.6 $\pm$ 0.1	81.4 $\pm$ 0.4	56.9 $\pm$ 0.2	81.5 $\pm$ 0.5	74.4 $\pm$ 0.6	73.2
MLSP (2022) [44]	85.7 $\pm$ 0.6	59.4 $\pm$ 1.3	82.3 $\pm$ 0.9	57.3 $\pm$ 0.7	82.2 $\pm$ 0.5	76.4 $\pm$ 0.5	73.8
DACS	85.5 $\pm$ 0.3	57.2 $\pm$ 0.3	76.6 $\pm$ 0.6	54.6 $\pm$ 1.2	80.0 $\pm$ 0.7	77.9 $\pm$ 1.6	71.9
DAS	**87.2** $\pm$ 0.9	**60.5** $\pm$ 0.2	**82.4** $\pm$ 0.7	**58.1** $\pm$ 0.8	**84.8** $\pm$ 2.3	**82.3** $\pm$ 1.5	**75.9**

We further validate our method on the widely used PointDA-10 dataset. We follow a similar setting to the experiments on the GraspNetPC-10 dataset by running the experiments on each domain adaptation scenario for 3 rounds. The recognition accuracy and standard error of the mean (SEM) of each method on each scenario are reported in Table 5.2. Here we use 'M', 'S' and 'S*' to denote the ModelNet-10, ShapeNet-10, and ScanNet-10 subsets, respectively. For example, M $\rightarrow$ S means we use ModelNet-10 as the source domain and ShapeNet-10 as the target domain. We observe that our proposed method achieves the state-of-the-art (SOTA) results on PointDA-10, including all 6 domain adaptation scenarios and the overall performance. With the help of both the DAPS

and DACS modules, we achieve an average accuracy of 75.9%, which outperforms the current SOTA method MLSP by a notable margin of 2.1%. The results clearly demonstrate the effectiveness of our proposed approach.

Specifically, we observe that our DACS method outperforms the Baseline method on all 6 domain adaptation scenarios, especially when using S^* as the source domain. A possible explanation is that the samples from the target domains are synthetic and are easy to be recognised in this situation, thus we can learn plentiful target-specific information from the pseudo-labelled target data, demonstrating the effectiveness of our proposed DACS module.

Ablation Study

Table 5.3: Ablation study on the effectiveness of our newly proposed adapter architecture.

# dimension of the hidden layer (*i.e.*, D_a)	Avg.
DAPS	70.8
$D_a = 4$	73.7
$D_a = 8$	74.1
$D_a = 16$	75.0
$D_a = 32$	**75.9**
$D_a = 64$	75.5

In this section, we first analyse the effectiveness of our newly proposed adapter architecture. We vary the dimension of the hidden layer in the adapters to verify how a light-weighted adapter architecture can be beneficial to target-specific information learning. We conduct the experiments for 3 rounds on 6 domain adaptation scenarios on PointDA-10. We conduct the experiments by varying the dimension of the hidden layer, D_a, in the adapters and the results are listed in Table 5.3, here we report the mean accuracies. It should be noticed that our DACS method is based on the learnt model by DAPS. Therefore, we also list the results without using our DACS method, *i.e.*, DAPS, as the baseline result.

We can observe that when the dimension of the adapters is too small, *e.g.*, $D_a = 4$, the adapters would be too light-weighted to capture useful

semantic information, leading to poor cross-domain recognition performance. However, when the dimension of the adapters is too large, *e.g.*, $D_a = 64$, the amount of the parameters would be large, and the selected pseudo-labels would be too little to fine-tune such a complex module, leading to a poor fine-tune result. When the dimension of the adapters is 32, we can have the best cross-domain recognition performance.

Visualization Results

To further investigate the capacity of our proposed domain adaptive sampling (DAS) approach for addressing the domain shift issue on point clouds, in Figure 5.2 we visualise the features of point clouds from the target domain by using t-SNE [76], where different colours denote different classes. Here we take two difficult domain adaptation scenarios in GraspNetPC-10 as examples, *i.e.*, Syn-to-RS and Kin-to-RS. Due to the domain shift, we observe that the target features extracted by the w/o DA method are less discriminative, and sometimes the instances from different categories are mixed with each other. With our DAS strategy, the learnt features of the target samples are more discriminative, which leads to a huge performance improvement for point cloud recognition on the target domain.

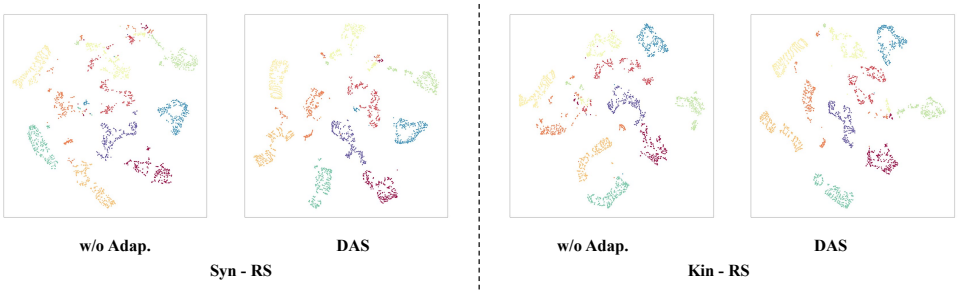

Figure 5.2: The t-SNE visualization results of the target domain samples with or without using our domain adaptive sampling strategy (DAS).

5.3 Summary

In this chapter, we have proposed a new domain adaptive cloud sampling (DACS) method for cross-domain point cloud recognition. Our DACS method conducts domain adaptive cloud sampling by using the newly proposed adapter structure to learn target-specific information, such that the model can be better adapted to the target domain for point cloud recognition. Comprehensive experiments on two benchmark datasets have demonstrated the effectiveness of our newly proposed DACS method. However, the introduction of the adapters will bring additional parameters, which may increase the dependence of the model on the GPU memory.

Chapter 6

Open-Set Domain Adaptive Sampling for Open-Set Cross-Domain Point Cloud Recognition

In Chapter 4 and Chapter 5, we have proposed a new domain adaptive sampling (DAS) method, including a point-level domain adaptive point sampling (DAPS) strategy and an instance-level domain adaptive cloud sampling (DACS) method, for cross-domain point cloud recognition. In this chapter, we extend our DAS method and propose a new open-set domain adaptive sampling (OS-DAS) method for open-set cross-domain point cloud recognition, which includes an open-set domain adaptive point sampling (OS-DAPS) module and an open-set domain adaptive cloud sampling (OS-DACS) module. In particular, we first use OS-DAPS to enable the model to extract domain-invariant features from each sample to learn a coarse classifier. Then, we select two sets of confidently predicted target samples belonging to the source-known classes and the target-specific class according to the entropy of the predictions, respectively. After the pseudo-labelled sample selection module, we use the predictions of the selected target samples as pseudo-labels to fine-tune the model using our proposed OS-DACS method. We validate our OS-DAS strategy on the benchmark dataset PointDA-10 to demonstrate the effectiveness of our method.

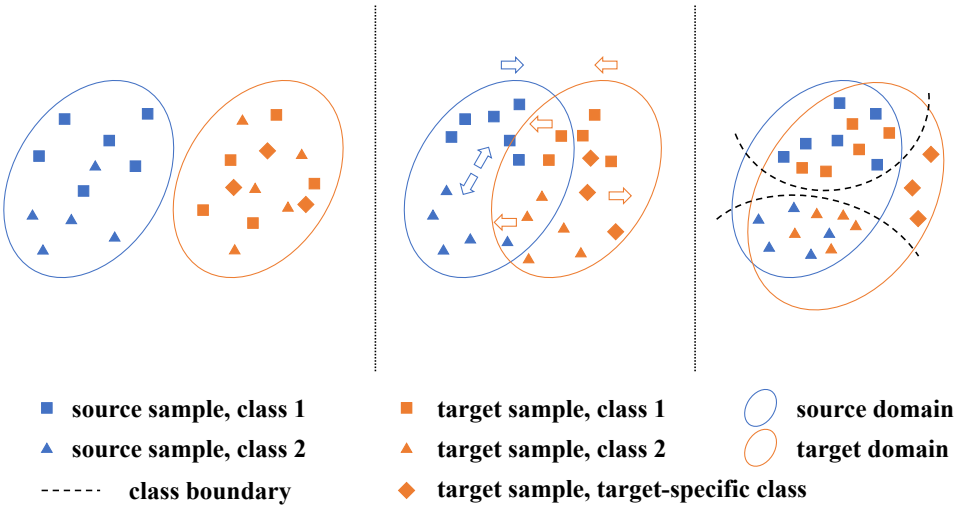

Figure 6.1: Visualization of the process of minimizing the domain discrepancy in the open-set scenario.

6.1 Motivations and Contributions

Existing unsupervised cross-domain point cloud recognition approaches only aim at solving the close-set domain adaptation problem, *i.e.*, the source domain and the target domain share the same categories. However, in practical applications, it is unrealistic that the samples from the source domain and the target domain belong to the same classes. The model would inevitably come across several unknown classes when applied to a new scenario and the learnt model is prone to generate incorrect predictions from those samples belonging to a source-unknown category. Therefore, it is necessary to train a model to identify whether the target sample belongs to a certain source-known class or the source-unknown class. This is known as the open-set domain adaption (OSDA) problem.

The main difficulties of the OSDA problem lie in two manifolds, the first one is how to separate the source-known classes and the target-specific class and the second one is how to align the target samples with the source samples belonging to the source-known classes, as shown in Fig. 6.1. OSBP [67] is a classic adversarial-based domain adaption method to solve the open-set cross-domain recognition problem in the

2D domain. However, such an adversarial-based method is not suitable for 3D domain alignment. In contrast, our proposed domain adaptive sampling (DAS) method in Chapter 4 and Chapter 5 can efficiently tackle the domain gap from a self-supervised learning paradigm, which is irrelevant to the specific category.

Therefore, in this work, we aim at extending our DAS method to tackle the open-set domain adaptation issue. Specifically, we first use our proposed open-set domain adaptive point sampling (OS-DAPS) strategy to sample a domain-invariant structure for each point cloud and extract features from the sampled domain-invariant structures. Then, we learn a coarse classifier to map the features extracted to corresponding categories. It should be noticed that only the labels from the source domain are available while the target domain contains an unknown class. We argue that the classifier learnt by the annotated source data can also recognise the samples from the source-known classes belonging to the target domain since our OS-DAPS method will reduce the domain gap. Therefore, during the training process, we treat all samples from the target domain as belonging to the target-specific class and reweight the classification loss to learn a coarse classifier for coarse recognition. Furthermore, we argue that the entropy calculated by the predictions of the samples belonging to the target-specific class is higher than the entropy calculated by the predictions of the samples from the source-known classes. Therefore, we select a set of confidently predicted samples from the predictions with low entropy, which we believe belong to the source-known classes, and we also select another set of samples with high entropy, which we believe belong to the target-specific category. We use the predictions of these selected samples as pseudo-labels, and finally, we use our proposed open-set domain adaptive cloud sampling (OS-DACS) strategy to fine-tune the model to obtain precise predictions.

The main contributions in this chapter can be summarised as follows: (1) We propose a new open-set domain adaptive sampling (OS-DAS) for open-set cross-domain point cloud recognition. (2) We propose a new training strategy for learning a coarse recognition model which can extract domain-invariant representations and distinguish target-specific

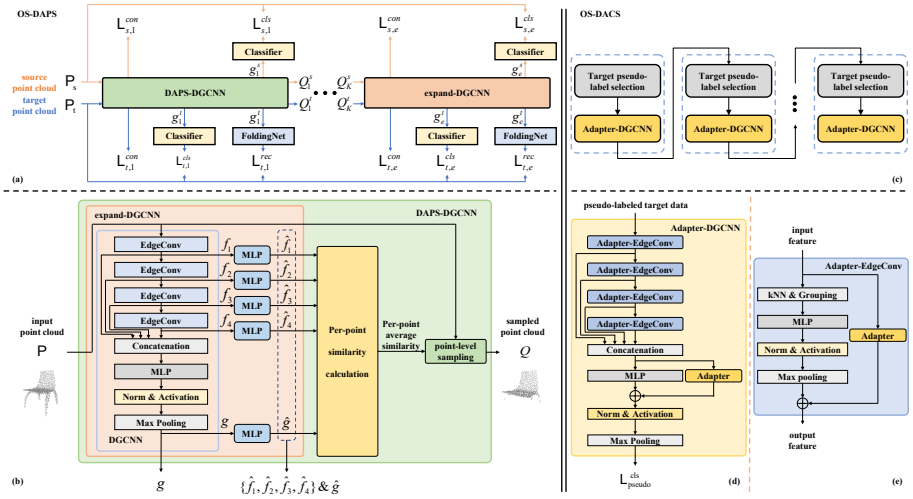

Figure 6.2: Overview of our open-set domain adaptive sampling (OS-DAS) method. Our OS-DAS method combines an open-set domain adaptive point sampling (OS-DAPS) module and an open-set domain adaptive cloud sampling (OS-DACS) module. (a) Our OS-DAPS method trains a coarse classification model to roughly distinguish whether the sample belongs to the source-known classes or the target-specific class. (b) The detailed network structure of our proposed DAPS-DGCNN, expand-DGCNN and vanilla DGCNN. (c) Our OS-DACS method is based on a self-paced learning paradigm (SPL) for fine-grained recognition. (d) The detailed network structure of our proposed Adapter-DGCNN. (e) The detailed network structure of our proposed Adapter-EdgeConv.

samples. (3) We propose a new entropy-based pseudo-label selection algorithm for model fine-tuning to get a fine recognition performance. Comprehensive experiments on the benchmark dataset have demonstrated the effectiveness of our newly proposed OS-DAS method.

6.2 Methodology

In this section, we will introduce our newly proposed open-set domain adaptive sampling strategy (OS-DAS) in detail. Our OS-DAS method is based on the domain adaptive point sampling (DAPS) strategy illustrated in Chapter 4 and the domain adaptive cloud sampling (DACS) method introduced in Chapter 5. The pipeline of our OS-DAS method is shown in Figure 6.2. We will only briefly introduce our DAPS and

DACS here and we mainly show the difference between our OS-DAS and DAS in this chapter.

6.2.1 Open-Set Domain Adaptive Point Sampling

As introduced before, there are two core difficulties in OSDA, the first one is the separation of the source-known classes and the target-specific class, and the second one is the domain alignment of the source-known categories. Recall that as illustrated in Chapter 4, our proposed DAPS strategy aims at searching for a domain-invariant structure for each sample to perform a low-level domain alignment so as to extract domain-invariant features, which performs domain alignment in a self-supervised manner. Intuitively, the model is prone to distinguish those commonly-seen geometric structures from those more unique structures, and these commonly-seen geometric structures are mostly shared across different domains, indicating that our DAPS strategy can efficiently shorten the domain discrepancy of the source-known categories.

However, it is still difficult to separate the source-known classes and the target-specific class. Inspired by OSBP, we hereby set a pseudo-label $\hat{y}$ for all of the samples belonging to the target domain as $C+1$ and we train a classifier based on the extracted domain-invariant features to roughly distinguish whether the sample belongs to the source-known classes or the target-specific class. The only difference between our OS-DAPS and our DAPS is Eq. 6.1 where $\mathcal{L}^{\mathrm{OS-DAPS}}$ can be calculated as:

$$\mathcal{L}^{\mathrm{OS-DAPS}} = \lambda_1 \mathcal{L}^{\mathrm{cls}} + \lambda_2 \mathcal{L}^{\mathrm{con}} + \lambda_3 \mathcal{L}^{\mathrm{rec}} + \lambda_4 \mathcal{L}^{\mathrm{pseudo}} \tag{6.1}$$

and

$$\mathcal{L}^{\mathrm{pseudo}} = -\frac{1}{N_t} \sum_{n=1}^{N_t} \ell^{ce}(\Phi(\mathcal{P}_n^t), \hat{y}_n^t) \tag{6.2}$$

We argue that training with $\mathcal{L}^{\mathrm{cls}}$ enables the model to generate the decision boundary of the source-known classes as the classifier maps the high-dimensional domain-invariant features to the corresponding categories. Training with $\mathcal{L}^{\mathrm{pseudo}}$ will do harm to the decision boundary

of the source-known classes as the pseudo-labels set for the target samples belonging to the source-known classes are also set as $C+1$, but it will also contribute to generating the decision boundary of the target-specific class. Moreover, the harm to decision boundaries for source-known classes can be mitigated by reweighing the weight of the pseudo-labelling loss function $\mathcal{L}^{\text{pseudo}}$. Intuitively, a small weight λ_4 for $\mathcal{L}^{\text{pseudo}}$ will mitigate the influence of the damage to the decision boundaries for source-known classes and a high weight λ_4 for $\mathcal{L}^{\text{pseudo}}$ will contribute to generating the decision boundary for the target-specific class.

The training process of our OS-DAPS is similar to the training of our DAPS, and we omit the details in this chapter.

6.2.2 Open-Set Domain Adaptive Cloud Sampling

Our proposed OS-DAPS enables the model to extract domain-invariant features for samples belonging to different domains and also searches for a coarse decision boundary for samples belonging to the source-known classes and the target-specific class. However, the decision boundary is not that precise. Therefore, we further propose a new open-set domain adaptive cloud sampling (OS-DACS) method which is based on our proposed OS-DAPS to refine the decision boundary. The training process of our OS-DACS is listed in Fig. 6.2 (c), which is similar to the training of our DACS. In particular, we propose a new entropy-based target pseudo-label selection algorithm for a better open-set cross-domain point cloud recognition performance.

Recall that a small weight λ_4 for $\mathcal{L}^{\text{pseudo}}$ will mitigate the influence of the damage to the decision boundaries for source-known classes and can also contribute to generating the decision boundary for the target-specific class in a degree. We argue that the decision boundary is not precise and only a portion of the samples can be correctly classified. However, the target labels cannot be reached, indicating that which target samples can be correctly classified is unknown. To tackle the issue, in this section, we propose an entropy-based target pseudo-label selection algorithm to select a set of target samples with low entropy and use the

predictions as the pseudo-labels of the source-known classes. We also select a set of target samples with high entropy and use the predictions as the pseudo-labels of the target-specific class.

Given a target point cloud $\mathcal{P}_n^t$, the prediction scores of the point clouds can be written as $y_n^t = \Phi_{[c]}(\mathcal{P}_n^t)$, where y_n^t is a $C+1$-dimensional vector and $y_{n,c}^t$ indicates the prediction score of the c-th dimension. Thus, we can get the pseudo label $\hat{y}_n^t$ and the entropy E_n^t as:

$$\hat{y}_n^t = \arg\max_c \Phi_{[c]}(\mathcal{P}_n^t) \tag{6.3}$$

and

$$E_n^t = \sum_{c=1}^{C+1} -y_{n,c}^t \times \log y_{n,c}^t \tag{6.4}$$

According to the predicted pseudo-labels of the samples and the corresponding calculated entropy, we divide samples into $C+1$ memory banks $\mathcal{M} = \{\mathcal{M}_c\}_{c=1}^{C+1}$. Recall that there are C source-known classes across domains and 1 target-specific class, we use the first C memory banks to save the target samples with the corresponding pseudo-labels and the entropy values, *i.e.*, $\mathcal{M}_c = \{(\mathcal{P}_n^t, \hat{y}_n^t, E_n^t) | \hat{y}_n^t = c, \forall n = 1, \ldots, N_t, c = 1, \ldots, C\}$. The last memory saves all of the target samples, *i.e.*, $\mathcal{M}_{C+1} = \{(\mathcal{P}_n^t, \hat{y}_n^t, E_n^t), \forall n = 1, \ldots, N_t\}$. We argue that the predictions generated by the model are not that accurate. On one hand, a small set of confidently predicted samples belonging to the source-known classes can be correctly classified. These samples are usually accompanied by low entropy values. On the other hand, the model cannot generate precise predictions for the samples belonging to the target-specific class, but these samples are easily to be misclassified into other classes. Thereby, the entropy value of these samples is relatively high. Based on the observations, we sample out two sets of target samples and use the predictions as the pseudo-labels to fine-tune the model. The first set of samples includes the samples belonging to the first C memory banks with low entropy values. The second set of samples includes the samples belonging to the last memory bank with high entropy values. In this work, we determine the number of the samples selected as 10% of the size of the corresponding memory bank.

When using the two sets of selected pseudo-labelled target point clouds to fine-tune the model, we follow the same fine-tuning strategy as our DACS introduced in Chapter 5 and we omit the details in this chapter.

6.2.3 Experiments

In this section, we first list the implementation details of our OS-DAS method. We further evaluate our OS-DAS method on the benchmark dataset. Finally, we analyse the effectiveness of the detailed module design of our OS-DAS method.

Implementation Details

Here, we follow the previous works [1, 101, 69] and adopt DGCNN as the backbone. We use two-layer MLPs with a hidden layer dimension of 512 to map the output features of each EdgeConv layer as well as the global feature to the same dimension of 512. The hidden layer dimension of each adapter is set to 32. Our methods are trained on the server with four NVIDIA RTX A6000 GPUs, and our implementation is based on the PyTorch framework. During training, we use the Adam optimizer together with an epoch-wise cosine annealing learning rate scheduler in which we set the learning rate to 0.001 and the weight decay as 0.00005. We set the hyper-parameters λ_1, λ_2, λ_3 and λ_4 to 1.0, 1.0, 1.0 and 0.2, respectively. Moreover, we set the number of DAPS-DGCNN training stages (*i.e.*, K) as 2 and the sampling ratio as 0.9, which means we perform domain adaptive sampling for twice and use the sampled points to train the expand-DGCNN. We train our OS-DAPS module for 100 epochs on the PointDA-10 dataset with a batch size of 18. In our OS-DACS module, we set the initial threshold ε for determining the pseudo labels as 0.8, and the increasing step Δ as 0.01 after each training circle. With the batch size of 18, we fine-tune the adapters for 10 epochs within one circle except for the first circle we train the adapters for 50 epochs as the adapters are randomly initialised in the first circle, and the OS-DACS training stage consists of 10 circles in total.

Experimental Results

We first come up with a baseline method by training a DGCNN with $\mathcal{L}^{\text{cls}}$ and $\mathcal{L}^{\text{pseudo}}$ (denoted as "Baseline"), then we re-implement the classic 2D open-set domain adaptation approach OSBP [67] for comparison.

For each method, we run the experiments on each domain adaptation scenario for 3 rounds. The recognition accuracy and standard error of the mean (SEM) of each method on each scenario are reported in Table 6.1. Here we use 'M', 'S' and 'S*' to denote the ModelNet-10, ShapeNet-10, and ScanNet-10 subsets, respectively. For example, M $\rightarrow$ S means we use ModelNet-10 as the source domain and ShapeNet-10 as the target domain.

We observe that our proposed method achieves the state-of-the-art (SOTA) results on PointDA-10, including all 6 domain adaptation scenarios and the overall performance. With the help of both the OS-DAPS and OS-DACS modules, we achieve an average accuracy of 62.7%, which outperforms OSBP by a notable margin of 20% and surpasses the Baseline method by 23.2%. The results clearly demonstrate the effectiveness of our proposed approach.

Table 6.1: The classification accuracies (mean $\pm$ SEM) of different methods over 3 rounds of experiments on the PointDA-10 dataset. The numbers in the brackets denote the year of the compared methods.

Methods	M $\rightarrow$ S	M $\rightarrow$ S*	S $\rightarrow$ M	S $\rightarrow$ S*	S* $\rightarrow$ M	S* $\rightarrow$ S	Avg.
Baseline	30.8 $\pm$ 0.2	42.6 $\pm$ 0.4	42.5 $\pm$ 0.4	48.0 $\pm$ 0.7	42.3 $\pm$ 1.3	30.5 $\pm$ 1.1	39.5
OSBP (2018) [67]	42.0 $\pm$ 0.4	47.1 $\pm$ 1.8	44.5 $\pm$ 0.3	49.2 $\pm$ 0.4	35.3 $\pm$ 2.1	36.7 $\pm$ 2.7	42.6
Ours (w/o DACS)	64.3 $\pm$ 0.4	48.2 $\pm$ 1.2	56.0 $\pm$ 0.3	56.5 $\pm$ 0.8	51.9 $\pm$ 1.1	70.6 $\pm$ 0.7	57.9
Ours	**70.2** $\pm$ 0.6	**49.7** $\pm$ 0.8	**63.3** $\pm$ 0.2	**58.2** $\pm$ 0.9	**54.2** $\pm$ 1.6	**80.5** $\pm$ 0.4	**62.7**

To further analyse the two modules, we also report the results using only the OS-DAPS module (*i.e.*, Ours (w/o OS-DACS)). We observe that our OS-DAPS method can also gain reasonable performance improvement over the Baseline method, validating the effectiveness of each individual module.

It should be mentioned that when using the ScanNet-10 (*i.e.*, S*) as the source domain and using the ShapeNet-10 (*i.e.*, S) as the target domain, the results of our OS-DAS approach surpasses the baseline method

by a large margin of around 50%. One main reason is that our method performs a self-supervised domain alignment, which is agnostic to the specific category. Therefore, our method can correctly classify the samples belonging to the source-known classes. Another reason is our entropy-based pseudo-label selection mechanism and our adapter-based fine-tuning strategy can refine the recognition model.

Ablation Study

In this part, we analyse the effectiveness of $\mathcal{L}^{\text{pseudo}}$ by varying the weight λ_4 in our open-set domain adaptive point sampling (OS-DAPS) method. We report the classification accuracies over 3 rounds of experiments on the PointDA-10 dataset. We can observe that a higher weight λ_4 will degrade the recognition performance. On one hand, a high λ_4 will force the model to treat all of the target samples as belonging to the target-specific category, thus damaging the decision boundary learnt by the $\mathcal{L}^{\text{cls}}$ and enabling the model to misclassify more samples to the target-specific category. On the other hand, a low λ_4 can hardly influence the decision boundary learnt by the $\mathcal{L}^{\text{cls}}$, thus the model will misclassify the samples to the source-known categories and can hardly distinguish the samples that belong to the target-specific class.

Table 6.2: Ablation study on the effectiveness of our open-set domain adaptive point sampling (OS-DAPS) method.

λ_4	0.1	0.2	0.3	0.4	0.5	1.0
Avg.	56.8	57.9	57.1	55.9	54.7	50.8

6.3 Summary

In this chapter, we have extended our proposed domain adaptive sampling (DAS) method to the open-set scenario and proposed a new open-set domain adaptive sampling (OS-DAS) approach. First, our OS-DAPS approach learns a coarse open-set classifier and then our DACS method refines the classifier with the selected pseudo-labelled target samples according to the entropy of the predictions. Comprehensive experiments on the benchmark dataset have demonstrated the effectiveness of our

newly proposed OS-DAS approach. However, our OS-DAS method is sensitive to the weights of the losses. Therefore, the performance of the model on different datasets using the same set of parameters may vary a lot.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

With the development of deep neural network (DNN) techniques, deep learning (DL) methods on point clouds have shown great potential in various vision tasks like autonomous driving and robotics. However, the success of DL methods is mainly due to the huge amount of annotated datasets and it takes great effort to collect precisely labelled data for training the networks. Moreover, a learnt model on one dataset may not generalise well on another one. These factors influence the implementation of DL methods. Therefore, domain adaptation methods on point clouds are attracting increasing attention. In this thesis, we have proposed two domain adaptation methods for cross-domain point cloud recognition, and we further extend our method to the open-set scenario. In this chapter, we conclude the contributions of this thesis.

The contributions of this thesis are summarised as follows:

- We have proposed a new point-level domain adaptive point sampling (DAPS) strategy for cross-domain point cloud recognition. Our proposed DAPS strategy enhances the domain-invariant representation of point clouds by progressively focusing on representative points within each point cloud based on geometric consistency.
- We have proposed a new instance-level domain adaptive cloud sampling (DACS) method for cross-domain point cloud recognition. Our DACS method learns target-specific information based

on a self-paced learning paradigm, where we select a set of useful pseudo-labelled target point clouds to train our designed light-weighted adapters without modifying the learnt domain-invariant representation.

- We have proposed a new two-stage open-set domain adaptive sampling (OS-DAS) for open-set cross-domain point cloud recognition. Our OS-DAS method learns an open-set recognition model in a coarse-to-fine manner. We first learn a coarse recognition model by performing shared categories domain alignment in a self-supervised manner. Then we select two sets of pseudo-labelled target samples based on the entropy of the predictions to fine-tune the recognition model. Our OS-DAS method can efficiently classify the samples belonging to the source-known classes as well as the target-specific class.

We have conducted extensive experiments on benchmark datasets to evaluate the effectiveness of our DAPS, DACS and OS-DAS approaches.

7.2 Future Work

There are three potential research directions in future work: (1) Few-shot cross-domain point cloud recognition; (2) Cross-domain point cloud recognition with noisy labels; and (3) Domain generalisation on the point cloud.

Few-shot cross-domain point cloud recognition. In this thesis, we only consider the domain adaptation problem where all samples from the source domain are annotated. However, the annotation of the source domain also takes great effort. Therefore, how to learn a recognition network with only a few annotated source data and large-scale unlabelled source and target data becomes an urgent task, which has not been explored so far.

Cross-domain point cloud recognition with noisy labels. In this thesis, we only consider the domain adaptation problem where all of the samples from the source domain are correctly annotated. However, is it

inevitable that some samples are mislabelled. Therefore, it is important to learn a recognition network with noisy annotated source data and unlabelled target data, which has also not been explored.

Domain generalisation on the point cloud. In this thesis, we only consider the domain adaptation problem where we can access the target data. However, in some scenarios, the target data might be unavailable and we also need the model to have a good recognition performance on any given target domain, which is known as the domain generalisation (DG) problem. This is a promising direction to explore but only a few works have studied this direction.

In conclusion, domain adaptation on point clouds is an urgent but not fully explored research topic. We believe that with the development of domain adaptation methods, the implementation of deep learning methods on point clouds will be accelerated in the future.

Bibliography

[1] I. Achituve, H. Maron, and G. Chechik. "Self-supervised learning for domain adaptation on point clouds". In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2021, pp. 123–133.

[2] H. Bao, L. Dong, S. Piao, and F. Wei. "Beit: Bert pre-training of image transformers". In: *arXiv preprint arXiv:2106.08254* (2021).

[3] P. Bateni, R. Goyal, V. Masrani, F. Wood, and L. Sigal. "Improved few-shot visual classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 14493–14502.

[4] A. Cardace, R. Spezialetti, P. Z. Ramirez, S. Salti, and L. Di Stefano. "Self-Distillation for Unsupervised 3D Domain Adaptation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 4166–4177.

[5] 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.

[6] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. "Emerging properties in self-supervised vision transformers". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9650–9660.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), pp. 834–848.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. "Semantic image segmentation with deep convolutional nets and fully connected crfs". In: *arXiv preprint arXiv:1412.7062* (2014).

[9] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. "Rethinking atrous convolution for semantic image segmentation". In: *arXiv preprint arXiv:1706.05587* (2017).

[10] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 801–818.

[11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[12] X. Chen and K. He. "Exploring simple siamese representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15750–15758.

[13] Z. Chen, S. Gu, G. Lu, and D. Xu. "Exploiting intra-slice and inter-slice redundancy for learning-based lossless volumetric image compression". In: *IEEE Transactions on Image Processing* 31 (2022), pp. 1697–1707.

[14] Z. Chen, G. Lu, Z. Hu, S. Liu, W. Jiang, and D. Xu. "LSVC: a learning-based stereo video compression framework". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 6073–6082.

[15] 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.

[16] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li. "Voxel r-cnn: Towards high performance voxel-based 3d object detection". In: *arXiv preprint arXiv:2012.15712* 1.2 (2020), p. 4.

[17] Z. Deng, Y. Luo, and J. Zhu. "Cluster alignment with a teacher for unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 9944–9953.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[19] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen, et al. "Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models". In: *arXiv preprint arXiv:2203.06904* (2022).

[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[21] H. Fan, H. Su, and L. J. Guibas. "A point set generation network for 3d object reconstruction from a single image". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 605–613.

[22] H. Fan, X. Chang, W. Zhang, Y. Cheng, Y. Sun, and M. Kankanhalli. "Self-Supervised Global-Local Structure Modeling for Point Cloud Domain Adaptation With Reliable Voted Pseudo Labels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 6377–6386.

[23] H.-S. Fang, C. Wang, M. Gou, and C. Lu. "Graspnet-1billion: A large-scale benchmark for general object grasping". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11444–11453.

[24] Q. Feng, G. Kang, H. Fan, and Y. Yang. "Attract or distract: Exploit the margin of open set". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 7990–7999.

[25] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. "Dual attention network for scene segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 3146–3154.

[26] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. "Domain-adversarial training of neural networks". In: *The journal of machine learning research* 17.1 (2016), pp. 2096–2030.

[27] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. "Deep Reconstruction-Classification Networks for Unsupervised Domain Adaptation". In: *Proceedings of the European conference on computer vision (ECCV)*. 2016, pp. 597–613.

[28] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. "Bootstrap your own latent-a new approach to self-supervised learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21271–21284.

[29] M.-H. Guo, C.-Z. Lu, Q. Hou, Z. Liu, M.-M. Cheng, and S.-M. Hu. "SegNeXt: Rethinking Convolutional Attention Design for Semantic Segmentation". In: *arXiv preprint arXiv:2209.08575* (2022).

[30] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. "Masked autoencoders are scalable vision learners". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16000–16009.

[31] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738.

[32] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[33] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. "Cycada: Cycle-consistent adversarial domain adaptation". In: *International conference on machine learning*. Pmlr. 2018, pp. 1989–1998.

[34] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. "Parameter-efficient transfer learning for NLP". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2790–2799.

[35] C. Huang, Z. Cao, Y. Wang, J. Wang, and M. Long. "Metasets: Meta-learning on point sets for generalizable representations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8863–8872.

[36] S. Huang, J. Ma, G. Han, and S.-F. Chang. "Task-adaptive negative envision for few-shot open-set recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7171–7180.

[37] M. Jing, J. Li, L. Zhu, Z. Ding, K. Lu, and Y. Yang. "Balanced open set domain adaptation via centroid alignment". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 9. 2021, pp. 8013–8020.

[38] G. Kang, L. Zheng, Y. Yan, and Y. Yang. "Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 401–416.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[41] W.-H. Li, X. Liu, and H. Bilen. "Cross-domain Few-shot Learning with Task-specific Adapters". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7161–7170.

[42] X. Li, H. Zhao, L. Han, Y. Tong, S. Tan, and K. Yang. "Gated fully fusion for semantic segmentation". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 07. 2020, pp. 11418–11425.

[43] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai. "BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers". In: *arXiv preprint arXiv:2203.17270* (2022).

[44] H. Liang, H. Fan, Z. Fan, Y. Wang, T. Chen, Y. Cheng, and Z. Wang. "Point Cloud Domain Adaptation via Masked Local 3D Structure Prediction". In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*. Springer. 2022, pp. 156–172.

[45] H. Liu, Z. Cao, M. Long, J. Wang, and Q. Yang. "Separate to adapt: Open set domain adaptation via progressive separation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2927–2936.

[46] J. Liu, J. Guo, and D. Xu. "GeometryMotion-Transformer: An End-to-End Framework for 3D Action Recognition". In: *IEEE Transactions on Multimedia* (2022).

[47] Y. Liu, Y. Tian, Y. Chen, F. Liu, V. Belagiannis, and G. Carneiro. "Perturbed and strict mean teachers for semi-supervised semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 4258–4267.

[48] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. "Swin transformer: Hierarchical vision transformer using shifted windows". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10012–10022.

[49] Z. Liu, Z. Zhang, Y. Cao, H. Hu, and X. Tong. "Group-free 3d object detection via transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2949–2958.

[50] X. Luo, S. Liu, K. Fu, M. Wang, and Z. Song. "A learnable self-supervised task for unsupervised domain adaptation on point clouds". In: *arXiv preprint arXiv:2104.05164* (2021).

[51] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu. "Rethinking network design and local geometry in point cloud: A simple residual MLP framework". In: *arXiv preprint arXiv:2202.07123* (2022).

[52] D. Maturana and S. Scherer. "Voxnet: A 3d convolutional neural network for real-time object recognition". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 922–928.

[53] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. "DINOv2: Learning Robust Visual Features without Supervision". In: *arXiv preprint arXiv:2304.07193* (2023).

[54] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan. "Masked autoencoders for point cloud self-supervised learning". In: *arXiv preprint arXiv:2203.06604* (2022).

[55] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. "Film: Visual reasoning with a general conditioning layer". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[56] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych. "AdapterFusion: Non-destructive task composition for transfer learning". In: *arXiv preprint arXiv:2005.00247* (2020).

[57] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. "Pointnet: Deep learning on point sets for 3d classification and segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 652–660.

[58] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas. "Volumetric and multi-view cnns for object classification on 3d data". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 5648–5656.

[59] 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* 30 (2017).

[60] Z. Qi, R. Dong, G. Fan, Z. Ge, X. Zhang, K. Ma, and L. Yi. "Contrast with Reconstruct: Contrastive 3D Representation Learning Guided by Generative Pretraining". In: *arXiv preprint arXiv:2302.02318* (2023).

[61] C. Qin, H. You, L. Wang, C.-C. J. Kuo, and Y. Fu. "Pointdan: A multi-scale 3d domain adaption network for point cloud representation". In: *Advances in Neural Information Processing Systems* 32 (2019).

[62] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. "Learning transferable visual models from natural language supervision". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763.

[63] Y. Rao, J. Lu, and J. Zhou. "Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 5376–5385.

[64] Y. Rao, J. Lu, and J. Zhou. "PointGLR: Unsupervised Structural Representation Learning of 3D Point Clouds". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[65] D. Rukhovich, A. Vorontsova, and A. Konushin. "TR3D: Towards Real-Time Indoor 3D Object Detection". In: *arXiv preprint arXiv:2302.02858* (2023).

[66] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. "Maximum classifier discrepancy for unsupervised domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3723–3732.

[67] K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada. "Open set domain adaptation by backpropagation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 153–168.

[68] J. Sauder and B. Sievers. "Self-supervised deep learning on point clouds by reconstructing space". In: *Advances in Neural Information Processing Systems* 32 (2019).

[69] Y. Shen, Y. Yang, M. Yan, H. Wang, Y. Zheng, and L. J. Guibas. "Domain Adaptation on Point Clouds via Geometry-Aware Implicits". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7223–7232.

[70] 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. 10529–10538.

[71] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[72] A. C. Stickland and I. Murray. "Bert and pals: Projected attention layers for efficient adaptation in multi-task learning". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5986–5995.

[73] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[74] J. Tian, J. Zhang, W. Li, and D. Xu. "Vdm-da: Virtual domain modeling for source data-free domain adaptation". In: *IEEE Transactions on Circuits and Systems for Video Technology* (2021).

[75] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. "Adversarial discriminative domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7167–7176.

[76] L. Van der Maaten and G. Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[77] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[78] K. V. Vishwanath, D. Gupta, A. Vahdat, and K. Yocum. "Modelnet: Towards a datacenter emulation environment". In: *2009 IEEE Ninth International Conference on Peer-to-Peer Computing*. IEEE. 2009, pp. 81–82.

[79] F. Wang, W. Li, and D. Xu. "Cross-dataset point cloud recognition using deep-shallow domain adaptation network". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 7364–7377.

[80] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, et al. "Deep high-resolution representation learning for visual recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 43.10 (2020), pp. 3349–3364.

[81] R. Wang, D. Tang, N. Duan, Z. Wei, X. Huang, G. Cao, D. Jiang, M. Zhou, et al. "K-adapter: Infusing knowledge into pre-trained models with adapters". In: *arXiv preprint arXiv:2002.01808* (2020).

[82] Y. Wang, H. Wang, Y. Shen, J. Fei, W. Li, G. Jin, L. Wu, R. Zhao, and X. Le. "Semi-Supervised Semantic Segmentation Using Unreliable Pseudo-Labels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 4248–4257.

[83] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. "Dynamic graph cnn for learning on point clouds". In: *Acm Transactions On Graphics (tog)* 38.5 (2019), pp. 1–12.

[84] Z. Wang, Z. Zhao, L. Zhou, D. Xu, X. Xing, and X. Kong. "Conflict-Based Cross-View Consistency for Semi-Supervised Semantic Segmentation". In: *arXiv preprint arXiv:2303.01276* (2023).

[85] X. Wei, X. Gu, and J. Sun. "Learning Generalizable Part-based Feature Representation for 3D Point Clouds". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 29305–29318.

[86] X. Wu, L. Peng, H. Yang, L. Xie, C. Huang, C. Deng, H. Liu, and D. Cai. "Sparse fuse dense: Towards high quality 3d detection with depth completion". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5418–5427.

[87] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. "3d shapenets: A deep representation for volumetric shapes". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1912–1920.

[88] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. "SegFormer: Simple and efficient design for semantic segmentation with transformers". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 12077–12090.

[89] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu. "Simmim: A simple framework for masked image modeling". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9653–9663.

[90] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao. "St++: Make self-training work better for semi-supervised semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 4268–4277.

[91] Y. Yang, C. Feng, Y. Shen, and D. Tian. "Foldingnet: Point cloud auto-encoder via deep grid deformation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 206–215.

[92] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu. "Point-bert: Pre-training 3d point cloud transformers with masked point modeling". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 19313–19322.

[93] Y. Yuan, R. Fu, L. Huang, W. Lin, C. Zhang, X. Chen, and J. Wang. "Hrformer: High-resolution vision transformer for dense predict". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 7281–7293.

[94] X. Yue, Z. Zheng, S. Zhang, Y. Gao, T. Darrell, K. Keutzer, and A. S. Vincentelli. "Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13834–13844.

[95] B. Zhang, Z. Tian, Q. Tang, X. Chu, X. Wei, C. Shen, and Y. Liu. "SegViT: Semantic Segmentation with Plain Vision Transformers". In: *arXiv preprint arXiv:2210.05844* (2022).

[96] W. Zhang, W. Ouyang, W. Li, and D. Xu. "Collaborative and adversarial network for unsupervised domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3801–3809.

[97] W. Zhang, D. Xu, W. Ouyang, and W. Li. "Self-Paced Collaborative and Adversarial Network for Unsupervised Domain Adaptation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.6 (2021), pp. 2047–2061. DOI: 10.1109/TPAMI.2019.2962476.

[98] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun. "Point transformer". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 16259–16268.

[99] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. "Pyramid scene parsing network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2881–2890.

[100] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, et al. "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 6881–6890.

[101] L. Zou, H. Tang, K. Chen, and K. Jia. "Geometry-Aware Self-Training for Unsupervised Domain Adaptation on Object Point Clouds".

In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6403–6412.