UAB
Universitat Autònoma de Barcelona

Towards Source-Free Domain Adaption of Neural Networks in an Open World

Shiqi Yang

Universitat Autònoma de Barcelona

Towards Source-Free Domain Adaption of Neural Networks in an Open World

A dissertation submitted by **Shiqi Yang** to the Universitat Autònoma de Barcelona in fulfilment of the degree of **Doctor of Philosophy** in the Departament de Ciències de la Computació.

Bellaterra, May 11, 2023

Director

Dr. Joost van de Weijer
Centre de Visió per Computador
Universitat Autònoma de Barcelona

Thesis committee

Dr. Yannis Kalantidis
NAVER LABS Europe

Dr. Elisa Ricci
Department of Information Engineering and Computer Science
University of Trento

Dr. Antonio Lopez
Centre de Visió per Computador
Universitat Autònoma de Barcelona

This document was typeset by the author using LaTeX 2$_{\varepsilon}$.

ISBN: 978-84-126409-3-9

Printed by Ediciones Gráficas Rey, S.L.

Acknowledgements

First I would like to express my sincere gratitude to my supervisor Dr. Joost van de Weijer, for his invaluable supervision and kind academic support during my whole PhD research. His insightful advice and rigorous academic attitude guide me to head for an independent and excellent researcher.

I would like to also thank all staffs in my institute, Computer Vision Center. Their jobs ensure a supportive and wonderful research environment for we phd students. Their eagerness and generosity for both research and non-research support are invaluable for those overseas students and researchers like me. I am also grateful for the research financial support from my supervisor and my institute, which allow me to expand my research impact, and participate in international conferences and other academic activities to build academic connections, and also widen my horizon.

I would like to thank all my coauthors during my phd research, the collaboration and discussion are always informative, and always lead to interesting idea which results in amazing research output.

I am also grateful to my friends in Spain, China and Japan, who have provided me with emotional support throughout these 4 years. Their friendship supports me through the challenging period of the PhD journey. Finally, I would like to greatly thank my family for their love and support throughout my whole life path. Their unwavering belief in my abilities and consistent love are always the strong support.

In conclusion, I would like to express my sincere appreciation to those who have contributed to my academic career, those who have accompanied in my personal life. Their kind support and encouragement are invaluable in this journey, which finally enable me to arrive in the current stage in both academic and personal path.

Abstract

Though they achieve great success, deep neural networks typically require a huge amount of labeled data for training. However, collecting labeled data is often laborious and expensive. It would, therefore, be ideal if the knowledge obtained from label-rich datasets could be transferred to unlabeled data. However, deep networks are weak at generalizing to unseen domains, even when the differences are only subtle between the datasets. In real-world situations, a typical factor impairing the model generalization ability is the distribution shift between data from different domains, which is a long-standing problem usually termed as (unsupervised) domain adaptation.

A crucial requirement in the methodology of these domain adaptation methods is that they require access to source domain data during the adaptation process to the target domain. Accessibility to the source data of a trained source model is often impossible in real-world applications, for example, when deploying domain adaptation algorithms on mobile devices where the computational capacity is limited or in situations where data privacy rules limit access to the source domain data. Without access to the source domain data, existing methods suffer from inferior performance. Thus, in this thesis, we investigate domain adaptation without source data (termed as source-free domain adaptation) in multiple different scenarios that focus on image classification tasks.

We first study the source-free domain adaptation problem in a closed-set setting, where the label space of different domains is identical. Only accessing the pretrained source model, we propose to address source-free domain adaptation from the perspective of unsupervised clustering. We achieve this based on nearest neighborhood clustering. In this way, we can transfer the challenging source-free domain adaptation task to a type of clustering problem. The final optimization objective is an upper bound containing only two simple terms, which can be explained as discriminability and diversity. We show that this allows us to relate several other methods in domain adaptation, unsupervised clustering and contrastive learning via the perspective of discriminability and diversity.

Following the source-free domain adaptation setting, we also investigate the catastrophic forgetting issue after adaptation, where the adapted model should keep good performance on the source or all trained domains. To address the forgetting issue, we propose to use randomly generated domain attention masks to regularize the model updating during adaptation. This succeeds to keep the knowledge on old domains while not influence adaptation to new target domains.

In real-world applications, there could be some unseen categories in the target data; without extra processing, the model cannot handle these open classes. To prepare the method to generalize to target environments where there may exist unseen categories, we propose an elegant and simple solution by inserting an additional dimension into the classifier head. Together with an additional cross-entropy loss during source pretraining, the model is empowered with strong open-set recognition performance, which could be directly used for target adaptation and excels at distinguishing open classes during adaptation.

Key words: *source-free domain adaptation, generalized source-free domain adaptation, continual source-free domain adaptation, source-free open-partial domain adaptation*

Resumen

Aunque las redes neuronales profundas logran un gran éxito, suelen requerir una enorme cantidad de datos etiquetados para su entrenamiento. Sin embargo, la recopilación de datos etiquetados a menudo es laboriosa y costosa. Sería ideal si el conocimiento obtenido de conjuntos de datos ricos en etiquetas pudiera transferirse a datos no etiquetados. Sin embargo, las redes profundas son débiles para generalizarse a dominios no vistos, incluso cuando las diferencias entre los conjuntos de datos sean sutiles. En situaciones reales, un factor típico que afecta a la capacidad de generalización del modelo es el cambio de distribución entre los datos de diferentes dominios, lo que es un problema de larga data generalmente denominado adaptación de dominio (no supervisada).

Un requisito crucial en la metodología de estos métodos de adaptación de dominio es que requieren acceso a los datos del dominio fuente durante el proceso de adaptación al dominio objetivo. El acceso a los datos fuente de un modelo fuente entrenado a menudo es imposible en aplicaciones del mundo real, por ejemplo, al implementar algoritmos de adaptación de dominio en dispositivos móviles donde la capacidad computacional es limitada o en situaciones donde las reglas de privacidad de los datos limitan el acceso a los datos del dominio fuente. Sin acceso a los datos del dominio fuente, los métodos existentes sufren un rendimiento inferior. Por lo tanto, en esta tesis, investigamos la adaptación de dominio sin datos fuente (denominada como adaptación de dominio sin fuente) en múltiples escenarios diferentes que se centran en tareas de clasificación de imágenes.

Primero estudiamos el problema de adaptación de dominio sin fuente en un entorno de conjunto cerrado, donde el espacio de etiquetas de diferentes dominios es idéntico. Solo accediendo al modelo fuente pre-entrenado, proponemos abordar la adaptación de dominio sin fuente desde la perspectiva de la agrupación no supervisada. Lo logramos basándonos en la agrupación de vecinos más cercanos. De esta manera, podemos transferir la desafiante tarea de adaptación de dominio sin fuente a un tipo de problema de agrupamiento. El objetivo de optimización final es una cota superior que contiene solo dos términos simples, que pueden explicarse como discriminabilidad y diversidad.

Mostramos que esto nos permite relacionar varios otros métodos en la adaptación de dominio, la agrupación no supervisada y el aprendizaje contrastivo a través de la perspectiva de discriminabilidad y diversidad.

Siguiendo la configuración de adaptación de dominio sin fuente, también investigamos el problema de olvido catastrófico después de la adaptación, donde el modelo adaptado debe mantener un buen rendimiento en el dominio fuente o en todos los dominios entrenados. Para abordar el problema de olvido, proponemos utilizar máscaras de atención de dominio generadas al azar para regularizar la actualización del modelo durante la adaptación. Esto logra mantener el conocimiento de los dominios antiguos sin influir en la adaptación a los nuevos dominios objetivo.

En aplicaciones del mundo real, puede haber algunas categorías no vistas en los datos objetivo; sin un procesamiento adicional, el modelo no puede manejar estas clases abiertas. Para preparar el método para generalizarse a entornos objetivo donde puedan existir categorías no vistas, proponemos una solución elegante y simple mediante la inserción de una dimensión adicional en la cabeza del clasificador. Junto con una pérdida adicional de entropía cruzada durante el preentrenamiento de origen, el modelo está capacitado con un fuerte desempeño de reconocimiento de conjunto abierto, que se puede utilizar directamente para la adaptación del objetivo y sobresale en la distinción de clases abiertas durante la adaptación.

Palabras clave: *Adaptación de dominio sin fuente, adaptación de dominio sin fuente generalizada, adaptación de dominio sin fuente continua, adaptación de dominio parcialmente abierta sin fuente*

Resum

Tot i que aconsegueixen un gran èxit, les xarxes neuronals profundes solen requerir una gran quantitat de dades etiquetades per a la formació. Tanmateix, recollir dades etiquetades sovint és laboriós i costós. Per tant, seria ideal que el coneixement obtingut a partir de conjunts de dades rics en etiquetes es pogués transferir a dades sense etiquetar. Tanmateix, les xarxes profundes són febles per generalitzar-se a dominis invisibles, fins i tot quan les diferències només són subtils entre els conjunts de dades. En situacions del món real, un factor típic que perjudica la capacitat de generalització del model és el canvi de distribució entre dades de diferents dominis, que és un problema de llarga data que se sol denominar adaptació de domini (no supervisada).

Un requisit crucial en la metodologia d'aquests mètodes d'adaptació del domini és que requereixen accés a les dades del domini font durant el procés d'adaptació al domini objectiu. L'accessibilitat a les dades font d'un model font entrenat sovint és impossible en aplicacions del món real, per exemple, quan es desplega algorismes d'adaptació de domini en dispositius mòbils on la capacitat computacional és limitada o en situacions en què les regles de privadesa de dades limiten l'accés a les dades del domini font. . Sense accés a les dades del domini d'origen, els mètodes existents pateixen un rendiment inferior. Així, en aquesta tesi, investiguem l'adaptació del domini sense dades font (anomenada adaptació del domini sense font) en múltiples escenaris diferents que se centren en tasques de classificació d'imatges.

Primer estudiem el problema d'adaptació de dominis sense font en un entorn tancat, on l'espai d'etiquetes de diferents dominis és idèntic. Accedint només al model font preentrenat, proposem abordar l'adaptació del domini sense font des de la perspectiva de l'agrupació no supervisada. Ho aconseguim basant-nos en l'agrupació de barris més propers. D'aquesta manera, podem transferir la difícil tasca d'adaptació del domini sense fonts a un tipus de problema d'agrupació. L'objectiu final d'optimització és un límit superior que conté només dos termes simples, que es poden explicar com a discriminabilitat i diversitat. Mostrem que això ens permet relacionar diversos altres mètodes d'adaptació de dominis, agrupació no supervisada i aprenentatge contrastiu des de la perspectiva de la discriminabilitat i la diversitat.

Seguint la configuració d'adaptació del domini sense font, també investiguem el problema de l'oblit catastròfic després de l'adaptació, on el model adaptat hauria de mantenir un bon rendiment a la font o a tots els dominis entrenats. Per abordar el problema de l'oblit, proposem utilitzar màscares d'atenció de domini generades aleatòriament per regularitzar l'actualització del model durant l'adaptació. Això aconsegueix mantenir el coneixement en dominis antics sense influir en l'adaptació a dominis objectiu nous.

A les aplicacions del món real, podria haver-hi algunes categories no vistes a les dades objectiu; sense processament addicional, el model no pot gestionar aquestes classes obertes. Per preparar el mètode per generalitzar-se a entorns objectiu on hi pugui haver categories no vistes, proposem una solució elegant i senzilla inserint una dimensió addicional al capçal del classificador. Juntament amb una pèrdua d'entropia creuada addicional durant l'entrenament previ a la font, el model té un fort rendiment de reconeixement obert, que es podria utilitzar directament per a l'adaptació d'objectius i destaca per distingir classes obertes durant l'adaptació.

Paraules clau: *adaptació del domini sense font, adaptació generalitzada del domini sense font, adaptació contínua del domini sense font, adaptació del domini obert i parcial sense font*

Contents

List of Figures

List of Tables

1 Introduction

Over the past decade, due to the information explosion in the digital era, artificial intelligence has again emerged as one of the most notable topics in both the academic community and industry. Until now, the successful application cases of artificial intelligence cover a wide range of topics, including computer vision, natural language processing, and speech processing, to name a few. With the recent powerful artificial intelligence tools, the quality of the provided service in above-mentioned application scenarios has risen sharply in recent years, such as face recognition, chatbot services, and image/speech synthesis.

The recent breakthroughs in almost all artificial intelligence fields come from the renaissance of deep learning (deep neural networks), mainly thanks to the rapid expansion of computational hardware and digitization progress. Unlike using handcrafted features, which long dominated progress in the computer vision community, it is possible to train a large model in an end-to-end manner which is capable of automatically learning feature representations. In the last few years, the trend of training large models, which are trained with huge amounts of cross modal data such as billions of images and text data, has becoming more prevalent.

Though we are achieving promising success with deep learning, there are still some barriers hindering the comprehensive deployment of deep learning models in some real-world applications. Training a powerful model typically demands a large amount of labeled data, it is infeasible and expensive to always collect manually labeled data to train a new model for every new upcoming task. It would be ideal if the knowledge obtained on label-rich datasets could be transferred to unlabeled data; in other words, the pretrained model could be transferred to a new environment or task. This is a long-standing problem in machine learning and other related communities, which is usually called transfer learning. Transfer learning aims to transfer the knowledge of a pretrained model to a new environment and another model.

In this thesis, we investigate a sub-topic in transfer learning, *i.e.*, unsupervised domain adaptation. More specifically, we focus on a new paradigm of domain adaptation termed as source-free domain adaptation, where the pretrained model can be efficiently adapted to new target tasks without demanding any labeled data and without access to the labeled source data during the adaptation period.

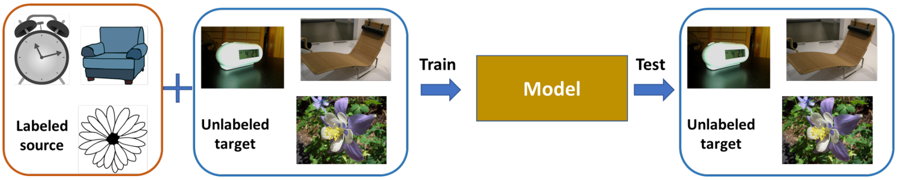

Figure 1.1: Illustration of the classic domain adaptation paradigm. The model is trained with both labeled source data and unlabeled target data, where the label space of the two domains is the same. The goal is to make the model perform well in the target domain after adaptation. In this thesis, we will consider other setups for the domain adaptation problem. Firstly, we will consider excluding source data during the adaptation process to the target data (Section 1.1.1). Secondly, we will consider optimizing for both source and target performance (Section 1.1.2). And thirdly, we will consider non-overlapping label spaces for source and target data (Section 1.1.3).

1.1 Domain Adaptation

In recent years, deep learning models have shown remarkable performance in various domains such as image classification, speech recognition, and natural language processing. However, one of the main challenges of these deep models is their lack of robustness when applied to new or unseen domains. The typical supervised deep neural network is easily overfitting and thus has poor generalization ability to new tasks where there exists a difference between data of the original training data and new data. And in some cases, even when difference is quite small, the deep model will still have degraded performance [97]. This is widely studied in the transfer learning community. Related topics include fine-tuning which typically retrains the model with the new coming labeled data, knowledge distillation [18] which aims to transfer the learned knowledge of an existed model to a new model, multitask learning [128] which lets a single model learn a group of relevant tasks, semi-supervised learning [7] which aims to improve the model performance with abundant unlabeled data, and domain adaptation aiming to address the discrepancy between data from different domains or tasks.

In real-world situations, a typical and most common factor impairing the model's generalization ability is the distribution shift between data from different domains. This problem is known as distributional or domain shift. And in most situations, there may be only a few labeled data samples, since collecting labels is time-consuming and sometimes infeasible in the specific application, or even without any labeled data but quite a lot of unlabeled data. Thus, in this circumstance, it is a desirable property of model's that they can generalize to new tasks or environments where there are only unlabeled data. The specific research direction to address domain shift is domain generalization and unsupervised domain adaptation*. Domain generalization[40, 63] only demands labeled source date, the goal is to drive the source model to produce domain invariant feature representation, and after training the model could be directly deployed to new domains. Domain generalization is a challenging problem as it does not allow the model to adapt to data in the new task. While usually, it is relatively easy to get enough unlabeled data in the new domains or tasks, and it is possible to improve the model with those unlabeled data. To deal with this situation, domain adaptation is proposed.

Domain adaptation aims to address the performance degradation of a model when it is applied to a different unlabeled domain than the labeled one it has been trained on, which is illustrated in Fig. 1.1. Early works [38, 98] learn domain-invariant features based on traditional methods, like kernel method, to link the target domain to the source domain. Along with the growing popularity of deep learning, many works benefit from its powerful representation learning ability for domain adaptation [20, 83, 86, 88, 135, 164]. Those methods typically either try to minimize the distribution discrepancy between two domains [82, 83, 86], or deploy adversarial training [20, 88, 135, 164] to achieve domain invariant feature learning. There are also methods resorting to reconstruction [35], normalization [92] and optimal transport [150]. And some recent methods address domain adaptation by clustering [132] or by exploiting an intermedium domain [94].

In a nutshell, those methods try to find a way to minimize the distance/discrepancy between the labeled source domain and unlabeled target domain, which is usually defined explicitly such as maximal mean discrepancy or optimal transport, or explicitly such as using adversarial training. The forementioned methods mainly focus on image classification tasks, while there is lots of work also studying domain adaptation problem on other tasks, such as segmentation [32, 71, 176], object detection [42, 47, 172] and video understanding [16, 19]. There are also works dealing with data in other modality, such as point cloud [2, 120].

Based on an analysis of the state-of-the-art of domain adaptation, we have identified three main research directions that we pursue in this thesis and which we will outline in the following sections.

1.1.1 Source-free Domain Adaptation

Most of the existing domain adaptation methods assume that the labeled data in the source domain is available during the adaptation phase; in other words, the model is

*In the following text, we will use *domain adaptation* instead of *unsupervised domain adaptation*, since most works that we consider are on unsupervised domain adaptation.

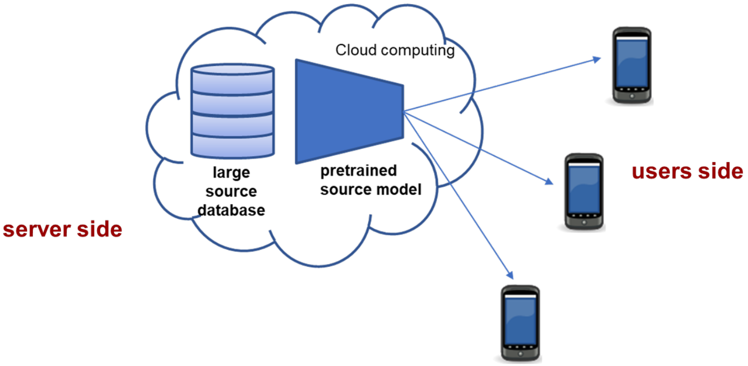

Figure 1.2: Example of an application with source-free domain adaptation. The server will dispatch the pretrained source model to the user sides, which are mobile phones. The user side will conduct the model adaptation with only the local unlabeled data, where there exist domain shifts between the user and server sides. Due to the potential property/privacy issues and the limited computation resource, the user sides cannot access the source images which are on the server side.

trained with both a labeled source domain and an unlabeled target domain. However, this assumption may not always hold, as accessing source data is often impossible in real-world applications, for example, when the algorithm is running on edge devices where there is limited computation capacity to deal with large amounts of source data, or when the labeled source data have some privacy or property issues. A typical real-world application example is shown in Fig. 1.2, where the company dispatches the pretrained model to users, and the model should be adapted to unlabeled user data without access to the large amount of labeled pretraining data. Without access to the source data during adaptation, the existing domain adaptation methods will suffer from inferior performance. To address those mentioned issues, the source-free domain adaptation setting have been proposed in recent years.

Source-free domain adaptation is a challenging problem, as it requires the model to learn from the target domain without any labeled data from the source domain. This problem has attracted a lot of attention from the research community due to its practical applications in various domains, as well as its high practical value in real-world applications.

After receiving the source pretrained model, there are mainly two types of source-free domain adaptation methods. The first type synthesizes some labeled samples [58, 59, 69]. In this way, the model can benefit from the supervision of the generated labeled data. Another type is applying self-training only based on the in-hand unlabeled target

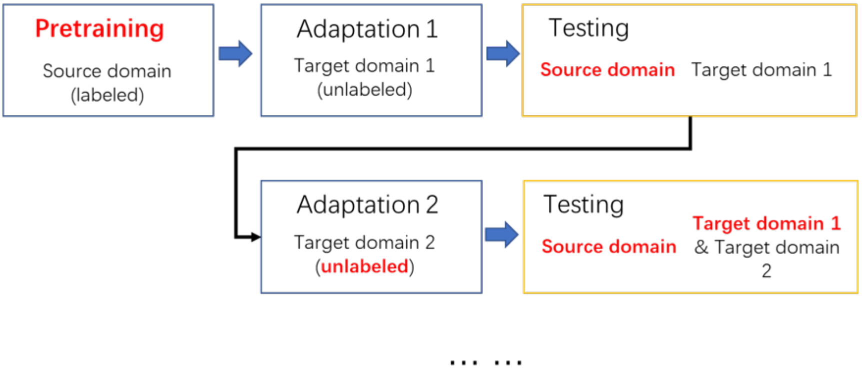

Figure 1.3: Illustration of continual source-free domain adaptation, where the model will be continually adapted to a sequence of unlabeled target domain, and the adapted model is expected to keep good performance on all seen domains.

data. It could be achieved by either finding better pseudo-labels or by clustering, or by combining these two ways together at the same time. But usually existing methods need complex extra modules for feature generation [69] or self-training [149]. *In this thesis, we aim to design computationally efficient methods that exploit the intrinsic neighborhood structure of target data to improve source-free domain adaptation.*

1.1.2 Generalized Source-free Domain Adaptation

If the model is directly adapted to the target domain under the source-free domain adaptation setting, the model may have degraded performance on the source domain. And in some real-world applications, the model will be even adapted to a sequence of unlabeled target domains, which is shown in Fig. 1.3. In this case, the ideal model should not only have satisfactory performance on the current target domain but keep good performance on all seen (source and old target) domains after adaptation, since it is infeasible to deploy one model for each domain, which demands extra computational resources and is not efficient in real deployment.

Current source-free domain adaptation methods focus on the performance of the target domain by fine-tuning the source model, leading to forgetting of old domains. This forgetting issue is typically investigated in the continual learning community, which aims to train a model with a sequence of target tasks and maintain good performance on all seen tasks. But the situation here under source-free domain

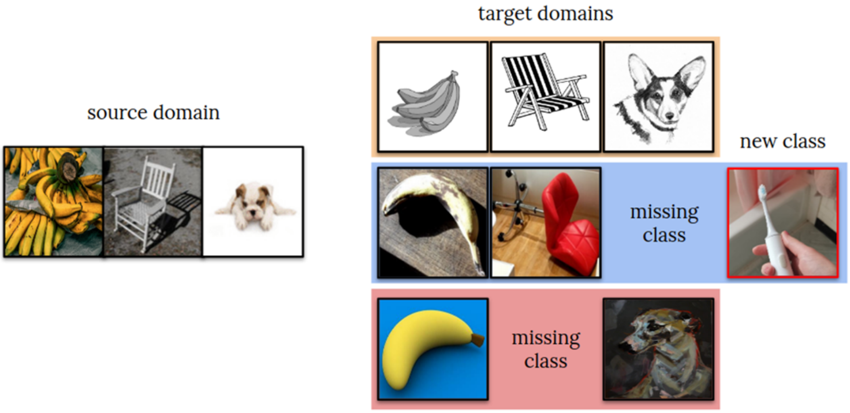

Figure 1.4: Illustration of domain adaptation scenario where the label spaces of source and target domains are not identical, both source and target domains have their own private categories. The image is taken from the VisDA 2021 challenge [5].

adaptation setting is much more challenging compared to typical continual learning scenarios, because in source-free domain adaptation the target domains do not have any labeled data and domain shift exists between different domains. Thus, existing methods cannot directly be deployed to handle the situation described above. *Therefore, in this chapter, we investigate the forgetting issue under source-free domain adaptation setting and aim to develop methods that can adapt to a new domain while maintaining good performance on previous domains.*

1.1.3 Source-Free Domain Adaptation in the Open-World

Besides domain shift, another inevitable obstacle lying on the path to deploying deep learning methods in real-world environments is the presence of potential unseen categories in new domains, since the dataset used for model pretraining cannot cover all possibly appearing categories in the target domain. When there are newly appearing categories in the target domain, the ideal model should distinguish them.

This problem is usually defined as *open-set recognition*(OSR) [15, 33, 95, 126, 131, 139, 162] where the model should be able to reject samples as coming from unseen categories. In recent years, there have emerged several works called novel category discovery [138, 167] or open-world semi-supervised learning [10], which aim to distinguish every unseen categories. Those works do not consider a domain shift between the different domains. Recently, several works also introduce open-set recognition problems (*i.e.*, category shift) into domain adaptation, which are called as *open-set domain adaptation* (OSDA) [9, 27, 28, 54, 79, 99, 116] where target domain has all source categories and will also have some unseen categories, and *universal domain adaptation* (UNDA) or *open-partial domain adaptation* (OPDA) [29, 66, 75, 111, 113, 161] where source and target domains will have some private classes and also some shared classes. In these setting, there is no prior information about which categories are novel or missing. An example of category shift under the domain adaptation task is shown in Fig. 1.4.

While there exist only few works under source-free domain adaptation setting considering the category shift between domains, most of these methods [58, 59] rely on unknown samples generation and the whole pipeline is rather complex. *In this thesis, we aim to investigate source-free open-partial domain adaptation.*

1.2 Objectives and approach

In this thesis, we investigate multiple scenarios under the source-free domain adaptation setting, including the vanilla source-free domain adaptation, generalized/continual source-free domain adaptation and source-free open-partial domain adaptation. Here, we briefly define our objectives and approaches to solve the problems mentioned in the previous subsection.

1.2.1 Source-free Domain Adaptation

Most domain adaptation methods need access to labeled source data during the whole adaptation stage. This is often infeasible in real-world applications, where there may exist data privacy or property issues towards source data. Therefore, we investigate source-free domain adaptation, where a source pretrained model is adapted to the target domain without access to the source data. Unlike the existing methods which demands feature generation or complex pipelines, we therefore define the following objective:

> **Neighborhood Clustering for Source-free Domain Adaptation:** Propose clustering methods for source-free domain adaptation, which turn the challenging source-free adaptation problem to an unsupervised clustering task. By exploiting the intrinsic neighborhood structure, we aim to improve source-free domain adaptation.

To address the challenges of source-free domain adaptation, we propose two nearest neighborhood clustering based methods, that encourage local smoothness and overall diversity in the output space. In a first method dubbed as *Neighborhood Reciprocal Clustering*, we explore direct neighborhood clustering for source-free domain adaptation, which utilizes 2-hop neighbors, and consider the mutual neighborhood relation. In a second method, dubbed *Attracting-and-Dispersing*, we formalize the neighborhood clustering for source-free domain adaptation in the form of log-likelihood optimizing and optimize the upper bound of the loss function, which can be explained as discriminability and diversity. This also allows us to relate several existing methods in domain adaptation, unsupervised clustering, and contrastive learning via the perspective of discriminability and diversity.

1.2.2 Generalized Source-free Domain Adaptation

In many practical situations, models should perform well on both the target and source domain. For example, in real world application, it is ideal that we can use only one model in multiple different environments, in other words, the single model will be continually adapted to multiple target domains. The current source-free domain adaptation methods only aim to improve the target performance by fine-tuning the source model, which will result in forgetting on old domains. To address the forgetting issue on the old domains, we therefore define the following objective:

> **Generalized and Continual Source-free Domain Adaptation:** We aim to efficiently adapt the source-pretrained model to one or a sequence of unlabeled target domains under source-free domain adaptation setting. The adapted model should have good performance on both source and target domains.

First to achieve model adaptation in the source-free setting, we propose a simple clustering-based method, called Local Structure Clustering (LSC), where we encourage local smoothness in the prediction space judging by the corresponding feature similarity in the local neighborhood. Then to avoid forgetting towards the source domain after adaptation, we propose to use an attention-based regularization, called sparse domain attention (SDA). It will be deployed as a binary mask to the features, and during adaptation, the source domain attention mask will be utilized to regularize the model updating to avoid potential forgetting on the source domain. Furthermore, it can be easily extended to continual source-free domain adaptation.

1.2.3 Source-free Domain Adaptation in the Open-World

The source data for model pretraining only contains a limited set of categories, and in many real-world scenarios previously unseen categories can appear in the test data.

Most existing source-free domain adaptation methods cannot address the open-set setting where the label spaces of source and target domains are not identical. And the current methods, considering category shift in source-free domain adaptation, demand feature generation and complex pipelines. We therefore pursue the following objective:

> **Source-free Open-partial Domain Adaptation:** Under the source-free domain adaptation settings, the target domain may have unseen categories and also some source classes will no longer exist in the target domain. The model should be able to reject all unseen categories while correctly recognize seen classes.

To deal with the potential unseen categories in the target domain, we introduce an additional category dimension in the classifier head, it corresponds to the unknown (or novel) categories. During the source pretraining stage, the model will be trained with only seen categories, and the classifier is expected to output the maximal prediction score for the ground-truth class, and the second maximal score will be assigned to the unknown category. In this way, the model possesses strong open-set recognition ability without training with data from unseen categories. The model could be simply adapted to target domains, where there are novel classes, by weighted entropy minimization. It could be further improved to be combined with existing closed-set source-free domain adaptation methods.

2 Exploiting the Intrinsic Neighborhood Structure for Source-free Domain Adaptation[*]

2.1 Introduction

Most deep learning methods rely on training on large amount of labeled data, while they cannot generalize well to a related yet different domain. One research direction to address this issue is Domain Adaptation (DA), which aims to transfer learned knowledge from a source to a target domain. Most existing DA methods demand labeled source data during the adaptation period, however, it is often not practical that source data are always accessible, such as when applied on data with privacy or property restrictions. Therefore, recently, there have emerged a few works [58, 59, 69, 73] tackling a new challenging DA scenario where instead of source data only the source pretrained model is available for adapting, *i.e.*, source-free domain adaptation (SFDA). Among these methods, USFDA [58] addresses universal DA [161] and SF [59] addresses open-set DA [116]. In both universal and open-set DA the label set is different for source and target domains. SHOT [73] and 3C-GAN [69] are for closed-set DA where source and target domains have the same categories. 3C-GAN [69] is based on target-style image generation with a conditional GAN, and SHOT [73] is based on mutual information maximization and pseudo labeling. Finally, BAIT [156] extends MCD [115] to the SFDA setting. However, these methods ignore the intrinsic neighborhood structure of the target data in feature space which can be very valuable to tackle SFDA.

In this chapter, we focus on closed-set source-free domain adaptation. Our main observation is that current DA methods do not exploit the intrinsic neighborhood structure of the target data. We use this term to refer to the fact that, even though the target data might have shifted in the feature space (due to the covariance shift), target data of the same class is still expected to form a cluster in the embedding space. This can be implied to some degree from the t-SNE visualization of target features on the source model which suggests that significant cluster structure is preserved (see Fig. 2.1 (a)). This assumption is implicitly adopted by most DA methods, as instantiated by a recent DA work [132]. A well-established way to assess the structure of points in high-dimensional spaces is by considering the nearest neighbors of points,

[*] This chapter is based on a publication in the Advances in Neural Information Processing Systems (NeurIPS) 2021 [155]

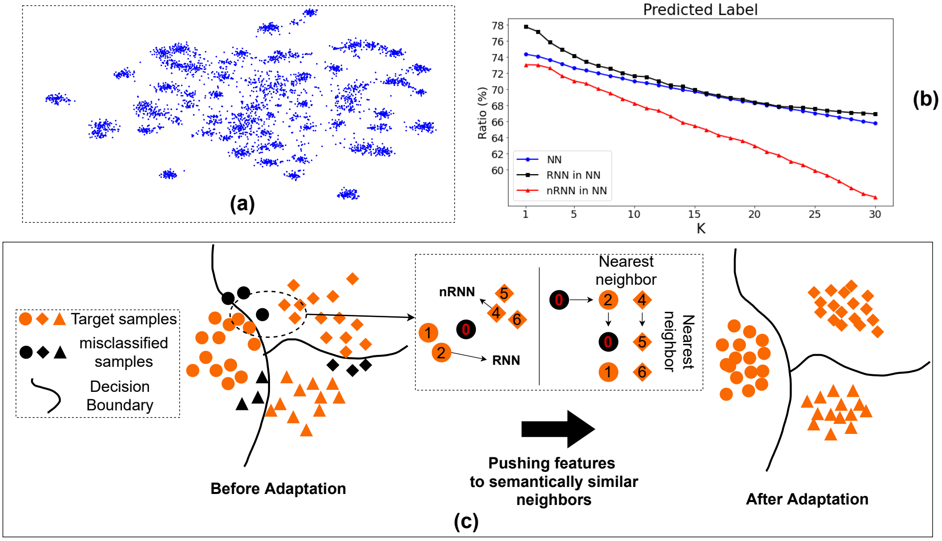

Figure 2.1: (**a**) t-SNE visualization of target features by source model. (**b**) Ratio of different type of nearest neighbor features of which: the *predicted* label is the same as the feature, K is the number of nearest neighbors. The features in (a) and (b) are on task Ar→Rw of Office-Home. (**c**) Illustration of our method. In the left shows we distinguish reciprocal and non-reciprocal neighbors. The adaptation is achieved by pushed the features towards reciprocal neighbors heavily.

which are expected to belong to the same class. However, this assumption is not true for all points; the blue curve in Figure 1(b) shows that around 75% of the nearest neighbors has the correct label. In this chapter, we observe that this problem can be mitigated by considering reciprocal nearest neighbors (RNN); the reciprocal neighbors of a point have the point as their neighbor. Reciprocal neighbors have been studied before in different contexts [50, 106, 168]. The reason why reciprocal neighbors are more trustworthy is illustrated in Fig. 2.1(c). Fig. 2.1(b) shows the ratio of neighbors which have the *correct prediction* for different kinds of nearest neighbors. The curves show that reciprocal neighbors indeed have more chances to predict the *true* label than non-reciprocal nearest neighbors (nRNN).

The above observation and analysis motivate us to assign different weights to the supervision from nearest neighbors. Our method, called Neighborhood Reciprocity Clustering (*NRC*), achieves source-free domain adaptation by encouraging reciprocal neighbors to concord in their label prediction. In addition, we will also consider a weaker connection to the non-reciprocal neighbors. We define affinity values to describe the degree of connectivity between each data point and its neighbors, which is also utilized to encourage class-consistency between neighbors, and we propose to use a self-regularization to decrease the negative impact of potential noisy neighbors. Furthermore, inspired by recent graph based methods [4, 173] which show that the higher order neighbors can provide relevant context, and also considering neighbors of neighbors is more likely to provide datapoints that are close on the data manifold [133]. Thus, to aggregate wider local information, we further retrieve the expanded neighbors, *i.e*, neighbor of the nearest neighbors, for auxiliary supervision.

Our contributions can be summarized as follows, to achieve source-free domain adaptation: (i) we explicitly exploit the fact that same-class data forms cluster in the target embedding space, we do this by considering the predictions of neighbors and reciprocal neighbors, (ii) we further show that considering an extended neighborhood of data points further improves results (iii) the experiments results on three 2D image datasets and one 3D point cloud dataset show that our method achieves state-of-the-art performance compared with related methods.

2.2 Related Work

Domain Adaptation. Most DA methods tackle domain shift by aligning the feature distributions. Early DA methods such as [83, 130, 136] adopt moment matching to align feature distributions. And in recent years, plenty of works have emerged that achieve alignment by adversarial training. DANN [31] formulates domain adaptation as an adversarial two-player game. The adversarial training of CDAN [84] is conditioned on several sources of information. DIRT-T [124] performs domain adversarial training with an added term that penalizes violations of the cluster assumption. Additionally, [88, 115] adopts prediction diversity between multiple learnable classifiers to achieve local or category-level feature alignment between source and target domains. AFN [151] shows that the erratic discrimination of target features stems from much smaller norms than those found in the source features. SRDC [132] proposes to directly uncover the intrinsic target discrimination via discriminative clustering to achieve adaptation. More related, [99] resorts to K-means clustering for open-set adaptation while considering global structure. Our method instead only focuses on nearest neighbors (local structure) for source-free adaptation.

Source-free Domain Adaptation. Source-present methods need supervision from the source domain during adaptation. Recently, there are several methods investigating source-free domain adaptation. USFDA [58] and FS [59] explore source-free universal DA [161] and open-set DA [116], and they propose to synthesize extra training samples to make the decision boundary compact, thereby allowing to recognise the open classes. For closed-set DA setting. SHOT [73] proposes to fix the source classifier and match the target features to the fixed classifier by maximizing mutual information and a proposed pseudo label strategy which considers global structure. 3C-GAN [69] synthesizes labeled target-style training images based on the conditional GAN to provide supervision for adaptation. Finally, SFDA [81] is for segmentation based on synthesizing fake source samples.

Graph Clustering. Our method shares some similarities with graph clustering work such as [118, 153, 154] by utilizing neighborhood information. However, our methods are fundamentally different. Unlike those works which require labeled data to train the graph network for estimating the affinity, we instead adopt reciprocity to assign affinity.

2.3 Method

Notation. We denote the labeled source domain data with n_s samples as $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, where the y_i^s is the corresponding label of x_i^s, and the unlabeled target domain data with n_t samples as $\mathcal{D}_t = \{x_j^t\}_{j=1}^{n_t}$. Both domains have the same C classes (closed-set setting). Under the SFDA setting $\mathcal{D}_s$ is only available for model pretraining. Our method is based on a neural network, which we split into two parts: a feature extractor f, and a classifier g. The feature output by the feature extractor is denoted as $\boldsymbol{z}(x) = f(x)$, the output of network is denoted as $p(x) = \delta(g(z)) \in \mathcal{R}^C$ where δ is the softmax function, for readability we will abandon the input and use $\boldsymbol{z}, p$ in the following sections.

Overview. We assume that the source pretrained model has already been trained. As discusses in the introduction, the target features output by the source model form clusters. We exploit this intrinsic structure of the target data for SFDA by considering the neighborhood information, and the adaptation is achieved with the following objective:

$$\mathcal{L} = -\frac{1}{n_t} \sum_{x_i \in \mathcal{D}_t} \sum_{x_j \in \text{Neigh}(x_i)} \frac{D_{sim}(p_i, p_j)}{D_{dis}(x_i, x_j)} \tag{2.1}$$

where the $\text{Neigh}(x_i)$ means the nearest neighbors of x_i, D_{sim} computes the similarity between predictions, and D_{dis} is a constant measuring the semantic distance (dissimilarity) between data. The principle behind the objective is to push the data towards their semantically close neighbors by encouraging similar predictions. In the next sections, we will define D_{sim} and D_{dis}.

2.3.1 Encouraging Class-Consistency with Neighborhood Affinity

To achieve adaptation without source data, we use the prediction of the nearest neighbor to encourage prediction consistency. While the target features from the source model are not necessarily totally intrinsic discriminative, meaning some neighbors belong to different class and will provide the wrong supervision. To decrease the potentially negative impact of those neighbors, we propose to weigh the supervision from neighbors according to the connectivity (semantic similarity). We define *affinity* values to signify the connectivity between the neighbor and the feature, which corresponds to the $\frac{1}{D_{dis}}$ in Eq. 2.1 indicating the semantic similarity.

To retrieve the nearest neighbors for batch training, similar to [111, 148, 175], we build two memory banks: $\mathscr{F}$ stores all target features, and $\mathscr{S}$ stores corresponding prediction scores:

$$\mathscr{F} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{n_t}] \text{and } \mathscr{S} = [p_1, p_2, \ldots, p_{n_t}] \tag{2.2}$$

We use the cosine similarity for nearest neighbors retrieving. The difference between ours and [111, 148] lies in the fact that we utilize the memory bank to retrieve nearest neighbors while [111, 148] adopts the memory bank to compute the instance discrimination loss. Before every mini-batch training, we simply update the old items in the memory banks corresponding to current mini-batch. Note that updating the memory bank is only done to replace the old low-dimension vectors with new ones computed by the model, and does not require any additional computation.

We then use the prediction of the neighbors to supervise the training weighted by the affinity values, with the following objective adapted from Eq. 2.1:

$$\mathscr{L}_{\mathscr{N}} = -\frac{1}{n_t} \sum_i \sum_{k \in \mathscr{N}_K^i} A_{ik} \mathscr{S}_k^\top p_i \tag{2.3}$$

where we use the dot product to compute the similarity between predictions, corresponding to D_{sim} in Eq.2.1, the k is the index of the k-th nearest neighbors of $\boldsymbol{z}_i$, $\mathscr{S}_k$ is the k-th item in memory bank $\mathscr{S}$, A_{ik} is the affinity value of k-th nearest neighbors of feature $\boldsymbol{z}_i$. Here the $\mathscr{N}_K^i$ is the index set[†] of the K-nearest neighbors of feature $\boldsymbol{z}_i$. Note that all neighbors are retrieved from the feature bank $\mathscr{F}$. With the affinity value as weight, this objective pushes the features to their neighbors with strong connectivity and to a lesser degree to those with weak connectivity.

To assign larger affinity values to semantic similar neighbors, we divide the nearest

[†] All indexes are in the same order for the dataset and memory banks.

neighbors retrieved into two groups: reciprocal nearest neighbors (RNN) and non-reciprocal nearest neighbors (nRNN). The feature $\boldsymbol{z}_j$ is regarded as the RNN of the feature $\boldsymbol{z}_i$ if it meets the following condition:

$$j \in \mathcal{N}_K^i \wedge i \in \mathcal{N}_M^j \tag{2.4}$$

Other neighbors which do not meet the above condition are nRNN. Note that the normal definition of reciprocal nearest neighbors [106] applies $K = M$, while in this chapter K and M can be different. We find that reciprocal neighbors have a higher potential to belong to the same cluster as the feature (Fig. 2.1(b)). Thus, we assign a high affinity value to the RNN features. Specifically for feature $\boldsymbol{z}_i$, the affinity value of its j-th K-nearest neighbor is defined as:

$$A_{i,j} = \begin{cases} 1 & \text{if } j \in \mathcal{N}_K^i \wedge i \in \mathcal{N}_M^j \\ r & \text{otherwise.} \end{cases} \tag{2.5}$$

where r is a hyperparameter. If not specified r is set to 0.1.

To further reduce the potential impact of noisy neighbors in $\mathcal{N}_K$, which belong to the different class but still are RNN, we propose a simply yet effective way dubbed *self-regularization*, that is, to not ignore the current prediction of ego feature:

$$\mathcal{L}_{self} = -\frac{1}{n_t} \sum_{i}^{n_t} \mathcal{S}_i^\top p_i \tag{2.6}$$

where $\mathcal{S}_i$ means the stored prediction in the memory bank, note this term is a *constant vector* and is identical to the p_i since we update the memory banks before the training, here the loss is only back-propagated for variable p_i.

To avoid the degenerated solution [34, 122] where the model predicts all data as some specific classes (and does not predict other classes for any of the target data), we encourage the prediction to be balanced. We adopt the prediction diversity loss which is widely used in clustering [34, 37, 49] and also in several domain adaptation works [73, 122, 132]:

$$\mathcal{L}_{div} = \sum_{c=1}^{C} \mathrm{KL}(\bar{p}_c || q_c), \text{with } \bar{p}_c = \frac{1}{n_t} \sum_i p_i^{(c)} \text{ ,and } q_{\{c=1,..,C\}} = \frac{1}{C} \tag{2.7}$$

where the $p_i^{(c)}$ is the score of the c-th class and $\bar{p}_c$ is the empirical label distribution, it represents the predicted possibility of class c and q is a uniform distribution.

Algorithm 1 Neighborhood Reciprocity Clustering for Source-free Domain Adaptation

```
Require: $\mathcal{D}_s$ (only for source model training), $\mathcal{D}_t$
Pre-train model on $\mathcal{D}_s$
Build feature bank $\mathcal{F}$ and score bank $\mathcal{S}$ for $\mathcal{D}_t$
while Adaptation do
    Sample batch $\mathcal{T}$ from $\mathcal{D}_t$
    Update $\mathcal{F}$ and $\mathcal{S}$ corresponding to current batch $\mathcal{T}$
    Retrieve nearest neighbors $\mathcal{N}$ for each of $\mathcal{T}$
    Compute affinity value $A$                                  ▷ Eq.2.5
    Retrieve expanded neighborhoods $E$ for each of $\mathcal{N}$
    Compute loss and update the model                           ▷ Eq. 2.9
end while
```

2.3.2 Expanded Neighborhood Affinity

As mentioned in Sec. 2.1, a simple way to achieve the aggregation of more information is by considering more nearest neighbors. However, a drawback is that larger neighborhoods are expected to contain more datapoint from multiple classes, defying the purpose of class consistency. A better way to include more target features is by considering the M-nearest neighbor of each neighbor in $\mathcal{N}_K$ of $\boldsymbol{z}_i$ in Eq. 2.4, *i.e.*, the expanded neighbors. These target features are expected to be closer on the target data manifold than the features that are included by considering a larger number of nearest neighbors [133]. The expanded neighbors of feature $\boldsymbol{z}_i$ are defined as $E_M(\boldsymbol{z}_i) = \mathcal{N}_M(\boldsymbol{z}_j)\ \forall j \in \mathcal{N}_K(\boldsymbol{z}_i)$, *note that* $E_M(\boldsymbol{z}_i)$ *is still an index set and* i *(ego feature)* $\notin E_M(\boldsymbol{z}_i)$. We directly assign a small affinity value r to those expanded neighbors, since they are further than nearest neighbors and may contain noise. We utilize the prediction of those expanded neighborhoods for training:

$$\mathcal{L}_E = -\frac{1}{n_t}\sum_i \sum_{k\in\mathcal{N}_K^i} \sum_{m\in E_M^k} r\mathcal{S}_m^\top p_i \tag{2.8}$$

where E_M^k contain the M-nearest neighbors of neighbor k in $\mathcal{N}_K$.

Although the affinity values of all expanded neighbors are the same, it does not necessarily mean that they have equal importance. Taking a closer look at the expanded neighbors $E_M(\boldsymbol{z}_i)$, some neighbors will show up more than once, for example $\boldsymbol{z}_m$ can be the nearest neighbor of both $\boldsymbol{z}_h$ and $\boldsymbol{z}_j$ where $h, j \in \mathcal{N}_K(\boldsymbol{z}_i)$, and the nearest neighbors can also serve as expanded neighbor. It implies that those neighbors form compact cluster, and we posit that those duplicated expanded neighbors have potential to be semantically closer to the ego-feature $\boldsymbol{z}_i$. Thus, we do not remove duplicated features in $E_M(\boldsymbol{z}_i)$, as those can lead to actually larger affinity value for those expanded neighbors. This is one advantage of utilizing expanded neighbors instead of more nearest neighbors, we will verify the importance of maintaining the duplicated features in the experimental section.

Final objective. Our method, called *Neighborhood Reciprocity Clustering* (*NRC*), is illustrated in Algorithm. 1. The final objective for adaptation is:

$$\mathcal{L} = \mathcal{L}_{div} + \mathcal{L}_{\mathcal{N}} + \mathcal{L}_E + \mathcal{L}_{self}. \tag{2.9}$$

Table 2.1: Accuracies (%) on Office-31 for ResNet50-based methods.

Method	SF	A→D	A→W	D→W	W→D	D→A	W→A	Avg
DAN [83]	✗	78.6	80.5	97.1	99.6	63.6	62.8	80.4
DANN [31]	✗	79.7	82.0	96.9	99.1	68.2	67.4	82.2
ADDA [135]	✗	77.8	86.2	96.2	98.4	69.5	68.9	82.9
MCD [115]	✗	92.2	88.6	98.5	**100.0**	69.5	69.7	86.5
CDAN [84]	✗	92.9	94.1	98.6	**100.0**	71.0	69.3	87.7
MDD [165]	✗	90.4	90.4	98.7	99.9	75.0	73.7	88.0
BNM [21]	✗	90.3	91.5	98.5	**100.0**	70.9	71.6	87.1
DMRL [147]	✗	93.4	90.8	99.0	**100.0**	73.0	71.2	87.9
BDG [152]	✗	93.6	93.6	99.0	**100.0**	73.2	72.0	88.5
MCC [53]	✗	95.6	95.4	98.6	100.0	72.6	73.9	89.4
SRDC [132]	✗	95.8	95.7	99.2	100.0	76.7	77.1	90.8
RWOT [150]	✗	94.5	95.1	**99.5**	100.0	**77.5**	77.9	90.8
RSDA-MSTN [39]	✗	95.8	**96.1**	99.3	**100.0**	77.4	**78.9**	**91.1**
USFDA [58]	-	-	-	-	-	-	85.4	
SHOT [73]	✓	94.0	90.1	98.4	99.9	74.7	74.3	88.6
3C-GAN [69]	✓	92.7	93.7	98.5	99.8	75.3	77.8	89.6
NRC	✓	**96.0**	90.8	99.0	**100.0**	75.3	75.0	89.4

2.4 Experiments

Datasets. We use three 2D image benchmark datasets and a 3D point cloud recognition dataset. **Office-31** [110] contains 3 domains (Amazon, Webcam, DSLR) with

Table 2.2: Accuracies (%) on Office-Home for ResNet50-based methods.

Method	SF	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	**Avg**
DAN [83]	✗	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN [31]	✗	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
MCD [115]	✗	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
CDAN [84]	✗	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
SAFN [151]	✗	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
Symnets [164]	✗	47.7	72.9	78.5	64.2	71.3	74.2	64.2	48.8	79.5	74.5	52.6	82.7	67.6
MDD [165]	✗	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	**60.2**	82.3	68.1
TADA [144]	✗	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
MDD+IA [52]	✗	56.0	77.9	79.2	64.4	73.1	74.4	64.2	54.2	79.9	71.2	58.1	83.1	69.5
BNM [21]	✗	52.3	73.9	80.0	63.3	72.9	74.9	61.7	49.5	79.7	70.5	53.6	82.2	67.9
BDG [152]	✗	51.5	73.4	78.7	65.3	71.5	73.7	65.1	49.7	81.1	74.6	55.1	84.8	68.7
SRDC [132]	✗	52.3	76.3	81.0	**69.5**	76.2	78.0	**68.7**	53.8	81.7	**76.3**	57.1	85.0	71.3
RSDA-MSTN [39]	✗	53.2	77.7	81.3	66.4	74.0	76.5	67.9	53.0	82.0	75.8	57.8	85.4	70.9
SHOT [73]	✓	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
NRC	✓	**57.7**	**80.3**	**82.0**	68.1	**79.8**	**78.6**	65.3	**56.4**	**83.0**	71.0	58.6	**85.6**	**72.2**

31 classes and 4,652 images. **Office-Home** [141] contains 4 domains (Real, Clipart, Art, Product) with 65 classes and a total of 15,500 images. **VisDA** [102] is a more challenging dataset, with 12-class synthetic-to-real object recognition tasks, its source domain contains of 152k synthetic images while the target domain has 55k real object images. **PointDA-10** [105] is the first 3D point cloud benchmark specifically designed for domain adaptation, it has 3 domains with 10 classes, denoted as ModelNet-10, ShapeNet-10 and ScanNet-10, containing approximately 27.7k training and 5.1k testing images together.

Evaluation. We compare with existing source-present and source-free DA methods. *All results are the average on three random runs.* **SF** in the tables denotes source-free.

Model details. For fair comparison with related methods, we also adopt the backbone of ResNet-50 [41] for Office-Home and ResNet-101 for VisDA, and PointNet [103] for PointDA-10. Specifically, for 2D image datasets, we use the same network architecture as SHOT [73], *i.e.*, the final part of the network is: fully connected layer − Batch Normalization [48] − fully connected layer with weight normalization [117]. And for PointDA-10 [103], we use the code released by the authors for fair comparison with PointDAN [103], and only use the backbone without any of their proposed modules. To train the source model, we also adopt label smoothing as SHOT does. We adopt SGD with momentum 0.9 and batch size of 64 for all 2D datasets, and Adam for PointDA-10. The learning rate for Office-31 and Office-Home is set to 1e-3 for all layers, except for the last two newly added fc layers, where we apply 1e-2. Learning rates are set 10 times smaller for VisDA. Learning rate for PointDA-10 is set to 1e-6. We train 30 epochs for Office-31 and Office-Home while 15 epochs for VisDA, and 100 for PointDA-10. For the number of nearest neighbors (K) and expanded neighborhoods (M), we use 3,2 for Office-31, Office-Home and PointDA-10, since VisDA is much larger we set K, M to 5. Experiments are conducted on a TITAN Xp.

2.4.1 Results

2D image datasets. We first evaluate the target performance of our method compared with existing DA and SFDA methods on three 2D image datasets. As shown in Table 2.1-2.3, the top part shows results for the source-present methods *with access to source data during adaptation*. The bottom shows results for the source-free DA methods. On Office-31, our method gets similar results compared with source-free method 3C-GAN and lower than source-present method RSDA-MSTN. And our method achieves state-of-the-art performance on Office-Home and VisDA, especially

Table 2.3: Accuracies (%) on VisDA-C (Synthesis → Real) for ResNet101-based methods.

Method	SF	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
DANN [31]	✗	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
DAN [83]	✗	87.1	63.0	76.5	42.0	90.3	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
ADR [114]	✗	94.2	48.5	84.0	72.9	90.1	74.2	92.6	72.5	80.8	61.8	82.2	28.8	73.5
CDAN [84]	✗	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
CDAN+BSP [17]	✗	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
SAFN [151]	✗	93.6	61.3	84.1	70.6	94.1	79.0	91.8	79.6	89.9	55.6	89.0	24.4	76.1
SWD [62]	✗	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
MDD [165]	✗	-	-	-	-	-	-	-	-	-	-	-	-	74.6
DMRL [147]	✗	-	-	-	-	-	-	-	-	-	-	-	-	75.5
MCC [53]	✗	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
STAR [88]	✗	95.0	84.0	**84.6**	73.0	91.6	91.8	85.9	78.4	94.4	84.7	87.0	42.2	82.7
RWOT [150]	✗	95.1	80.3	83.7	**90.0**	92.4	68.0	**92.5**	82.2	87.9	78.4	**90.4**	**68.2**	84.0
3C-GAN [69]	✓	94.8	73.4	68.8	74.8	93.1	95.4	88.6	**84.7**	89.1	84.7	83.5	48.1	81.6
SHOT [73]	✓	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
NRC	✓	**96.8**	**91.3**	82.4	62.4	**96.2**	**95.9**	86.1	80.6	**94.8**	**94.1**	90.4	59.7	**85.9**

Table 2.4: Accuracies (%) on PointDA-10. *The results except ours are from PointDAN [105].* M: Model, Sh: Shape, Sc: Scan.

	SF	M→Sh	M→Sc	Sh→M	Sh→Sc	Sc→M	Sc→Sh	Avg
MMD [85]	✗	57.5	27.9	40.7	26.7	47.3	54.8	42.5
DANN [30]	✗	58.7	29.4	42.3	30.5	48.1	56.7	44.2
ADDA [135]	✗	61.0	30.5	40.4	29.3	48.9	51.1	43.5
MCD [115]	✗	62.0	31.0	41.4	31.3	46.8	59.3	45.3
PointDAN [105]	✗	64.2	**33.0**	47.6	**33.9**	49.1	64.1	48.7
Source-only		43.1	17.3	40.0	15.0	33.9	47.1	32.7
NRC	✓	**64.8**	25.8	**59.8**	26.9	**70.1**	**68.1**	**52.6**

on VisDA our method surpasses the source-free method SHOT and source-present method RWOT by a wide margin (3% and 1.9% respectively). The reported results clearly demonstrate the efficiency of the proposed method for source-free domain adaptation. Interestingly, like already observed in the SHOT paper, source-free methods outperform methods that have access to source data during adaptation.

3D point cloud dataset. We also report the result for the PointDA-10. As shown in Table 2.4, our method outperforms PointDA [105], which demands source data for adaptation and is specifically tailored for point cloud data with extra attention modules, by a large margin (4%).

Table 2.5: Ablation study of different modules on Office-Home (**left**) and VisDA (**middle**), comparison between using expanded neighbors and larger nearest neighbors (**right**).

$\mathscr{L}_{div}$	$\mathscr{L}_{\mathscr{N}}$	$\mathscr{L}_{E}$	$\mathscr{L}_{\hat{E}}$	A	Avg
					59.5
✓					62.1
✓	✓				69.1
✓	✓			✓	71.1
✓	✓	✓			65.2
✓	✓	✓		✓	**72.2**
✓	✓		✓	✓	69.1

$\mathscr{L}_{div}$	$\mathscr{L}_{\mathscr{N}}$	$\mathscr{L}_{E}$	$\mathscr{L}_{\hat{E}}$	A	Acc
					44.6
✓					47.8
✓	✓				81.5
✓	✓			✓	82.7
✓	✓	✓			61.2
✓	✓	✓		✓	**85.9**
✓	✓		✓	✓	82.0

Method&Dataset	Acc
VisDA (K=M=5)	**85.9**
VisDA w/o E (K=30)	84.0
OH (K=3,M=2)	**72.2**
OH w/o E (K=9)	69.5

Table 2.6: Runtime analysis on SHOT and our method. For SHOT, pseudo labels are computed at each epoch. 20%, 10% and 5% denote the percentage of target features which are stored in the memory bank.

VisDA	**Runtime** (s/epoch)	**Per-class** (%)
SHOT	618.82	82.9
NRC	540.89	85.9
NRC(20% for memory bank)	507.15	85.3
NRC(10% for memory bank)	499.49	85.2
NRC(5% for memory bank)	499.28	85.1

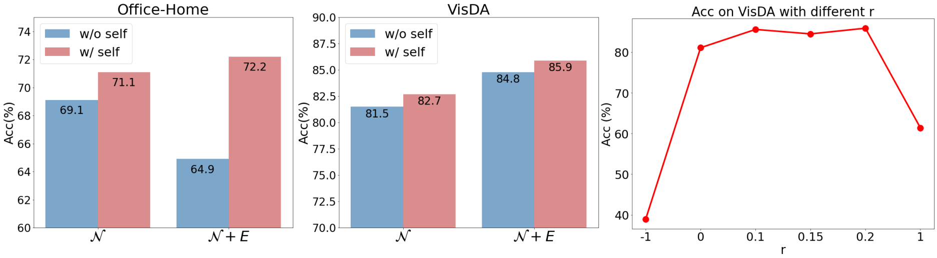

Figure 2.2: (**Left and middle**) Ablation study of $\mathcal{L}_{self}$ on Office-Home and VisDA respectively. (**Right**) Performance with different r on VisDA.

2.4.2 Analysis

Ablation study on neighbors $\mathcal{N}$, E and affinity A. In the first two tables of Table 2.5, we conduct the ablation study on Office-Home and VisDA. The 1-st row contains results from the source model and the 2-nd row from only training with the diversity loss $\mathcal{L}_{div}$. From the remaining rows, several conclusions can be drawn.

First, the original supervision, which considers all neighbors equally can lead to a decent performance (69.1 on Office-Home). Second, considering higher affinity values for reciprocal neighbors leads to a large performance gain (71.1 on Office-Home). Last but not the least, the expanded neighborhoods can also be helpful, but only when combined with the affinity values A (72.2 on Office-Home). Using expanded neighborhoods without affinity obtains bad performance (65,2 on Office-Home). We conjecture that those expanded neighborhoods, especially those neighbors of nRNN, may be noisy as discussed in Sec. 2.3.2. Removing the affinity A means we treat all those neighbors equally, which is not reasonable.

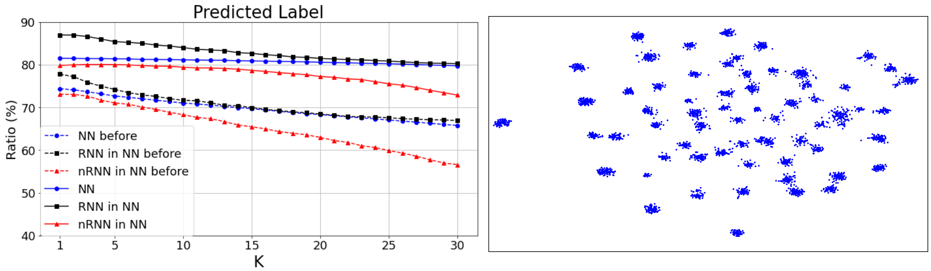

Figure 2.3: (**Left**) Ratio of different type of nearest neighbor features which have the correct predicted label, before and after adaptation. (**Right**) Visualization of target features after adaptation.

We also show that duplication in the expanded neighbors is important in the last row of Table 2.5, where the $\mathscr{L}_{\hat{E}}$ means we remove duplication in Eq. 2.8. The results show that the performance will degrade significantly when removing them, implying that the duplicated expanded neighbors are indeed more important than others.

Next we ablate the importance of the expanded neighborhood in the right of Table2.5. We show that if we increase the number of datapoints considered for class-consistency by simply considering a larger K, we obtain significantly lower scores. We have chosen K so that the total number of points considered is equal to our method (i.e. 5+5*5=30 and 3+3*2=9). Considering neighbors of neighbors is more likely to provide datapoints that are close on the data manifold [133], and are therefore more likely to share the class label with the ego feature.

Runtime analysis. Instead of storing all feature vectors in the memory bank, we follow the same memory bank setting as in [25] which is for nearest neighbor retrieval. The method only stores a fixed number of target features, we update the memory bank at the end of each iteration by taking the n (batch size) embeddings from the current training iteration and concatenating them at the end of the memory bank, and discard the oldest n elements from the memory bank. We report the results with this type of memory bank of different buffer size in the Table 2.6. The results show that indeed this could be an efficient way to reduce computation on very large datasets.

Ablation study on self-regularization. In the left and middle of Fig 2.2, we show the results with and without self-regularization $\mathscr{L}_{self}$. The $\mathscr{L}_{self}$ can improve the performance when adopting only nearest neighbors $\mathscr{N}$ or all neighbors $\mathscr{N}+E$. The results imply that self-regularization can effectively reduce the negative impact of the potential noisy neighbors, especially on the Office-Home dataset.

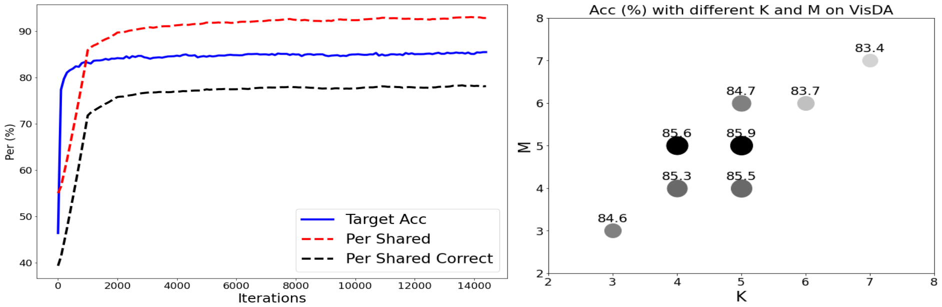

Figure 2.4: (**Left**) The three curves are (on VisDA): target accuracy (*Blue*), ratio of features which have 5-nearest neighbors all sharing the same predicted label (*dashed Red*), and ratio of features which have 5-nearest neighbors all sharing the same and *correct* predicted label (*dashed Black*). (**Right**) Ablation study on choice of K and M on VisDA.

Sensitivity to hyperparameter. There are three hyperparameters in our method: K and M which are the number of nearest neighbors and expanded neighbors, r which is the affinity value assigned to nRNN. We show the results with different r in the right of Fig. 2.2. *Note we keep the affinity of expanded neighbors as 0.1.* $r = 1$ means no affinity. $r = -1$ means treating supervision of nRNN feature as totally wrong, which is not always the case and will lead to quite lower result. $r = 0$ can also achieve good performance, signifying RNN can already work well. Results with $r = 0.1/0.15/0.2$ show that our method is not sensitive to the choice of a reasonable r. Note in DA, there is no validation set for hyperparameter tuning, we show the results varying the number of neighbors in the right of Fig. 2.4, demonstrating the robustness to the choice of K and M.

Training curve. We show the evolution of several statistics during adaptation on VisDA in the left of Fig. 2.4. The blue curve is the target accuracy. The dashed red and black curves are the ratio of features which have 5-nearest neighbors all sharing the same (*dashed Red*), or the same and also **correct** (*dashed Black*) predicted label. The curves show that the target features are clustering during the training. Another interesting finding is that the curve 'Per Shared' correlates with the accuracy curve, which might therefore be used to determine training convergence.

Accuracy of supervision from neighbors. We also show the accuracy of supervision from neighbors on task Ar→Rw of Office-Home in Fig. 2.3(left). It shows that after adaptation, the ratio of all types of neighbors having more correct predicted label, proving the effectiveness of the method.

t-SNE visualization. We show the t-SNE feature visualization on task Ar→Rw of target features before (Fig. 2.1(a)) and after (Fig. 2.3(right)) adaptation. After adaptation, the features are more compactly clustered.

2.5 Conclusions

We introduce a source-free domain adaptation (SFDA) method by uncovering the intrinsic target data structure. We propose to achieve the adaptation by encouraging label consistency among local target features. We differentiate between nearest neighbors, reciprocal neighbors and expanded neighborhood. Experimental results verify the importance of considering the local structure of the target features. Finally, our experimental results on both 2D image and 3D point cloud datasets testify the efficacy of our method.

3 Attracting and Dispersing: A Simple Approach for Source-free Domain Adaptation[*]

3.1 Introduction

Supervised learning methods which are based on training with huge amounts of labeled data are advancing almost all fields of computer vision. However, the learned models typically perform decently on test data which have a similar distribution with the training set. Significant performance degradation will occur if directly applying those models to a new domain different from the training set, where the data distribution (such as variation of background, styles or camera parameter) is considerably different. This kind of distribution shift is formally denoted as domain/distribution shift. It limits the generalization of the model to unseen domains which is important in real-world applications. There are several research fields trying to tackle this problem. One of them is *Domain Adaptation* (DA), which aims to reduce the domain shift between the labeled source domain and unlabeled target domain. Typical works [38, 98] resort to learn domain-invariant features, thus improving generalization ability of the model between different domains. And in the past few years, the main research line of domain adaptation is either trying to minimize the distribution discrepancy between two domains [82, 83, 86], or deploying adversarial training on features to learn domain invariant representation [20, 88, 135, 164]. Some methods also tackle domain shift from the view of semi-supervised learning [74, 163] or clustering [21, 23, 132].

Many recent methods [45, 69, 76, 149, 155, 157] focus on *source-free domain adaptation* (SFDA), where source data are unavailable during target adaptation, due to data privacy and intellectual property concerns of both users and businesses. Some SFDA methods resort to neighborhood clustering and pseudo labeling. However, pseudo labeling methods [76] may suffer from negative impact from noisy labels, and neighborhood clustering methods [155, 157] fail to investigate the potential information from dissimilar samples. Other methods either demand complex extra modules/processing [69, 149] or the storing of historical models for contrastive learning [45].

Based on the fact that target features from the source model already form some semantic structure and following the intuition that for a target feature from a (source-

[*] This chapter is based on a publication in the Advances in Neural Information Processing Systems (NeurIPS) 2022 [158]

Table 3.1: Detailed comparison of SFDA methods on **VisDA**. 'ODA/PDA' means whether the method reports the results for open-set or partial-set DA. $|\mathscr{L}|$ means number of training objective terms.

Method	**Extra Modules/Processing**	**ODA/PDA**	$\lvert\mathscr{L}\rvert$	**Per-class**
SHOT [73]	Access all target data for pseudo labeling	✓	3	82.9
3C-GAN [69]	Data generation by conditional GAN	✗	5	81.6
A^2Net [149]	Self-supervised learning with extra classifiers	✗	5	84.3
G-SFDA [157]	Store features for nearest neighbor retrieval	✗	2	85.4
NRC [155]	Store features for 2-hop nearest neighbor retrieval	✗	4	85.9
HCL [45]	Store historical models	✓	2	83.5
AaD	Store features for nearest neighbor retrieval	✓	2	**88.0**

pretrained) model, similar features should have closer predictions than dissimilar ones, we propose a new objective dubbed as Attracting-and-Dispersing (**AaD**) to achieve it. we upperbound this objective, resulting in a simple final objective which only contains two types of terms, which encourage discriminability and diversity respectively. Further, we unify several popular domain adaptation, source-free domain adaptation and contrastive learning methods from the perspective of discriminability and diversity. Experimental results on several benchmarks prove the superiority of our proposed method. Our simple method improves the state-of-the-art on the challenging VisDA with 2.1% to 88.0%. Additionally, extra experiments on open-set and partial-set DA further prove the effectiveness of our method. A preliminary comparison between different SFDA method is shown in Tab. 3.1, which shows the simplicity and generalization ability of our method: it only requires the storing of features and a few nearest neighbors searches without any additional module like a generator [69] or a classifier [149].

We summary our contributions as follows:

- We propose to tackle source-free domain adaptation by optimizing an upperbound of the proposed clustering objective, which is surprisingly simple.
- We relate several popular existing methods in domain adaptation, source-free domain adaptation and contrastive learning via the perspective of discriminability and diversity, which is helpful to understand existing methods and beneficial for future improvement.
- The experimental results prove the efficacy of our method, especially we achieve new state-of-the-art on the challenging VisDA, and the method can be also extended

to source-free open-set and partial-set domain adaptation.

3.2 Related Work

Domain Adaptation. Early DA methods such as [83, 130, 136] adopt moment matching to align feature distributions. For adversarial learning methods, DANN [31] formulates domain adaptation as an adversarial two-player game. The adversarial training of CDAN [84] is conditioned on several sources of information. DIRT-T [124] performs domain adversarial training with an added term that penalizes violations of the cluster assumption. Additionally, [62, 88, 115] adopts prediction diversity between multiple learnable classifiers to achieve local or category-level feature alignment between source and target domains. SRDC [132] proposes to directly uncover the intrinsic target discrimination via discriminative clustering to achieve adaptation. CST [80] proposes a simple self-training strategy to improve the rough pseudo label under domain shift.

Source-free Domain Adaptation. The above-mentioned normal domain adaptation methods need to access source domain data at all time during adaptation. In recent years plenty of methods emerge trying to tackle source-free domain adaptation. USFDA [58] and FS [59] resort to synthesize extra training samples in order to get compact decision boundaries, which is beneficial for both the detection of open classes and also target adaptation. SHOT [73] proposes to freeze the source classifier and it clusters target features by maximizing mutual information along with pseudo labeling for extra supervision. 3C-GAN [69] synthesizes labeled target-style training images. It is based on a conditional GAN to provide supervision for adaptation. BAIT [156] extends MCD [115] to source-free setting. A^2Net [149] proposes to learn an additional target-specific classifier for hard samples and adopts a contrastive category-wise matching module to cluster target features. HCL [45] adopts Instance Discrimination [148] for features from current and historical models to cluster features, along with a generated pseudo label conditioned on historical consistency. G-SFDA [157] and NRC [155] propose neighborhood clustering which enforces prediction consistency between local neighbors.

Deep Clustering and Contrastive Learning. Recent Deep Clustering methods can be roughly divided into two groups, they the differ in how they learn the feature representation and cluster assignments, either simultaneously or alternatively. For example, DAC [13] and DCCM [146] alternately update cluster assignments and between-sample similarity. Simultaneous clustering methods IIC [51] and ISMAT [43] are based on mutual information maximizing between samples and theirs augmentations. LA [175] depends on a huge amount of nearest neighbor searches and multiple extra runs of *k-means* clustering to aggregate features. Recent unsupervised clustering works [70, 121, 134] start to rely on contrastive learning, where InfoNCE [96] is typically deployed. And recently NNCLR [25] proposes to use nearest neighbors in the latent space as positives in contrastive learning to cover more semantic variations than pre-defined transformations. However an inevitable problem of normal contrastive learning is class collision where negative samples are from the same class. To tackle this issue, recent works [46, 67] propose to estimate cluster prototypes and integrate them into contrastive learning.

3.3 Method

For source-free domain adaptation (SFDA), we are given source-pretrained model in the beginning and an unlabeled target domain with N_t samples as $\mathscr{D}_t = \{x_i^t\}_{i=1}^{N_t}$. Target domain have same C classes as source domain in this chapter (known as the closed-set setting). The goal of SFDA is to adapt the model to target domain without source data. We divide the model into two parts: the feature extractor f, and the classifier g. The output of the feature extractor is denoted as feature ($\boldsymbol{z_i} = f(x) \in \mathbb{R}^h$), where h is dimension of the feature space. The output of classifier is denoted as ($p_i = \delta(g(z_i)) \in \mathbb{R}^C$) where δ is the softmax function. We denote $P \in \mathbb{R}^{bs \times C}$ as the prediction matrix in a mini-batch. Regarding the SFDA as an unsupervised clustering problem, we address SFDA problem by clustering target features based on the proposed AaD. In additionally, we relate our method with several existing DA, SFDA and contrastive learning methods.

3.3.1 Attracting and Dispersing for Source-free Domain Adaptation

Since the source-pretrained model already learns a good feature representation, it can provides a decent initialization for target adaptation. We propose to achieve SFDA by attracting predictions for features that are located close in feature space, while dispersing predictions of those features farther away in feature space.

We define p_{ij} as the probability that the feature $z_i \in \mathbb{R}^h$ has similar (or the same) prediction to feature z_j: $p_{ij} = \frac{e^{p_i^T p_j}}{\sum_{k=1}^{N_t} e^{p_i^t p_k}}$. It can be interpreted as the possibility that p_j is selected as the neighbor of p_i in the output space [36].

We then define two sets for each feature z_i: close neighbor set $\mathscr{C}_i$ containing K-nearest neighbors of z_i (with distances as cosine similarity), and background set $\mathscr{B}_i$ which contains the features that are not in $\mathscr{C}_i$ (features potentially from different

Algorithm 2 Attracting and Dispersing for SFDA

```
Require: Source-pretrained model and target data $\mathscr{D}_t$
Build memory bank storing all target features and predictions
while Adaptation do
    Sample batch $\mathscr{T}$ from $\mathscr{D}_t$ and Update memory bank
    For each feature $z_i$ in $\mathscr{T}$, retrieve $K$-nearest neighbors ($\mathscr{C}_i$) and their predictions from memory bank
    Update model by minimizing Eq. 3.6
end while
```

classes). To retrieve nearest neighbors for training, we build two memory banks to store all *target* features along with their predictions just like former works [74, 111, 155, 157], which is efficient in both memory and computation, since only the features along with their predictions computed in each mini-batch are used to update the memory bank.

Intuitively, for each feature z_i, the features in $\mathscr{B}_i$ should have less similar predictions than those in $\mathscr{C}_i$[†]. To achieve this, we first define two likelihood functions:

$$P(\mathscr{C}_i|\theta) = \prod_{j\in\mathscr{C}_i} p_{ij} = \prod_{j\in\mathscr{C}_i} \frac{e^{p_i^T p_j}}{\sum_{k=1}^{N_t} e^{p_i^T p_k}}, \tag{3.1}$$

$$P(\mathscr{B}_i|\theta) = \prod_{j\in\mathscr{B}_i} p_{ij} = \prod_{j\in\mathscr{B}_i} \frac{e^{p_i^T p_j}}{\sum_{k=1}^{N_t} e^{p_i^T p_k}} \tag{3.2}$$

where θ denotes parameters of the model, for readability we omit θ in following equations. The probability p_j in Eq. 3.1 is the stored prediction for neighborhood feature z_j, which is retrieved from the memory bank.

We then propose to achieve target features clustering by minimizing the following negative log-likelihood, denoted as *AaD* (**A**ttracting-**a**nd-**D**ispersing):

$$\tilde{L}_i(\mathscr{C}_i,\mathscr{B}_i) = -\log \frac{P(\mathscr{C}_i)}{P(\mathscr{B}_i)} \tag{3.3}$$

Noting that, if we only have $P(\mathscr{C}_i)$, it will be similar to Instance Discrimination [148], but we also consider $P(\mathscr{B}_i)$ and we operate on predictions instead of features. If regarding weights of the classifier g as classes prototypes, optimizing Eq. 3.3 is not only pulling features towards their closest neighbors and pushing them away from

[†] For better understanding, we refer to $\mathscr{B}_i$ and $\mathscr{C}_i$ as index sets.

background features, but also towards (or away from) corresponding class prototypes. Therefore, we can achieve feature clustering and cluster assignment simultaneously.

To simplify the training, instead of manually and carefully sampling background features, we use all other features except z_i in the mini-batch as $\mathscr{B}_i$, which can be regarded as an estimation of the distribution of the whole dataset. We can reasonably believe that overall similarity of features in $\mathscr{C}_i$ is potentially higher than that of $\mathscr{B}_i$, even if $\mathscr{B}_i$ has intersection with $\mathscr{C}_i$ since features in $\mathscr{C}_i$ are the closest ones to feature z_i. By optimizing Eq. 3.3, we are encouraging features in $\mathscr{C}_i$, which have a higher chance of belonging to the same class, to have more similar predictions to z_i than those features in $\mathscr{B}_i$, which have a lower chance of belonging to the same class. Note all features will show up in both the first and second term; intra-cluster alignment and inter-cluster separability are expected to be achieved after training.

One problem optimizing Eq. 3.3 is that all target data are needed to compute Eq. 3.1, which is infeasible in real-world situation. Here we resort to get an upper-bound of Eq. 3.3:

$$\begin{aligned}\tilde{L}_i(\mathscr{C}_i,\mathscr{B}_i) &= -\log\frac{P(\mathscr{C}_i)}{P(\mathscr{B}_i)}\\ = -\sum_{j\in\mathscr{C}_i}[p_i^T p_j - \log(\sum_{k=1}^{N_t} e^{p_i^T p_k})] &+ \sum_{m\in\mathscr{B}_i}[p_i^T p_m - \log(\sum_{k=1}^{N_t} e^{p_i^T p_k})]\\ = -\sum_{j\in\mathscr{C}_i} p_i^T p_j + \sum_{m\in\mathscr{B}_i} p_i^T p_m &+ (N_{\mathscr{C}_i} - N_{\mathscr{B}_i})\log(\sum_{k=1}^{N_t} e^{p_i^T p_k})\end{aligned} \tag{3.4}$$

Since we set $N_{\mathscr{C}_i} < N_{\mathscr{B}_i}$, with Jensen's inequality:

$$\begin{aligned}\tilde{L}_i(\mathscr{C}_i,\mathscr{B}_i) &\le -\sum_{j\in\mathscr{C}_i} p_i^T p_j + \sum_{m\in\mathscr{B}_i} p_i^T p_m + (N_{\mathscr{C}_i} - N_{\mathscr{B}_i})(\sum_{k=1}^{N_t}\frac{1}{N_t} p_i^T p_k + \log N_t)\\ &\simeq \sum_{m\in\mathscr{B}_i} p_i^T p_m - \sum_{j\in\mathscr{C}_i} p_i^T p_j + (N_{\mathscr{C}_i} - N_{\mathscr{B}_i})(\sum_{k\in\mathscr{B}_i}\frac{p_i^T p_k}{N_{\mathscr{B}_i}} + \log N_t)\\ &= -\sum_{j\in\mathscr{C}_i} p_i^T p_j + \frac{N_{\mathscr{C}_i}}{N_{\mathscr{B}_i}}\sum_{m\in\mathscr{B}_i} p_i^T p_m + (N_{\mathscr{C}_i} - N_{\mathscr{B}_i})\log N_t\end{aligned} \tag{3.5}$$

where $N_{\mathscr{C}_i}$ and $N_{\mathscr{B}_i}$ is the number of features in $\mathscr{C}_i$ and $\mathscr{B}_i$. Note that we cannot get this upper-bound without $P(\mathscr{B}_i)$. The approximation above in the penultimate line is to estimate the average dot product using the mini-batch data. This leads to the

Table 3.2: Decomposition of methods into two terms: discriminability (*dis*) and diversity (*div*), which will be minimized for training.

Method	**Task**	***dis* term**	***div* term**
MI	SFDA&Clustering	$H(Y\|X)$	$-H(Y)$
BNM	DA&SFDA	$-\\|P\\|_F$	$-rank(P)$
NC	SFDA	$-g(W_{ij}p_i^T p_j)$	$\sum_{c=1}^{C} \mathrm{KL}(\bar{p}_c\\|q_c)$
InfoNCE	Contrastive	$-f(x)^T f(y)/\tau$	$\log(\frac{e}{\tau}+\sum_i e^{f(x_i^-)^T f(x)/\tau})$
AaD	SFDA	$-\sum_{j\in\mathscr{C}_i} p_i^T p_j$	$\sum_{m\in\mathscr{B}_i} p_i^T p_m$

surprisingly simple final objective for unsupervised domain adaptation:

$$L = \mathbb{E}[L_i(\mathscr{C}_i,\mathscr{B}_i)], \text{with } L_i(\mathscr{C}_i,\mathscr{B}_i) = -\sum_{j\in\mathscr{C}_i} p_i^T p_j + \lambda \sum_{m\in\mathscr{B}_i} p_i^T p_m \tag{3.6}$$

Note the gradient will come from both p_i and p_m. The first term aims to enforce prediction consistency between local neighbors, and the naive interpretation of second term is to disperse the prediction of potential dissimilar features, which are all other features in the mini-batch. Note that the dot product between two softmaxed predictions will be maximal when two predictions have the same predicted class and are close to one-hot vector. Our algorithm is illustrated in Algorithm. 2.

Unlike using a constant for the second term in Eq. 3.5 we empirically found that using a hyperparameter λ to decay second term (starting from 1) works better, we will adopt ***SND*** [112] to tune this hyperparameter unsupervisedly. One reason may be that the approximation inside Eq. 3.3.1 is not necessarily accurate. And as training goes on, features are gradually clustering, the role of the second term for dispersing should be weakened. Additionally, considering the current mini-batch with the correctly predicted features z_i and z_m belonging to the same class. In this case the second term in both $L_i(\mathscr{C}_i,\mathscr{B}_i)$ and $L_m(\mathscr{C}_m,\mathscr{B}_m)$ tends to push p_m to the wrong direction, while the first term in $L_m(\mathscr{C}_m,\mathscr{B}_m)$ can potentially keep current (correct) prediction unchanged. Hence, this will suppress the negative impact of the second term. We will further deepen the understanding of these two terms in the next subsection.

3.3.2 Relation to Existing Works

In this section, we will relate several popular DA, SFDA and contrastive learning methods through two objectives, *discriminability* and *diversity*. This can improve our understanding of domain adaptation methods, as well as improve the understanding of our method.

Mutual Information maximizing (MI). SHOT-IM [73] proposes to achieve source-free domain adaptation by maximizing the mutual information, which is actually widely used in unsupervised clustering [37, 43, 109]:

$$L_{MI} = H(Y|X) - H(Y) \tag{3.7}$$

which contains two terms: conditional entropy term $H(Y|X)$ to encourages unambiguous cluster assignments, and marginal entropy term $H(Y)$ to encourage cluster sizes to be uniform to avoid degeneracy. In practice, $H(Y)$ is approximated by the current mini-batch instead of using whole dataset [43, 127].

Batch Nuclear-norm Maximization (BNM). BNM [21, 22] aims to increase prediction discriminability and diversity to tackle domain shift. It is originally achieved by maximizing F-norm (for discriminability) and rank of prediction matrix (for diversity) respectively:

$$L = -\|P\|_F - rank(P) \tag{3.8}$$

In their paper, they further prove merely maximizing the nuclear norm $\|P\|_*$ can achieve these two goals simultaneously. In relation to our method, if target features are well clustering during training, we can presume the K-nearest neighbors of feature z_i have the same prediction, the first term in Eq. 3.6 can be seen as the summation of diagonal elements of matrix PP^T, which is actually the square of F-norm ($\|P\|_F = \sqrt{trace(PP^T)}$), then it is actually minimizing prediction entropy [21]. As for second term, we can regard it as the summation of non-diagonal element of PP^T, it encourages all these non-diagonal elements to be 0 thus the $rank(PP^T) = rank(P)$ is supposed to increase, which indicates larger prediction diversity [21]. In a nutshell, compared to SHOT and BNM our method first considers local feature structure to cluster target features, which can be treated as an alternative way to increase discriminability at the late training stage, meanwhile as discussed above our method is also encouraging diversity.

Neighborhood Clustering (NC). G-SFDA [157] and NRC [155], which is (will be) illustrated in Ch. 4 and Ch. 2, are based on neighborhood clustering to tackle SFDA problem. Those works basically contain two major terms in their optimizing objective: a neighborhood clustering term for prediction consistency and a marginal entropy term $H(Y)$ for prediction diversity. NRC [155] further introduces neighborhood reciprocity to weight the different neighbors. Their loss objective can be written as:

$$L_i = -\sum_{j\in\mathscr{C}_i} g(W_{ij} p_i^T p_j) + \sum_{c=1}^{C} \mathrm{KL}(\bar{p}_c || q_c), \tag{3.9}$$

$$\text{with } \bar{p}_c = \frac{1}{n_t}\sum_i p_i^{(c)} \text{ ,and } q_{\{c=1,..,C\}} = \frac{1}{C}$$

where W_{ij} will weight the importance of neighbor and $g(\cdot)$ is *log* or *identity* function. Although the first term of G-SFDA and NRC is the same as that of our final loss objective Eq. 3.6, note that our motivation is different as we simultaneously consider similar and dissimilar features, and Eq. 3.6 is deduced as an approximated upper-bound of our original objective Eq. 3.3.

And actually here $-H(Y) = \sum_{c=1}^{C} \bar{p}_c \log \bar{p}_c = \sum_{c=1}^{C} \mathrm{KL}(\bar{p}_c || q_c) - \log C$. Although the second term of those methods are favoring prediction diversity to avoid the trivial solution where all images are only assigned to some certain classes, the margin entropy term presumes the prior that whole dataset or the mini-batch is class balance/uniformly distributed, which is barely true for current benchmarks or in real-world environment. In conclusion, the above three types of methods are actually all to increase discriminability and meanwhile maximize diversity of the prediction, but through different ways.

Contrastive Learning. Here we also link our method to InfoNCE [96]), which is widely used in contrastive learning. As a recent paper [143] points out that InfoNCE loss can be decomposed into 2 terms:

$$\begin{aligned} L_{infoNCE} &= \mathbb{E}_{(x,y)\sim p_{pos}}[-f(x)^T f(y)/\tau] \\ &+ \mathop{\mathbb{E}}_{\substack{x\sim p_{data} \; \{x_i^-\}_{i=1}^M \sim p_{data}}} [\log(e^{1/\tau} + \sum_i e^{f(x_i^-)^T f(x)/\tau})] \end{aligned} \tag{3.10}$$

The first term is denoted as *alignment* term (with positive pairs) is to make positive pairs of features similar, and the second term denoted as *uniformity* term with negative pairs encouraging all features to roughly uniformly distributed in the feature space.

The Eq. 3.10 shares some similarity with all the above domain adaptation methods in that the first term is for the alignment with positive pairs and the second term is to encourage diversity. But note that the remarkable difference is that the above domain adaptation methods operate in the output (prediction) space while contrastive learning is conducted in the (spherical) feature space. Therefore, simultaneously feature representation learning and cluster assignment can be achieved for those domain adaptation methods. Note in normal contrastive learning methods, extra KNN or a linear learnable classifier needs to be deployed for final classification, while our model can directly give predictions.

We list all above methods in Tab. 3.2. Finally, returning to Eq. 3.6, we can also regard the second term as a variant of diversity loss to avoid degeneration solution, but without making any category prior assumption. Intuitively, with target features forming groups during training, the second term should play less and less important role, otherwise it may destabilize the training. This is similar to the class collision issue in contrastive learning. If our second term contains too many features belonging to the same class. Thus it is reasonable to decay the second term.

3.4 Experiments

Datasets. We conduct experiments on three benchmark datasets for image classification: Office-31, Office-Home and VisDA-C 2017. **Office-31** [110] contains 3 domains (Amazon, Webcam, DSLR) with 31 classes and 4,652 images. **Office-Home** [141] contains 4 domains (Real, Clipart, Art, Product) with 65 classes and a total of 15,500 images. **VisDA** (VisDA-C 2017) [102] is a more challenging dataset, with 12-class synthetic-to-real object recognition tasks, its source domain contains of 152k synthetic images while the target domain has 55k real object images.

Evaluation. The column **SF** in the tables denotes source-free. For Office-31 and Office-Home, we show the results of each task and the average accuracy over all tasks (*Avg* in the tables). For VisDA, we show accuracy for all classes and average over those classes (*Per-class* in the table). All results are the average of three random runs for target adaptation.

Model details. To ensure fair comparison with related methods, we adopt the backbone of a ResNet-50 [41] for Office-Home and ResNet-101 for VisDA. Specifically, we use the same network architecture as SHOT [73], BNM-S [22], G-SFDA [157] and NRC [155], *i.e.*, the final part of the network is: *fully connected layer - Batch Normalization [48] - fully connected layer with weight normalization [117].* We adopt SGD with momentum 0.9 and batch size of 64 for all datasets. The learning rate for Office-31 and Office-Home is set to 1e-3 for all layers, except for the last two newly added fc layers, where we apply 1e-2. Learning rates are set 10 times smaller for VisDA. We train 40 epochs for Office-31 and Office-Home while 15 epochs for VisDA.

There are two hyperparameters $N_{\mathcal{C}_i}$ (number of nearest neighbors) and λ, to ensure fair comparison we set $N_{\mathcal{C}_i}$ to the same number as previous works G-SFDA [157] and NRC [155], which also resort to nearest neighbors. That is, we set $N_{\mathcal{C}_i}$ to 3 on

Table 3.3: Accuracies (%) on Office-Home for ResNet50-based methods. We highlight the best result and underline the second best one.

Method	SF	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	**Avg**
ResNet-50 [41]	✗	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
MCD [115]	✗	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
CDAN [84]	✗	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
SAFN [151]	✗	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
MDD [165]	✗	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
TADA [144]	✗	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
SRDC [132]	✗	52.3	76.3	81.0	69.5	76.2	78.0	68.7	53.8	81.7	76.3	57.1	85.0	71.3
LAMDA [61]	✗	57.2	78.4	82.6	66.1	80.2	81.2	65.6	55.1	82.8	71.6	59.2	83.9	72.0
SHOT [73]	✓	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
A^2Net [149]	✓	58.4	79.0	82.4	67.5	79.3	78.9	68.0	56.2	82.9	74.1	60.5	85.0	**72.8**
G-SFDA [157]	✓	57.9	78.6	81.0	66.7	77.2	77.2	65.6	56.0	82.2	72.0	57.8	83.4	71.3
NRC [155]	✓	57.7	80.3	82.0	68.1	79.8	78.6	65.3	56.4	83.0	71.0	58.6	85.6	72.2
BNM-S [22]	✓	57.4	77.8	81.7	67.8	77.6	79.3	67.6	55.7	82.2	73.5	59.5	84.7	72.1
AaD	✓	59.3	79.3	82.1	68.9	79.8	79.5	67.2	57.4	83.1	72.1	58.5	85.4	72.7

Table 3.4: Accuracies (%) on VisDA-C (Synthesis → Real) for ResNet101-based methods. We highlight the best result and underline the second best one.

Method	SF	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	**Per-class**
ResNet-101 [41]	✗	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
CDAN+BSP [17]	✗	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
MCC [53]	✗	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
STAR [88]	✗	95.0	84.0	84.6	73.0	91.6	91.8	85.9	78.4	94.4	84.7	87.0	42.2	82.7
RWOT [150]	✗	95.1	80.3	83.7	90.0	92.4	68.0	92.5	82.2	87.9	78.4	90.4	68.2	84.0
3C-GAN [69]	✓	94.8	73.4	68.8	74.8	93.1	95.4	88.6	84.7	89.1	84.7	83.5	48.1	81.6
SHOT [73]	✓	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
A^2Net [149]	✓	94.0	87.8	85.6	66.8	93.7	95.1	85.8	81.2	91.6	88.2	86.5	56.0	84.3
G-SFDA [157]	✓	96.1	88.3	85.5	74.1	97.1	95.4	89.5	79.4	95.4	92.9	89.1	42.6	85.4
NRC [155]	✓	96.8	91.3	82.4	62.4	96.2	95.9	86.1	80.6	94.8	94.1	90.4	59.7	85.9
HCL [45]	✓	93.3	85.4	80.7	68.5	91.0	88.1	86.0	78.6	86.6	88.8	80.0	74.7	83.5
AaD	✓	97.4	90.5	80.8	76.2	97.3	96.1	89.8	82.9	95.5	93.0	92.0	64.7	**88.0**

Table 3.5: (**Left**) Accuracies (%) on Office-31 for ResNet50-based methods. We highlight the best result and underline the second best one. (**Right**) Ablation study on number of nearest neighbors $N_{\mathscr{C}_i}$. We highlight the best score and underline the second best one.

Method	SF	A→D	A→W	D→W	W→D	D→A	W→A	Avg
MCD [115]	✗	92.2	88.6	98.5	100.0	69.5	69.7	86.5
CDAN [84]	✗	92.9	94.1	98.6	100.0	71.0	69.3	87.7
MDD [165]	✗	90.4	90.4	98.7	99.9	75.0	73.7	88.0
DMRL [147]	✗	93.4	90.8	99.0	100.0	73.0	71.2	87.9
MCC [53]	✗	95.6	95.4	98.6	100.0	72.6	73.9	89.4
SRDC [132]	✗	95.8	95.7	99.2	100.0	76.7	77.1	**90.8**
SHOT [73]	✓	94.0	90.1	98.4	99.9	74.7	74.3	88.6
3C-GAN [69]	✓	92.7	93.7	98.5	99.8	75.3	77.8	89.6
NRC [155]	✓	96.0	90.8	99.0	100.0	75.3	75.0	89.4
HCL [45]	✓	94.7	92.5	98.2	100.0	75.9	77.7	89.8
BNM-S [22]	✓	93.0	92.9	98.2	99.9	75.4	75.0	89.1
AaD	✓	96.4	92.1	99.1	100.0	75.0	76.5	89.9

$N_{\mathscr{C}_i}$	**Avg**
Office-31	
1	89.1
2	89.5
3	**89.9**
Office-Home	
1	72.2
2	72.6
3	**72.7**

$N_{\mathscr{C}_i}$	**Per-class**
VisDA	
3	86.7
4	87.4
5	**88.0**
6	**88.0**
7	**88.0**

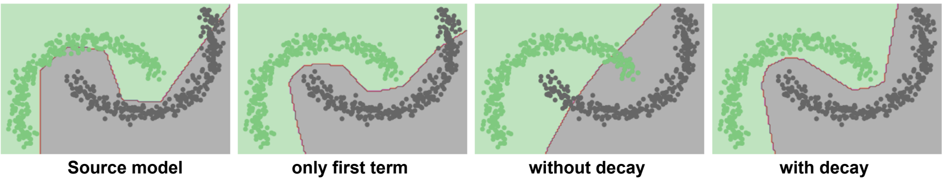

Figure 3.1: Visualization of decision boundary on target data with different training objective.

Office-31 and Office-Home, 5 on VisDA. For λ, we set it as $\lambda = (1 + 10 * \frac{iter}{max\_iter})^{-\beta}$, where the decay factor β controls the decaying speed. *We directly apply* ***SND*** *[112] to select β unsupervisedly*. Based on SND we set β to 0 on Office-Home, 2 on Office-31 and 5 on VisDA.

3.4.1 Results and Analysis

Quantitative Results. As shown in Tables 3.3-3.5(*Left*), where the top part shows results for the source-present methods that use source data during adaptation, and the bottom part shows results for the source-free DA methods. On Office-31 and VisDA, our method gets state-of-the-art performance compared to existing source-free domain adaptation methods, especially on VisDA our method outperforms others by a large margin (2.1% compared to NRC). And our method achieves similar results on Office-Home compared to the more complex A^2Net method (*which combines three classifiers and five objective functions*). The reported results clearly demonstrate the efficiency of the proposed method for source-free domain adaptation. It also achieves similar or better results compared to domain adaptation methods with access to source data on both Office-Home and VisDA. Note the extension of SHOT called SHOT++ [77] deploys extra self-supervised training and semi-supervised learning, which are general to improve the results (*an evidence is that the source model after these 2 tricks gets huge improvement, e.g., 60.2% improves to 66.6% on Office-Home.*), we do not list it here for fair comparison.

Toy dataset. We carry out an experiment on the twinning moona dataset to ablate the influence of two terms in our objective Eq. 3.6. For the twinning moons dataset, the data from the source domain are represented by two inter-twinning moons, which contain 300 samples each. Data in the target domain are generated through rotating source data by 30°. The domain shift here is instantiated as the rotation degree. First we train the model with 3 linear layers only on the source domain, and test the model

Table 3.6: **Unsupervised hyperparameter selection of β with *SND* [112]**, larger *SND* should correspond to better target model.

Office-31		
β	*SND*↑	**Avg**
0	4.1366	88.0
0.25	4.3016	89.7
1	4.4494	**89.9**
2	**4.4501**	**89.9**

Office-Home		
β	*SND*↑	**Avg**
0	**3.7515**	**72.7**
0.25	3.7402	72.6
0.5	3.7252	72.0
1	3.6923	70.6

VisDA		
β	*SND*↑	**Per-class**
0	8.1823	77.5
1	8.2584	83.8
2	8.3214	86.7
3	8.3311	87.6
4	8.3540	88.0
5	**8.3543**	88.0
7	8.3530	**88.1**

on all domains. As shown in the first image in Fig. 3.1, the source model performs badly on target data. Then we conduct several variants of our method to train the model. The visualization of the decision boundary in Fig. 3.1 indicates that both terms in Eq. 3.6 are necessary, and decay of second term is shown to be important.

Number of nearest neighbors ($N_{\mathcal{C}_i}$). For the number of nearest neighbors used for the first term in Eq. 3.6, we show in Tab. 3.5 (*Right*) our method is robust to the choice of $N_{\mathcal{C}_i}$, as the results imply that a reasonable choice of $N_{\mathcal{C}_i}$ (such as 3) works quite well on all datasets, since only considering few neighbors (such as 1/2) may be too noisy if all of them are misclassified, while setting $N_{\mathcal{C}_i}$ too larger may also potentially include samples of other categories. For larger dataset such as VisDA we can choose a relatively larger $N_{\mathcal{C}_i}$. Note the reason why we choose $N_{\mathcal{C}_i}$ as 5 in main experiments is to compare fairly with G-SFDA [157] and NRC [155].

Decay factor β. According to the analysis in Sec. 3.3.2, the second term acts like a diversity term to avoid that all target features collapse to a limited set of categories. The role of the second term should be weakened during the training, but how to decay the second term is non-trivial. We directly adopt ***SND*** [112] which computes Soft Neighborhood Density for unsupervised hyperparameter selection of β. The method is unsupervised and larger ***SND*** predicts a better target models. The results of ***SND*** with different β are shown in Tab. 3.6, the results prove that ***SND*** works well to choose optimal β.

Runtime analysis. Instead of storing all features in the memory bank, we can only stores a limited number of target features, by updating the memory bank at the end of each iteration by taking the n (batch size) embeddings from the current training

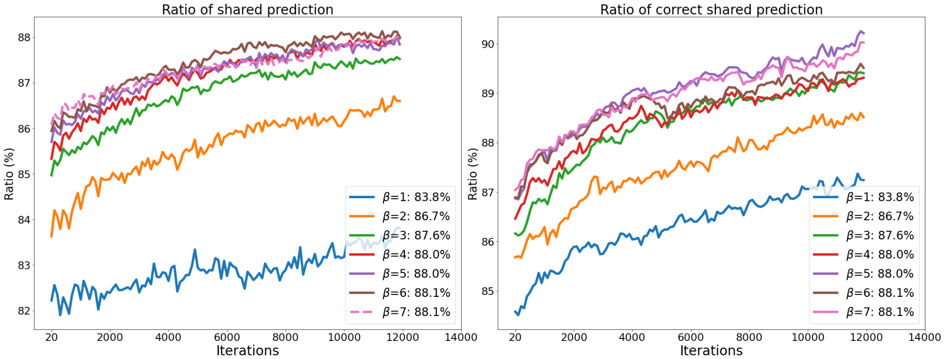

Figure 3.2: (**Left**) Ratio of features which have 3 nearest neighbor features sharing the same predicted label. (**Right**) Ratio among **above features** which have 3 nearest neighbor features sharing the same and **correct** predicted label.

Table 3.7: Runtime analysis on SHOT and our method. For SHOT, pseudo labels are computed at each epoch. 10% and 5% denote the percentage of target features which are stored in the memory bank.

VisDA	**Runtime** (s/epoch)	**Per-class** (%)
SHOT	618.82	82.9
AaD	520.13	88.0
AaD(10% for memory bank)	490.21	87.6
AaD(5% for memory bank)	482.77	87.5

iteration and concatenating them at the end of the memory bank, and discard the oldest n elements from the memory bank. We report the results with this type of memory bank of different buffer size in the Table 3.7. The results show that indeed this could be an efficient way to reduce computation on very large datasets.

Degree of clustering during training. We also plot how features are clustered with different decaying factors β on VisDA in Fig. 3.2. The left one shows the ratio of features which have 3-nearest neighbors all sharing the same prediction, which indicates the degree of clustering during training, and the right one shows the ratio among above features which have 3-nearest neighbor features sharing the same and *correct* predicted label. Those curves in Fig. 3.2 *left* show that the target features are clustering, and those in Fig. 3.2 *right* indicate that clear category boundaries are emerging. The numbers in the legends denote the deployed β and the corresponding final accuracy. From the figures we can draw the conclusion that with a larger decay factor β on VisDA, features are quickly clustering and forming inter-class boundaries, since the ratio of features which share the same and correct prediction with neighbors are increasing faster. When decaying factor β is too small, meaning training signal from the second term is strong, the clustering process is actually impeded. The curves in Fig. 3.2 (*left*) signify that this ratio can also be used to choose β with higher performance unsupervisedly.

Source-free partial-set and open-set DA. We provide additional results under source-free partial-set and open-set DA (PDA and ODA) setting in Tab. 3.8 and Tab. 3.9 respectively, where the open-set detection in ODA follows the same protocol to detect unseen categories as SHOT. On ODA, instead of reporting average *per-class* accuracy $OS = \frac{|\mathscr{C}_s| \times OS^*}{|\mathscr{C}_s|+1} + \frac{1 \times UNK}{|\mathscr{C}_s|+1}$ where $|\mathscr{C}_s|$ is the number of known categories on source domain, we report results of $HOS = \frac{2 \times OS^* \times UNK}{OS^* + UNK}$, which is *harmonic mean* between known categories accuracy OS^* and unknown accuracy *UNK*. As pointed out by [9], OS is problematic since this metric can be quite high even when unknown class accuracy UNK is 0, while unknown category detection is the key part in open-set DA. We reproduce SHOT under open-set DA and report results of OS^*, UNK and HOS in Tab. 3.8, which shows our method gets much better balance between known and unknown accuracy.

3.5 Conclusion

We proposed to tackle source-free domain adaptation by encouraging similar features in feature space to have similar predictions while dispersing predictions of dissimilar features in feature space, to achieve simultaneously feature clustering and cluster assignment. We introduced an upper bound to our proposed objective, resulting in two simple terms. Further we showed that we can unify several popular domain adaptation, source-free domain adaptation and contrastive learning methods from the perspective of discriminability and diversity. The approach is simple but achieves state-of-the-art performance on several benchmarks, and can be also adapted to source-free open-set and partial-set domain adaptation.

Table 3.8: Accuracy on Office-Home using ResNet-50 as backbone for **Source-free open-set DA**. *OS**, *UNK* and *HOS* mean average per-class accuracy across known classes, unknown accuracy and harmonic mean between known and unknown accuracy respectively.

	Ar → Cl			Ar → Pr			Ar → Rw			Cl→ Ar			Cl → Pr			Cl → Rw					
	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS			
SHOT	67.0	28.0	39.5	81.8	26.3	39.8	87.5	32.1	47.0	66.8	46.2	54.6	77.5	27.2	40.2	80.0	25.9	39.1			
AaD	50.7	66.4	**57.6**	64.6	69.4	**66.9**	73.1	66.9	**69.9**	48.2	81.1	**60.5**	59.5	63.5	**61.4**	67.4	68.3	**67.8**			
	Pr → Ar			Pr → Cl			Pr → Rw			Rw→ Ar			Rw → Cl			Rw → Pr			**Avg**.		
	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS
SHOT	66.3	51.1	57.7	59.3	31.0	40.8	85.8	31.6	46.2	73.5	50.6	59.9	65.3	28.9	40.1	84.4	28.2	42.3	74.6	33.9	45.6
AaD	47.3	82.4	**60.1**	45.4	72.8	**55.9**	68.4	72.8	**70.6**	54.5	79.0	**64.6**	49.0	69.6	**57.5**	69.7	70.6	**70.1**	58.2	71.9	**63.6**

Table 3.9: Accuracy on Office-Home using ResNet-50 as backbone for **Source-free partial-set DA (PDA)**.

PDA	Ar→Cl	Ar→Pr	Ar→Re	Cl→Ar	Cl→Pr	Cl→Re	Pr→Ar	Pr→Cl	Pr→Re	Re→Ar	Re→Cl	Re→Pr	**Avg.**
SHOT-IM	57.9	83.6	88.8	72.4	74.0	79.0	76.1	60.6	90.1	81.9	**68.3**	**88.5**	76.8
SHOT	64.8	**85.2**	92.7	76.3	**77.6**	**88.8**	**79.7**	64.3	89.5	80.6	66.4	85.8	79.3
AaD	**67.0**	83.5	**93.1**	**80.5**	76.0	87.6	78.1	**65.6**	**90.2**	**83.5**	64.3	87.3	**79.7**

4 Generalized Source-free Domain Adaptation[*]

4.1 Introduction

Though achieving great success, deep neural networks typically require a large amount of labeled data for training. However, collecting labeled data is often laborious and expensive. To tackle this problem, *Domain Adaptation* (DA) methods aim to transfer knowledge learned from label-rich datasets (source domains) to other unlabeled datasets (target domains), by reducing the domain shift between labeled source and unlabeled target domains.

A crucial requirement in most DA methods is that they require access to the source data during adaptation, which is often impossible in many real-world applications, such as deploying domain adaptation algorithms on mobile devices where the computation capacity is limited, or in situations where data-privacy rules limit access to the source domain. Because of its relevance and practical interest, the *source-free domain adaptation* (SFDA) setting, where instead of source data only source pretrained model is available, has started to get traction recently [58, 59, 69, 73, 156]. Among these methods, SHOT [73] and 3C-GAN [69] are most related to this chapter which is for close-set DA where source and target domains have the same categories. 3C-GAN [69] is based on target-style image generation by a conditional GAN, and SHOT [73] proposes to transfer the source hypothesis, i.e. the fixed source classifier, to the target data, together with maximizing mutual information.

However, in many practical situations models should perform well on both the target and source domain. For example, we would desire a recognition model deployed in an urban environment which works well for all four seasons (domains) after adapting model to the seasons sequentially. As shown in [160], the source performance of some DA methods will degrade after adaptation even with source data always at hand. And the current SFDA methods focus on the target domain by fine tuning the source model, leading to forgetting on old domains. Thus, existing methods cannot handle the situation described above. A simple way to address this setting is by just storing the source and target model, however, we aim for memory-efficient solutions that scale sub-linear with the number of domains. Therefore, in this chapter, we propose a

[*] This chapter is based on a publication in the International Conference on Computer Vision (ICCV), 2021 [157]

new DA paradigm where the model is expected to perform well on all domains after source-free domain adaptation. We call this setting *Generalized Source-free Domain Adaptation* (G-SFDA). For simplicity, in the paper we will first focus on a single target domain, and then we describe how to extend to Continual Source-free Domain Adaptation.

In this chapter, to perform adaptation to the target domain without source data, we first propose Local Structure Clustering (LSC), that clusters each target feature together with its nearest neighbors. The motivation is that one target feature should have similar prediction with its semantic close neighbors. To keep source performance, we propose to use sparse domain attention (SDA), applied to the output of the feature extractor, activating different feature channels depending on the particular domain. The source domain attention will be used to regularize the gradient during target adaptation to prevent forgetting of source information. With LSC and SDA, the adapted model can achieve excellent performance on both source and target domains. In the experiments, we show that for target performance our method is on par with or better than existing DA and SFDA methods on several benchmarks, specifically achieving state-of-the-art performance on VisDA (85.4%), while simultaneously keeping good source performance. We also extend our method to Continual Source-free Domain Adaptation, where there is more than one target domain, further demonstrating the efficiency of our method.

We summarize our contributions as follows:

- We propose a new domain adaptation paradigm denoted as Generalized Source-free Domain Adaptation (G-SFDA), where the source-pretrained model is adapted to target domains while keeping the performance on the source domain, in the absence of source data.

- We propose local structure clustering (LSC) to achieve source-free domain adaptation, which utilizes local neighbor information in feature space.

- We propose Sparse domain attention (SDA) which activates different feature channels for different domains, and regularizes the gradient of back propagation during target adaptation to keep information of the source domain.

- In experiments, we show that where existing methods suffer from forgetting and obtain bad performance on the source domain, our method is able to maintain source domain performance. Furthermore, when focusing on the target domain our method is on par with or better than existing methods, especially we achieve state-of-the-art target performance on VisDA.

4.2 Related Works

Here we discuss related domain adaptation settings.

Domain Adaptation. Early domain adaptation methods such as [83, 130, 136] adopt moment matching to align feature distributions. Inspired by adversarial learning, DANN [31] formulates domain adaptation as an adversarial two-player game. CDAN [84] trains a deep networks conditioned on several sources of information. DIRT-T [124] performs domain adversarial training with an added term that penalizes violations of the cluster assumption. Domain adaptation has also been tackled from other perspectives. MCD [115] adopts prediction diversity between multiple learnable classifiers to achieve local or category-level feature alignment between source and target domains. DAMN [6] introduces a framework where each domain undergoes a different sequence of operations. AFN [151] shows that the erratic discrimination of target features stems from much smaller norms than those found in source features. SRDC [132] proposes to directly uncover the intrinsic target discrimination via discriminative clustering to achieve adaptation. The most relevant paper to our LSC is DANCE [111], which is for universal domain adaptation and based on neighborhood clustering. But they are based on instance discrimination [148] between all features, while our method applies consistency regularization on only a few semantically close neighbors.

Source-free Domain Adaptation. Normal domain adaptation methods require access to source data during adaptation. Recently, there are several methods investigating source-free domain adaptation. USFDA [58] and FS [59] explore the source-free universal DA [161] and open-set DA [116], DECISION [3] is for multi-source DA. Related to our work are SHOT [73] and 3C-GAN [69], both for close-set DA. SHOT proposes to fix the source classifier and match the target features to the fixed classifier by maximizing mutual information and pseudo label. 3C-GAN synthesizes labeled target-style training images based on conditional GAN. Recently, BAIT [156] extends diverse classifier based domain adaptation methods to also be applicable for SFDA. Though achieving good target performance, these methods cannot maintain source performance after adaptation. Other than these methods, we aim to maintain source-domain performance after adaptation.

Continual Domain Adaptation. Continual learning (CL) [55, 72, 87, 90] specifically focuses on avoiding catastrophic forgetting when learning new tasks, but it is not tailored for DA since new tasks in CL usually have labeled data. Recently, a few works [8, 91, 129] have emerged that aim to tackle the *Continual Domain Adaptation*

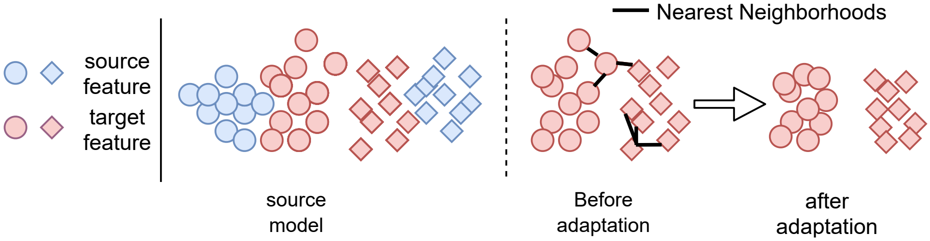

Figure 4.1: Local Structure Clustering (LSC). Some target features from source model will deviate from dense source feature regions due to domain shift. LSC aims to cluster target features by its semantically close neighbors (linked by black line).

(CDA) problem. [8] uses sample replay to avoid forgetting together with domain adversarial training, [91] builds a domain relation graph, and [129] builds a domain-specific memory buffer for each domain to regularize the gradient on both target and memory buffer. Although these methods achieve good performance, they all demand access to source data. And [60] is source-free but they focus on class incremental single target domain adaptation where there is only one-shot labeled target data per class, while our method is related to domain incremental learning and can be deployed for continual source-free domain adaptation.

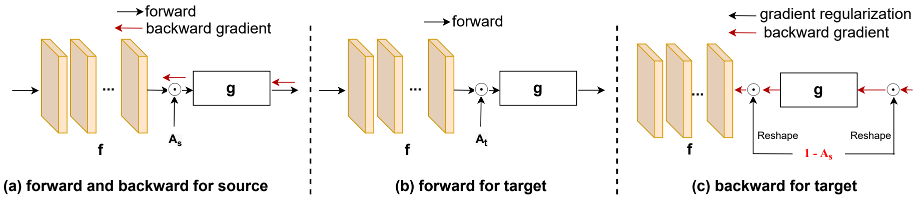

Figure 4.2: (a-c): Forward and Backward pass for two domains. **f**, **g** denote feature extractor, classifier. $\mathscr{A}_s$ and $\mathscr{A}_t$ are the sparse source and target domain attention.

4.3 Methods

In this section, we first propose an approach for source-free unsupervised domain adaptation. Then we introduce our method to prevent forgetting of the knowledge of the source model. Next, we elaborate how to unify the two modules to address generalized source-free domain adaptation (G-SFDA), and train a domain classifier for domain-agnostic evaluation. Finally, we extend our method to continual source-free domains. We call the proposed method as ***GSFDA***.

4.3.1 Problem Setting and Notations

We denote the labeled source domain data with n_s, the samples as $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, where the y_i^s is the corresponding label of x_i^s, and the unlabeled target domain data with n_t samples as $\mathcal{D}_t = \{x_j^t\}_{j=1}^{n_t}$. The number of classes is C. In the source-free setting we consider here $\mathcal{D}_s$ is only available during model pretraining. Our method is based on a neural network, which we split into two parts: a feature extractor f, and a classifier g that only contains one fully connected layer. The output of network is denoted as $p(x) = g(f(x)) \in \mathcal{R}^C$.

4.3.2 Local Structure Clustering

Most domain adaptation methods aim to align the feature distributions of the source and target domain. In source-free unsupervised domain adaptation (SFDA) this is not evident since the algorithm has no longer access to source domain data during adaptation. We identify two main sources of information that the trained source model provides with respect to the target data: a class prediction $p(x)$ and a location in the feature space $f(x)$. The main idea behind our method is that we expect the features of the target domain to be shifted with respect to the source domain, however, we expect that classes still form clusters in the feature space, and as such, we aim to move clusters of data points to their most likely class prediction.

Our algorithm is illustrated in Fig. 4.1 (left). Some target features (at the start of adaptation) deviate from the corresponding dense source feature region due to domain shift. This could result in wrong prediction of the classifier. However, we assume that the target features of the same class are clustered together. Therefore, the nearest neighbors of target features have a high probability to share category labels. To exploit this fact, we encourage features close in feature space to have similar prediction to their nearest neighbors. As a consequences clusters of points that are close in feature space will move jointly towards a common class. As shown in the right of Fig. 4.1, this process can correctly classify target features which would otherwise have been wrongly classified.

To find the semantically close neighbors, we build a feature bank $\mathcal{F} = \{(f(x_i))\}_{x_i \in \mathcal{D}_t}$ which stores the target features. This is similar to methods in unsupervised learning [44, 137, 148, 175] or domain adaptation [111]. The method [111] is for universal domain adaptation, and considers similarity based on instance discrimination [148] between all features in their loss function, and [44, 137, 175] perform unsupervised learning using neighborhood information. The work [137] needs pretext training and the nearest neighborhood *images* is retrieved only once by the embedding network from the pretext stage to train another classification network, while [44, 175] are also based on instance discrimination between all target features, and utilize neighbourhood selection to further improve the cluster performance. Different from them, we only use a few neighbors from the feature bank to cluster the target features with a consistency regularization.

Next, we build a score bank $\mathscr{S} = \{(g(f(x_i))\}_{x_i \in \mathscr{D}_t}$ storing corresponding softmaxed prediction scores. The local structure clustering is achieved by encouraging consistent predictions between the k-nearest *features* applying the following loss:

$$\begin{aligned}
\mathscr{L}_{\text{LSC}} &= -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} log[p(x_i)\cdot s(\mathscr{N}_k)] + \sum_{c=1}^{C}\text{KL}(\bar{p}_c||q_c) \\
\mathscr{N}_{\{1,..,K\}} &= \{\mathscr{F}_j|\ top\text{-}K\big(cos\big(f(x_i),\mathscr{F}_j\big), \forall \mathscr{F}_j\ \in \mathscr{F}\big)\}, \\
\bar{p} &= \frac{1}{n}\sum_{i=1}^{n} p_c(x_i)\ , \text{and}\ q_{\{c=1,..,C\}} = \frac{1}{C}
\end{aligned} \tag{4.1}$$

Here, we first find the k-nearest neighbors $\mathscr{N}$ in the feature bank for each current target feature based on the cosine similarity. We minimize the negative log value of the dot product between prediction score of the current target sample x_i and the stored prediction scores $s(\mathscr{N}_k)$ of $\mathscr{N}$, which is the first term in Eq. 4.1, aiming to encourage consistent predictions between the feature and its a few neighbors. The second term avoids the degenerated solution [34, 122], where the prediction of classes in the target data is highly imbalanced, by encouraging prediction balance. Here p_c is the empirical label distribution; it represents the predicted possibility of class c and q is a uniform distribution. And we simply replace the old items in the bank with the new ones corresponding to current mini-batch. In the experiments, we will prove the effectiveness of the proposed LSC by verifying whether the nearest neighbors are sharing the right predicted label.

4.3.3 Sparse Domain Attention

Under the G-SFDA setting, we want to not only have high target performance, but maintain source performance without accessing source data. Our work is inspired by continual learning (CL) methods [1, 90, 119] which put constraints on each layer for leaving out capacity for new tasks and prevent forgetting of previous tasks. We propose to only activate parts of the feature channels of $f(x) \in \mathscr{R}^d$ for different domains, by a sparse domain attention (SDA) vector $\mathscr{A}_{i\in\{s,t\}} \in \mathscr{R}^d$, which contain close-to binary values that will mask the output of the feature extractor. Inspired by [119], we adopt an embedding layer to automatically produce the domain adaptation.

$$\mathscr{A}_{\mathrm{i}\in[s,t]} = \sigma(100 \cdot e_i) \tag{4.2}$$

where e_i is the output of an embedding layer, σ is $sigmoid$ function, and the constant 100 is to ensure a near-binary output, but still differentiable. $\mathscr{A}_s$ and $\mathscr{A}_t$ are both trained on the source domain and are fixed during the adaptation to the target domain. Furthermore, when training on source, we use sparsity regularization and gradient compensation for the embedding layer just like [119]. Thus, we use SDA to build domain specific information flows where some channels are specific for each domain. We can maintain the source information by regularizing the gradient flowing into channels that are activated in the source mask.

For training the source domain, we apply the source attention $\mathscr{A}_s$, as shown in Fig. 4.2(a), the output is $g(f(x) \odot \mathscr{A}_s)$. In Fig. 4.2(b), we show that when adapting to the target domain, we use the sparse target attention $\mathscr{A}_t$ for the forward pass. To prevent forgetting, there should be no update to the feature channels which are present in $\mathscr{A}_s$. The reasons are twofold: firstly, the information of those channels is the only source information provided during source-free adaptation to the target domain; keeping this information may boost target adaptation, and secondly more importantly, under the G-SFDA setting we hope to keep the source performance after adapting, therefore target adaptation should not disturb the information flowing to those channels of feature associated with source domain. As shown in Fig. 4.2(c), during target adaptation we propose to use source attention A_s to regularize the gradients flowing to the classifier and feature extractor during back propagation:

$$W_{f_l} \leftarrow W_{f_l} - (\bar{\mathscr{A}_s} \mathbb{1}_h^T) \odot \frac{\partial \mathscr{L}}{\partial W_{f_l}} \tag{4.3}$$

$$W_g \leftarrow W_g - \frac{\partial \mathscr{L}}{\partial W_g} \odot (\mathbb{1}_C \bar{\mathscr{A}_s}^T) \tag{4.4}$$

where $\odot$ denotes element wise multiplication, $\mathbb{1}_k$ is an all-ones vector of dimensionality k, $\bar{\mathscr{A}_s} = 1 - \mathscr{A}_s$, $W_{f_l} \in \mathscr{R}^{d \times h}$ is the weight of the last layer in feature extractor, $W_g \in \mathscr{R}^{C \times d}$ is the weight of the classifier. Here the source attention $\mathscr{A}_s$ is used to regularize the gradient flowing into the source activated channels (for feature extractor) and also the corresponding neurons in the classifier. With Eq. 4.3 and Eq. 4.4, the source information is expected to be preserved.

In continual learning literature the masking of weights [89, 90] and activations [1, 93, 119] has been studied. Our method is related to the activation mask methods. However, other then these methods, our masking only prevents forgetting in the last two layers W_{f_l} and W_g. We ensure that the features that are crucial for source domain performance are only minimally changed, and that the target domain specific features are used to address the domain shift. Our approach does not prevent all forgetting of the source domain, since we do not regularize the gradient of the inner layers in feature extractor.

4.3.4 Unified Training

Algorithm 3 Generalized Source-free Domain Adaptation

```
Require: $\mathscr{D}_s$ (only for source model training), $\mathscr{D}_t$
Pre-train model on $\mathscr{D}_s$ with both $\mathscr{A}_s$ and $\mathscr{A}_t$ from SDA
Build feature bank $\mathscr{F}$ and score bank $\mathscr{S}$ for $\mathscr{D}_t$
while Adaptation do
    Sample batch $\mathscr{T}$ from $\mathscr{D}_t$
    Update $\mathscr{F}$ and $\mathscr{S}$ corresponding to current batch $\mathscr{T}$
    Compute $\mathscr{L}_{lsc}$ based on $\mathscr{F}$ and $\mathscr{S}$        ▷ Eq. 4.1,4.5
    Update network with SDA regularization        ▷ Eq. 4.3,4.4
end while
```

In this section, we first illustrate how to unify the training with SDA and LSC. As illustrated in Algorithm 3, first we train the model on $\mathscr{D}_s$ with the cross-entropy loss, with both source and target domain attention $\mathscr{A}_s$, $\mathscr{A}_t$, this is to provide a good initialization for target adaptation where only $\mathscr{A}_t$ is engaged. Then, we adapt the source model to the target domain with target attention $\mathscr{A}_t$ and only access to $\mathscr{D}_t$ with Eq. 4.1. During backpropagation we regularize the gradients according to Eq. 4.3 and Eq. 4.4. Unlike training with only LSC in Sec. 4.3.2, here we build the feature bank as $\mathscr{F} = \{(f(x_i) \odot \mathscr{A}_t)\}_{x_i \in \mathscr{D}_t}$, where we abandon the irrelevant channels since those channels will not contribute to current prediction and may contain noise. And for the same reason when using k-nearest neighbors, we also apply the target attention to the feature, so the $\mathscr{N}_{\{1,..,K\}}$ in Eq. 4.1 turns into:

$$\mathscr{N}_{\{1,..,K\}} = \{\mathscr{F}_j |\ top\text{-}K\big(cos\big(f(x_i) \odot \mathscr{A}_t, \mathscr{F}_j\big), \forall \mathscr{F}_j \in \mathscr{F}\big)\} \tag{4.5}$$

Domain-ID estimation. In the experimental section, we will consider both G-SFDA with (*domain-aware*) and without (*domain-agnostic*) access to the domain-id at inference time. In the more challenging setting the domain-ID is not available, and needs to be estimated. Therefore, we propose to train a domain classifier which takes in feature $f(x)$ to estimate the domain-ID of the test samples, by only storing a very small set of images of the source domain. We will show in the experiments that

Table 4.1: Accuracies (%) on VisDA-C (Synthesis → Real) for ResNet101-based unsupervised domain adaptation methods. Source-free means setting without access to source data during adaptation. Underlined results are second highest result. Our results are using target attention $\mathscr{A}_t$. SF means source-free. **GSFDA means method without domainID.**

Method	**SF**	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
source [41]	×	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
ADR [114]	×	94.2	48.5	84.0	72.9	90.1	74.2	92.6	72.5	80.8	61.8	82.2	28.8	73.5
CDAN [84]	×	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
BSP [17]	×	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
SWD [62]	×	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
MDD [165]	×	-	-	-	-	-	-	-	-	-	-	-	-	74.6
IA [52]	×	-	-	-	-	-	-	-	-	-	-	-	-	75.8
DMRL [147]	×	-	-	-	-	-	-	-	-	-	-	-	-	75.5
MCC [53]	×	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
DANCE [111]	×	-	-	-	-	-	-	-	-	-	-	-	-	70.4
DANCE [111]	√	-	-	-	-	-	-	-	-	-	-	-	-	70.2
SHOT [73]	√	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
3C-GAN [69]	√	94.8	73.4	68.8	74.8	93.1	95.4	88.6	84.7	89.1	84.7	83.5	48.1	81.6
GSFDA	√	96.1	88.3	85.5	74.1	97.1	95.4	89.5	79.4	95.4	92.9	89.1	42.6	**85.4**

we obtain similar results in the challenging domain-agnostic setting as in the easier domain-aware setting.

4.3.5 Continual Source-free Domain Adaptation

Here we illustrate how to extend our method to continual source-free domain adaptation, where the model is adapted to a sequence of target domains with only access to current target domain data. Assuming that there are N_t target domains. For source pretraining we train with all domain attention $\mathscr{A}_s$ and $\{\mathscr{A}_{t_i}\}_{i=1..N_t}$ from SDA, for a good initialization as mentioned before. And when adapting to the j-th target domain, we compute $\mathscr{A}'$ which considers all domain attention except the current one. We replace the $\mathscr{A}_s$ in Eq. 4.3 and Eq. 4.4 with $\mathscr{A}'$ for current gradient regularization:

$$\mathscr{A}' = \max(\mathscr{A}', \mathscr{A}_{t_i}), \; \forall i \in \{1, .., N_t\} \setminus j \tag{4.6}$$

where max is an element-wise operation and $\mathscr{A}'$ is initialized from $\mathscr{A}_s$. Using $\mathscr{A}'$ for gradient regularization means training on one target domain should not influence others.

Table 4.2: Accuracies (%) on Office-Home for ResNet50-based unsupervised domain adaptation methods. Source-free means source-free setting without access to source data during adaptation. Underline means the second highest result. Our results are using target attention $\mathscr{A}_t$.

Method	**SF**	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	**Avg**
ResNet-50 [41]	×	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
MCD [115]	×	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
CDAN [84]	×	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
MDD [165]	×	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
IA [52]	×	56.0	77.9	79.2	64.4	73.1	74.4	64.2	54.2	79.9	71.2	58.1	83.1	69.5
BNM [21]	×	52.3	73.9	80.0	63.3	72.9	74.9	61.7	49.5	79.7	70.5	53.6	82.2	67.9
BDG [152]	×	51.5	73.4	78.7	65.3	71.5	73.7	65.1	49.7	81.1	74.6	55.1	84.8	68.7
SRDC [132]	×	52.3	76.3	81.0	69.5	76.2	78.0	68.7	53.8	81.7	76.3	57.1	85.0	71.3
SHOT [73]	√	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	**71.8**
GSFDA w/ domainID	√	57.9	78.6	81.0	66.7	77.2	77.2	65.6	56.0	82.2	72.0	57.8	83.4	71.3

Table 4.3: Accuracy (%) of each method on Office-31 dataset using ResNet-50 as backbones. Randomly specifying 0.8/0.2 train/test split for the source dataset, the source accuracy is reported on the test set.

		Office-31																				
		A → D			A → W			D → A			D → W			W → A			W → D			**Avg.**		
	SF	S	T	**H**	S	T	**H**	S	T	**H**	S	T	**H**	S	T	**H**	S	T	**H**	S	T	**H**
Source model		88.8	79.7	84.0	88.8	76.0	81.9	97.0	63.7	76.9	97.0	94.0	95.5	100.0	64.8	78.6	100.0	98.0	99.0	**95.3**	79.4	86.0
SHOT	√	80.1	91.2	85.3	79.3	91.1	84.8	83.0	74.6	78.6	99.0	98.1	98.5	84.9	74.5	79.4	96.2	99.0	97.6	87.1	88.1	87.4
GSFDA	√	87.8	89.6	88.7	86.5	92.0	89.2	97.0	75.0	84.6	97.0	98.5	97.7	99.4	75.0	85.5	99.4	99.8	99.6	94.5	**88.3**	**90.9**

Table 4.4: Accuracy (%) of each method on VisDA dataset using ResNet-101 as backbone under **G-SFDA** setting. Randomly specifying 0.9/0.1 train/test split for the source dataset. T and S denote accuracy on target and source domain. **w/** and **w/o** denote whether our method has access to domain-ID during evaluation.

	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	**Avg.**	
	S/T	S/T	S/T	S/T	S/T	S/T	S /T	S /T	S /T	S /T	S /T	S /T	**S /T**	**H**
Source	99.9/70.6	99.9/15.6	99.3/45.6	99.1/80.9	99.9/63.0	99.9/5.1	99.4/79.2	100/24.9	99.9/64.0	100/39.6	99.3/84.8	98.3/6.3	**99.6** /48.1	64.9
SHOT	99.3/94.4	97.3/85.8	34.9/78.4	47.3/55.2	94.4/93.9	93.2/95.0	38.3/81.5	94.4/79.5	99.1/89.8	92.7/90.1	55.4/85.6	62.0/56.8	75.7/82.2	78.8
w/	99.7/95.9	98.7/88.1	98.4/85.4	80.0/72.5	94.6/96.1	98.4/93.7	76.2/88.5	97.8/80.6	98.8/92.3	99.9/92.2	75.6/87.6	67.3/44.8	90.4/**85.0**	**87.6**
w/o	99.7/95.4	98.7/87.7	98.4/85.7	80.0/71.5	94.6/96.1	98.4/94.8	76.2/89.2	97.8/80.4	98.8/92.0	99.9/88.6	75.6/87.4	67.3/44.1	90.4/84.4	87.3

Table 4.5: Accuracy (%) of each method on Office-Home dataset using ResNet-50 as backbone under **G-SFDA** setting. Randomly specifying 0.8/0.2 train/test split for the source dataset. T and S denote accuracy on target and source domain. domain-ID means having access to domain-ID during evaluation, w/o domain-ID means using the estimated domain-ID from domain classifier.

	Ar → Cl			Ar → Pr			Ar → Rw			Cl → Ar			Cl → Pr			Cl → Rw					
	S	T	H	S	T	H	S	T	H	S	T	H	S	T	H	S	T	H			
Source model	78.2	45.0	57.1	78.2	67.2	72.3	78.2	73.9	76.0	79.7	49.0	60.7	79.7	59.7	68.3	79.7	62.2	69.9			
SHOT [73]	60.9	55.3	58.0	65.2	77.4	70.8	71.6	80.8	75.9	65.9	68.4	67.1	63.5	76.9	69.6	67.4	75.7	71.3			
GSFDA w/ domain-ID	70.0	54.9	61.5	74.0	77.1	75.5	74.5	79.7	77.0	78.5	67.0	72.7	80.3	76.1	78.1	80.6	78.4	79.5			
GSFDA w/o domain-ID	68.8	54.7	60.9	72.0	75.6	73.8	74.5	78.5	76.4	77.2	66.6	71.5	79.7	74.0	76.7	78.5	78.4	78.4			
	Pr → Ar			Pr → Cl			Pr → Rw			Rw → Ar			Rw → Cl			Rw → Pr			**Avg.**		
	S	T	H	S	T	H	S	T	H	S	T	H	S	T	H	S	T	H	**S**	**T**	**H**
Source model	92.3	52.0	66.5	92.3	40.3	56.1	92.3	73.0	81.5	85.4	64.7	73.6	85.4	45.8	59.6	85.4	77.5	81.3	**83.9**	59.2	68.6
SHOT [73]	78.9	65.4	71.5	74.2	54.2	62.6	84.9	80.5	82.6	79.7	71.7	75.5	71.0	59.0	64.4	79.2	84.6	81.8	71.9	**70.8**	70.9
GSFDA w/ domain-ID	89.8	65.7	75.9	89.3	53.8	67.1	91.6	81.9	86.5	85.9	71.5	78.0	81.3	60.5	69.4	84.4	83.4	83.9	81.8	**70.8**	**75.5**
GSFDA w/o domain-ID	87.8	65.1	74.8	86.3	53.2	65.8	90.3	81.6	85.7	83.2	72.0	77.2	78.3	60.2	68.1	83.4	82.8	83.1	80.0	70.2	74.4

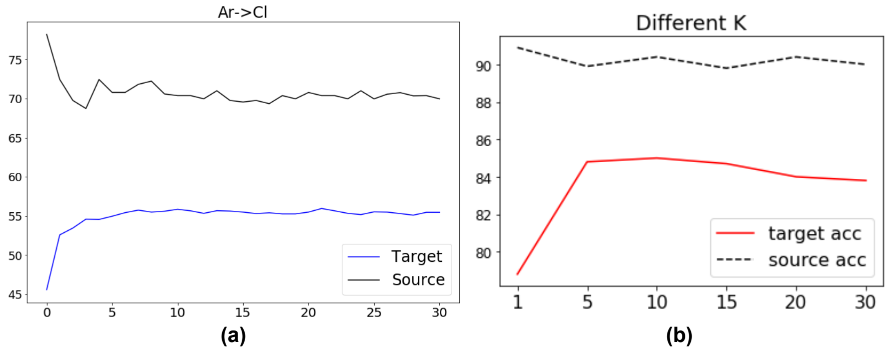

Figure 4.3: (a) Training curves on task Ar→Cl of Office-Home dataset. (b) Ablation study of different K on VisDA.

Table 4.6: (**Left** two) Ablation study on Office-Home and VisDA. The S and T means source and target accuracy. (**Right** two) Ablation on number of stored images per domain to train domain classifier.

Office-Home	S	T
Source model	**83.9**	59.2
GSFDA (w/o SDA)	72.4	70.2
GSFDA (w/ SDA)	81.8	**70.8**

VisDA	S	T
Source model	**99.6**	48.1
GSFDA (w/o SDA)	72.1	74.6
GSFDA (w/ SDA)	90.4	**85.0**

OH /*s*	S	T
65 (paper)	80.0	70.2
130	80.6	70.3
195	**80.8**	**70.4**

VisDA /*s*	S	T
16	89.0	83.6
32	90.2	84.2
64 (paper)	**90.4**	**84.4**

4.4 Experiments

Datasets. *Office-Home* [141] contains 4 domains (Real, Clipart, Art, Product) with 65 classes and a total of 15,500 images. *VisDA* [102] is a more challenging dataset

Table 4.7: Continual Source-free Domain Adaptation, the model is adapted from source domain (the first domain) to all target domain sequentially. The results on source domain are reported on the test set.

	test			
	Ar	Cl	Pr	Rw
Ar	74.5	42.0	61.3	68.2
Cl	71.4	56.6	61.2	67.9
Pr	70.9	55.7	73.0	71.2
Rw	72.6	55.6	72.7	77.2

	test			
	Cl	Ar	Pr	Rw
Cl	82.2	49.7	60.0	61.2
Ar	80.1	65.4	63.7	66.3
Pr	79.7	63.2	72.9	68.2
Rw	78.6	64.9	72.8	72.4

	test			
	Pr	Ar	Cl	Rw
Pr	92.0	49.7	41.0	71.0
Ar	91.0	63.6	42.7	72.6
Cl	89.2	61.8	53.1	70.4
Rw	88.6	63.1	51.5	76.5

	test			
	Rw	Ar	Cl	Pr
Rw	86.0	63.0	45.7	77.6
Ar	85.7	72.4	49.8	77.4
Cl	80.7	68.9	59.1	73.4
Pr	84.2	69.1	57.4	80.5

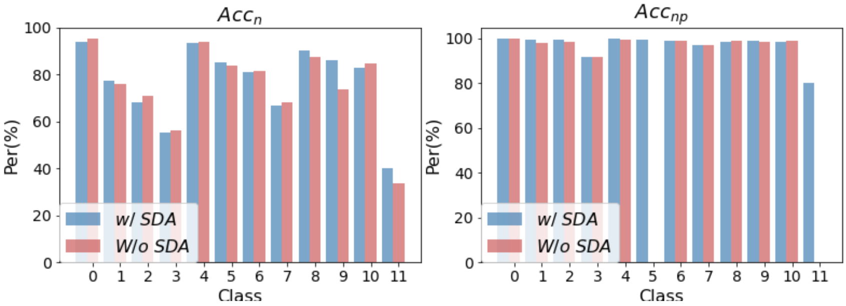

Figure 4.4: Ablation study of SDA on VisDA, which has 12 classes. Acc_n means the percentage of target features which share the same **predicted** label with its 3 nearest neighbors, and Acc_{np} means the percentage among above features which have the **correct** shared predicted class.

with 12 classes. Its source domain contains 152k synthetic images while the target domain has 55k real object images.

Evaluation. We mainly compare with existing methods under two different settings, one is the normal DA and SFDA setting where target performance is the only focus. Another is our proposed G-SFDA setting, where the adapted model is expected to have good performance on both source and target domains after source-free domain adaptation. In this setting, we compute the harmonic mean between source and target accuracy: $H = \frac{2*Acc_S*Acc_T}{Acc_S+Acc_T}$, and Acc_S and Acc_T are respectively the accuracy on source and target test data. For SFDA, we use all source data for model pretraining. And for G-SFDA we only use part (80% for Office-Home and 90% for VisDA), the remaining source data is used for evaluating source performance. We provide results under both the domain aware and domain agnostic setting (where we estimate the domain-ID with the domain classifier). Finally, we report results for continual source-free domain adaptation.

Model details. We adopt the backbone of ResNet-50 [41] for Office-Home and ResNet-101 for VisDA along with an extra fully connected (fc) layer as feature extractor, and a fc layer as classifier head. We adopt SGD with momentum 0.9 and batch size of 64 on all datasets. The learning rate for Office-Home is set to 1e-3 for all layers, except for the last two newly added fc layers, where we apply 1e-2. Learning rates are set 10 times smaller for VisDA. *On the source domain, we train the whole network with all domain attentions from SDA, while for target adaptation, we only train the BN layers and last layer in feature extractor, as well as the classifier.* We train 30 epochs on the target domain for Office-Home while 15 epochs for VisDA. For the number of nearest neighbors (K) in Eq. 4.1, we use 2 for Office-Home, since VisDA is much larger we set K to 10. All results are the average between three runs with random seeds. For training the domain classifier, we store one image per class for Office-Home (total 130 images for 65 classes, 2 domains), and randomly sample 64 images per domain for VisDA (total 128 images for 12 classes, 2 domains). The domain classifier only contains 2 fc layers.

As for SDA, there are two ways to produce attention masks: we can either manually generate those random binary attentions (75% channels each as in the paper), or deploy an embedding layer to automatically produce domain attention. When using embedding layer which takes input the domain id, the domain attention comes from:

$$\mathscr{A}_{\mathrm{i}\in[s,t]} = Sigmoid(100 \cdot e_i) \tag{4.7}$$

where e_i is the output of the embedding layer, and the constant 100 is to ensure a near-binary output, but still differentiable. When training on the source domain, we use all domain attention masks (both for source and target) as mentioned in the paper, *i.e.*, updating the embedding layer with all domain IDs. And we fix the embedding layer during adaptation, thus the domain attentions are fixed (while the embedding layer can also be updated). In the experiments, we find that using the embedding layer for producing SDA obtains similar target performance than manually generating them randomly and use this in our implementation.

4.4.1 Comparing with State-of-the-art

Target-oriented Domain Adaptation. We first evaluate the target performance of our method compared with existing DA and SFDA methods. The results on the VisDA and Office-Home dataset are shown in Tab. 4.1-4.2, our results are using target attention $\mathscr{A}_t$. In these tables, the top part (denoted by × in the *source-free* column) shows results for the normal setting with access to source data during adaptation. The bottom one (denoted by $\surd$ in the *source-free* column) shows results for the source-free setting. Our method achieves state-of-the-art performance on VisDA surpassing SHOT by a large margin (2.5%). The reported results clearly demonstrate the efficiency of the proposed method for source-free domain adaptation. Interestingly, like already observed in the SHOT paper, source-free methods outperform methods that have access to source data during adaptation. Our method is on par with existing DA methods on Office-Home, where our method gets the same results as the DA method SRDC [132] and is a little inferior to the SFDA method SHOT (0.5% lower than SHOT).In addition, we show the results of DANCE [111] with and without source data in Tab. 4.1 which are almost the same. Since both of DANCE and our method are using neighborhood information for adaptation, these results may imply that source data are not necessity when efficiently exploiting the target feature structure.

Generalized Source-free Domain Adaptation. Here we evaluate our method under the G-SFDA setting. Since we leave out part of the source data for evaluation, we need to reproduce current SFDA methods. 3C-GAN [69] did not release code, we therefore only compare with the source-free method SHOT [73] reproduced by ourselves based on the author's code. We also report the results under the GSFDA setting in Tab. 4.3, where 20% of source data is for evaluation on source data. The results from SHOT are reproduced by ourselves based on their code, since the original SHOT uses 90% of source data for pretraining and does not report the results on source domain. As shown in Tab. 4.4-4.5, first our method (w/ domain-ID) obtains a significantly higher H value improving SHOT by 8.8% on Office-Home and 4.6% on VisDA. The gain is mainly due to superior results on the source dataset, since SHOT suffers from forgetting. Compared with the source model, our method still has a drop of 2.1% and 9.2% lower on Office-Home and VisDA, implying there is still space to explore further techniques to reduce forgetting. We also report the results for domain agnostic evaluation, where we use the domain classifier to estimate domain-ID. As shown in the last row of Tab. 4.4 and Tab. 4.5, with the estimated domain-ID, our methods can get similar results compared with the domain aware method, and still report superior H values compared to SHOT. Note there is still source performance degradation, since we only deploy one SDA module before the classifier. The forgetting is caused in the layers inside the feature extractor. One factor is the statistics in the BN layers which will be replaced by the target statistics after adaptation. If we would adapt the BN parameters back to the source domain (by simply doing a forward pass to update BN statistics before evaluation), we found that this leads to a performance gain (0.7% and 1.6% on Office-Home and VisDA respectively) on the source domain.

4.4.2 Analysis and further experiments

Training curves. As shown in Fig. 4.3(a), with SDA the source performance during the whole adaptation stage is quite smooth, which proves the efficiency of SDA.

Number of nearest neighbors K. In Fig. 4.3(b), we show the results with different $K \in \{1, 5, 10, 15, 20, 30\}$ in Eq. 4.1 on VisDA. Our method is quite robust to the choice of K, only K is 1 results in lower results. We conjecture that only using a single nearest neighbor in Eq.4.1 maybe noisy if the feature locates in dense regions.

Ablation study of SDA. We show the results of removing the SDA in the left of Tab. 4.6. As expected removing SDA leads to a large drop in source performance. Unexpected is that removing SDA also deteriorates target performance: a lot on VisDA (10.4$\downarrow$), and a little for Office-Home (0.6$\downarrow$). To further investigate it, we check how well LSC works with and without SDA on VisDA in Fig. 4.4; here Acc_n means the percentage of target features which share the same predicted label with its 3 nearest neighbors, and among those features Acc_{np} means the percentage having the correct shared predicted label. According to the results, LSC can lead to good local structure (most neighbors share the same prediction), however the prediction maybe wrong if removing SDA, this is especially the case for class 5 and 11 which have totally wrong prediction (Acc_{np} is 0). This may imply keeping source information with SDA is helping target adaptation.

Domain classifier. We report results as a function of the number of stored images for training domain classifier (right of Tab. 4.6). For Office-Home, we ensure at least one image per class. The results show with a small amount of stored images, the learned domain-ID classifier works well.

t-SNE visualization. We visualize the features before and after adaptation, which are already masked by the different domain attentions, the source and target features are expected to cluster independently, just as shown in Fig. 4.5. The source clusters maintain well after adaptation, and the disordered target features turn into more structured after adaptation. We also visualize features in the shared and specific domain channels. As shown in Fig. 4.6, features in the shared domain channels cluster together, but features in the specific domain channels are totally separated across domains.

Continual Source-free Domain Adaptation. We also provide results (domain aware) of continual source-free domain adaptation in Tab. 4.7. The results show that it can work well for all domains. The interesting thing is that adapting to one target domain will improve the performance on not-seen target domain, for example, when adapting the model from source domain *Cl* to the first target domain *Ar*, the unseen target domain *Rw* also gains. The reason is that the information learned currently is also helpful for future target domain. Note for some target domains, the result is lower compared with directly adapting from source to the domain, the reason is that we decrease the learned channels by using more gradient regularization as in Eq. 4.6, implying more capacity is needed for adapting to more domains.

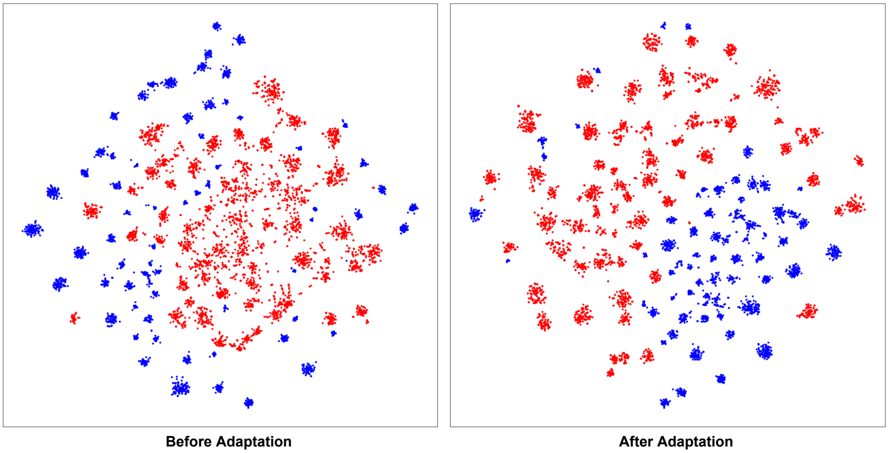

Figure 4.5: t-SNE visualization of features before and after adaptation on task Ar→Pr of Office-Home. The blue are source features while the red are target.

4.5 Conclusion

In this chapter, we propose a new domain adaptation paradigm denoted as Generalized Source-free Domain Adaptation, where the learned model needs to have good performance on both the target and source domains, with only access to the unlabeled

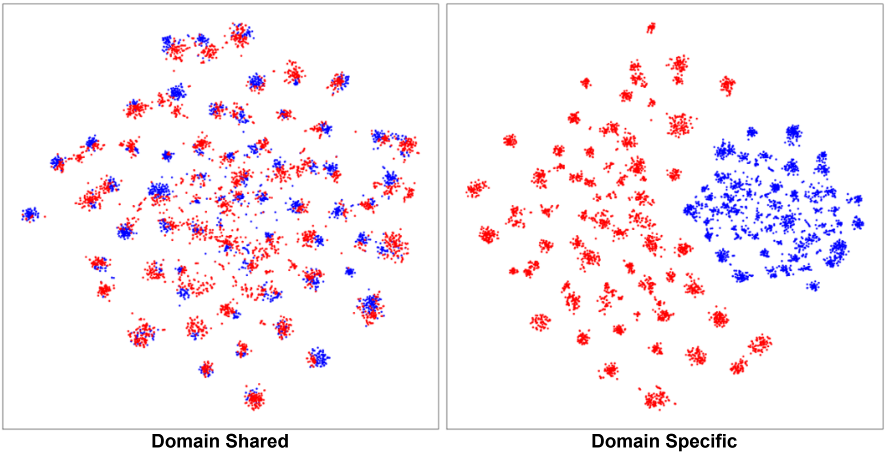

Figure 4.6: t-SNE of features from domain shared and domain specific channels after adaptation (task Ar→Pr on Office-Home). The blue are source features while red for target.

target domain during adaptation. We propose local structure clustering to keep local target cluster information in feature space, successfully adapting the model to the target domain without source domain data. We propose sparse domain attention, which activates different feature channels for different domains, and is also utilized to regularize the gradient during target training to maintain source domain information. Experiment results testify the efficacy of our method.

5 One Ring to Bring Them All: Model Adaptation under Domain and Category Shift[*]

5.1 Introduction

Modern deep learning models excel at close-set recognition tasks across various computer vision application areas. However, there are several inevitable obstacles lying on the path to deploying those methods to the challenging real world environments. As there may be 1) some unseen categories in practical scenarios, or 2) distributional shift between training and testing data. The first problem is usually defined as *open-set recognition* (OSR) [15, 33, 95, 126, 131, 139, 162] where the model should be able to distinguish samples as coming from unseen categories. The second problem is mostly investigated in the *domain generalization* (DG) [40, 108, 123, 140, 145] and *domain adaptation* (DA) community [20, 21, 23, 74, 82, 83, 86, 132, 135, 164]. DG aims to tackle the domain shift problem in the absence of target domains, while DA seeks to transfer knowledge from labeled source domains to unlabeled target domains with training on them with utilizing both labeled source and unlabeled target data, there is distribution/domain shift between source and target domains. In recent years, several works introduce open-set recognition into DG and DA, which are formalized as *open domain generalization* (ODG) [125, 174], *open-set domain adaptation* (OSDA) [9, 27, 28, 54, 79, 99, 116] and *universal domain adaptation* (UNDA) [29, 66, 75, 111, 113, 161], respectively.

Table 5.1: Related setting. $\mathscr{C}_s$ and $\mathscr{C}_t$ denote label set of source and target domain (for evaluation), $\mathscr{P}_s$ and $\mathscr{P}_t$ denote source and target distribution, transductive means model can be trained on target data.

Task	$\mathscr{C}_s = \mathscr{C}_t$	$\mathscr{P}_s = \mathscr{P}_t$	Transductive
Open-set Recognition (OSR)	✗	✓	✗
Domain Generalization (DG)	✓	✗	✗
Open Domain Generalization (ODG)	✗	✗	✗
Domain Adaptation (DA)	✓	✗	✓
Open-partial Domain adaptation (OPDA)	✗	✗	✓

[*] This chapter is a preprint under reviewing, 2022 [159]

The various settings described above are summarized in Tab. 5.1. Usually one method tailored for a specific setting in Tab. 5.1 does not work well under a different setting. Most existing works in *Open-set Recognition* are computationally demanding, either requiring the generation of unknown categories [95] or conducting additional learning [15, 57, 131]. Additionally, those methods are likely to suffer from performance degradation if test data are from different distributions. The recent Cross-Match [174] tackles *Open-set Single Domain Generalization* problem. It proposes to use multiple open class detectors which are put on top of existing single domain generalization methods, and it achieves good results at the expense of introducing multiple open-set detectors and auxiliary unknown sample generation. For *open-partial domain adaptation*, most works are based on an explicitly designed unknown-sample rejection module, which typically requires various hyper-parameters. More importantly, those *OPDA* methods all require access to source data during target adaptation, which is infeasible if having data privacy issues and deployed on devices of low computation capacity.

In this chapter we investigate how to detect open classes efficiently under the domain shift. Thus, a question arises, how to build a model training from only known categories aiming to learn to distinguish samples of unknown categories? Since we have no access to unknown class data, we can only use the known class data to train this classifier. We hypothesize that the closest (most similar) class to any known class can be an unknown class. Given the open-endedness of the unknown class this is a reasonable assumption. This hypothesis allows us to train the classifier, enforcing the most probable class to be the ground truth class, and the runner-up class to be the background class for all source data. This is achieved by introducing an extra category in the classifier which represents the unknown classes, during training on samples of known categories (yielding a $(n + 1)$-way classifier where n is the number of known classes), the classifier is expected to output the largest score for the ground truth class, and the second-largest score for unknown class. This way, the model can learn to reject samples of unknown categories by only training with known classes. The resulting model training on source data can be directly deployed to *open-set single domain generalization*, in other words, it can detect open class efficiently whether there is domain shift or not.

Furthermore, our source model with strong capacity to distinguish unknown categories can be easily adapted to target domain without access to source data under the challenging *source-free open-partial domain adaptation* setting, where both source and target domains have their private classes. We propose to simply use a weighted entropy minimization to achieve the adaptation.

We summarize our contributions as below:

- We propose a simple method called *OneRing*, which excels at recognizing open class (even with domain shift) after source training, thus it can be directly deployed to *open-set single domain generalization* (OS-SDG) and *open-set recognition* (OSR).

- We can easily adapt the source model to target domain by using weighted entropy minimization under *source-free open-partial domain adaptation* setting (SF-OPDA).
- In experiments, we show our method is on par with or outperform current state-of-the-art approaches on several benchmarks for various different tasks, which proves the efficacy and generalization ability of our method. Augmented with a close-set DA approach, our source-free method surpasses current open-partial domain adaptation methods by a significant margin.

5.2 Related Works

Open-set Recognition. *Open-set recognition* (OSR) aims to recognize samples of unknown categories which do not exist in the training set. Several recent methods in OSR do not utilize extra data for training. OpenHybrid [162] introduces a flow-based density estimation module, and ARPL [14, 15] proposes to learn a reciprocal point per category, which is intuitively regarded as the farthest point from the corresponding feature group. More recently [139] shows that actually OSR performance is enhanced when improving the model performance on the training set, for example by using improved data augmentation and other training tricks. In this chapter, we propose a simple model training directly with two cross entropy losses without either auxiliary data or an extra learning process. Our proposed OneRing classifier shares similarity with Proser [169], which aims to assign the second-largest logit to the unknown classes. However, Proser is much more complex compared to ours: it first trains a good $|C_s|$-way close-set classifier and then augment this classifier to $|C_s|+C$-way, and retrain; Further, it needs to synthesize novel samples for training the $|C_s|+C$-way classifier; And they also need to calibrate the output of the dummy classifier over the extra validation set by ensuring 95% of validation data are recognized as known. While in this chapter, we directly train the $|C_s|+1$-way classifier with a simple objective; Another main difference is that they only address open-set recognition, while in our paper we also consider the domain shift, *i.e.*, the challenging source-free open-partial domain adaptation.

Domain Generalization. In *Domain Generalization* (DG), a model is typically trained on multiple labeled source domains. It is expected to have good generalization ability on unseen target domains with which domain shift exists. A typical solution for domain generalization is to learn domain invariant features, which can be achieved by meta learning [24, 64, 65] or additional data generation [170, 171]. In recent years, there are several DG works that only use a single source domain. This setting is known as *single domain generalization* (SDG) [26, 68, 104, 145]. While most of those methods only consider the situation where source and target domains share the same label space, *Open Domain Generalization* (ODG) [125] is recently proposed to deal with the problem where the target domain contains open classes. More recently, CrossMatch [174] introduces an even more challenging setting called *Open-set Single Domain Generalization* (OS-SDG) which only relies on one source and where the target domains contains unknown categories. CrossMatch is built on a complex network model and needs to synthesize samples of unknown categories. It also applies entropy-based unknown class rejection with a manually set threshold. In this chapter, our simple source trained model can be directly deployed to OS-SDG task and gets surprisingly decent results.

Domain Adaptation. Early methods to tackle *domain adaptation* (DA) conduct feature alignment [83, 130, 136] to eliminate the domain shift. DANN [31], CDAN [84] and DIRT-T [124] further resort to adversarial training to learn domain invariable features. Similarly, [62, 88, 115] are based on multiple classifier discrepancy to achieve alignment between domains. Other methods like SRDC [132], CST [80] address domain shift from the perspective of either clustering or improved pseudo labeling. And there are also methods considering category shift source and target domains. They can be grouped into *partial-set DA* [11, 12, 78], *open-set DA* [9, 79, 100, 116] and *universal DA* [29, 66, 111, 113, 161] depending on the intersection degree of source and target label space. OVANet [113] is a universal DA method. It trains extra n binary classifiers with hard negative classifier sampling to reject unknown samples, OVANet needs to check the normal classifier head and the corresponding binary classifier for the final prediction. While in this chapter, we simply train a $n+1$-way classifier with normal cross entropy, and the final prediction is directly provided by the classifier.

Source-free Domain Adaptation. Recently, several works address *source-free domain adaptation* (SFDA), where a source pretrained model is adapted to target without source data. SHOT [73] proposes to use mutual information maximization along with pseudo labeling. BAIT [156] adapts MCD [115] to source-free setting. 3C-GAN [69] resorts to fake target-style images generation. HCL [45] conducts Instance Discrimination [148] over different historical models to cluster features, with the companion of pseudo labeling. A^2Net [149] learns extra classifier specifically for the target domain and introduce a category-wise matching module for feature clustering. G-SFDA [157] and NRC [155] are all based on neighborhood clustering through local prediction consistency. AaD [158] further treats SFDA as a typical unsupervised clustering problem and proposes to optimize an upperbound of a clustering objective. Beyond close-set

DA, FS [59] and USFDA [58], which are for *source-free open-set and open-partial DA* respectively. However, they both synthesize extra training samples of unknown categories, which help to detect the open classes. OSHT [28] tackles source-free open-set DA, which adopts pseudo labeling for adaptation and entropy-based metric to reject open classes. UMAD [75] is for source-free universal DA, it proposes an informative consistency score to detect open class, then adopts mutual information for source-free adaptation.. In this chapter, we show that our source pretrained model can be adapted to the target domain easily by simply minimizing entropy to achieve *source-free open-partial DA*.

5.3 Method

5.3.1 Preliminary

In this chapter, we divide data samples into two groups/domains: the labeled source domain with N_s samples as $\mathscr{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ on which the model will be first trained, and the unlabeled target domain with N_t samples as $\mathscr{D}_t = \{x_i^t\}_{i=1}^{N_t}$. $\mathscr{D}_t$ is used for evaluation. We denote $\mathscr{C}_s$ and $\mathscr{C}_t$ as the label set of the source and target domain, and $\mathscr{P}_s$ and $\mathscr{P}_t$ as the distribution of source and target data respectively. In this chapter, we consider three different tasks that vary in the relation between source and target domain data: 1) ***Open-set Recognition***[†] **(OSR)** where the model is only trained on the source domain, and directly tested on the target domain which contains some unknown categories but without domain shift ($\mathscr{C}_s \subset \mathscr{C}_t, \mathscr{P}_s = \mathscr{P}_t$, *inductive*); 2) ***Open-set Single Domain Generalization*** **[174] (OS-SDG)** which is similarly to OSR trained on a single source domain, however here there exists a domain shift between source and target domains ($\mathscr{C}_s \subset \mathscr{C}_t, \mathscr{P}_s \neq \mathscr{P}_t$, *inductive*); 3) ***Source-free open-partial domain adaptation*** **(SF-OPDA)** which is similar to OS-SDG, here the source model has to adapt to the target domain without access to any source data and both domains have private categories ($\mathscr{C}_s \cap \mathscr{C}_t \neq \emptyset / \mathscr{C}_s / \mathscr{C}_t, \mathscr{P}_s \neq \mathscr{P}_t$, *transductive*). For these settings, we use the same network model containing two parts: a feature extractor f and a classifier head g.

5.3.2 Source Training: One Ring to Find Unknown Categories

The first stage is to train a model on the labeled source domain which has $|\mathscr{C}_s|$ categories. We expect the resulting model to have the ability to detect unknown categories which do not exist in the source data. To achieve this, we build a classifier head as a ($|\mathscr{C}_s| + \mathbf{1}$)-way classifier, where the additional dimension aims to distinguish

[†] Results for OSR are in the appendix, only aiming to show the generalization ability of our method.

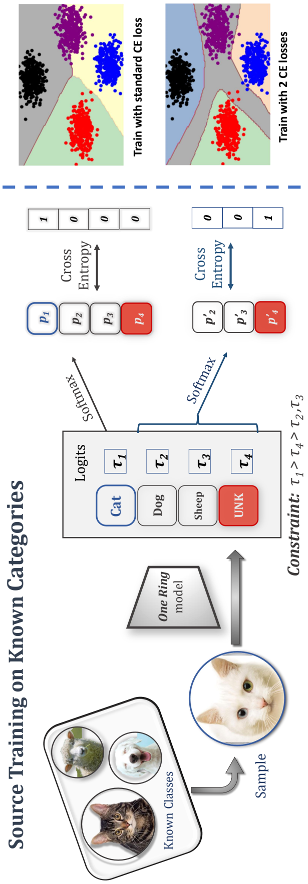

Figure 5.1: (**Left**) Illustration of training *OneRing* model on source data with only known categories. (**Right**) Toy Example, the decision boundaries and prediction regions (*colorized randomly*) after training on 3 known classes with $(3+1)$-way classifier. Purple points are from unknown category.

unknown categories. Then the following problem arises: how to train a $(|\mathscr{C}_s|+1)$-way classifier without any sample from the last/unknown category? Note, if only training with the normal cross entropy (CE) loss on the source data, the model cannot directly give prediction to unknown categories.

As mentioned in Sec. 5.1, we hypothesize that any non-ground-truth category could be regarded as unknown categories. This hypothesis gives us a feasible solution to train a open-set classifier without actually accessing open classes. Specifically, we propose to use a simple variant of cross entropy loss with only samples of known categories to train the $(|\mathscr{C}_s|+1)$-way classifier, which has 2 properties: 1) The largest output logit of the source samples corresponds to the ground truth class and 2) The second-largest output logit of source samples will be the unknown class (($(|\mathscr{C}_s|+1)$-th class in classifier). This way, the model is expected to detect samples of unknown categories even without training on them. The proposed objective to achieve it is formalized as follows:

$$\mathscr{L}_{source} = \mathbb{E}_{x_i \sim \mathscr{D}_s}[\mathscr{L}_{ce}(p(x_i), y_i) + \mathscr{L}_{ce}(\hat{p}(x_i), \hat{y}_i)] \tag{5.1}$$

where $p(x_i) = g(f(x_i)) \in \mathbb{R}^{|C_s|+1}$ is the output vector of the $(|\mathscr{C}_s|+1)$-way classifier, while $\hat{p}(x_i) \in \mathbb{R}^{|C_s|}$ is the output vector removing the dimension corresponding to the ground truth class, and $\hat{y}_i \in \mathbb{R}^{|C_s|}$ is a one-hot label with unknown class as ground truth label. As illustrated in Fig. 5.1 (*right*), if we have a sample x_i belonging to the ***first*** class, the first CE loss in Eq. 5.1 is the typical CE loss on $p(x_i)$ with ground truth label, $\hat{p}(x_i)$ is produced by removing the ***first*** dimension and the second CE loss is applied on $\hat{p}(x_i)$ with unknown (last) category as label.

We adopt a toy example to illustrate it. As shown in upper part of Fig. 5.1 (*right*), we generate isotropic Gaussian blobs with 4 categories, where the last one is treated as the unknown category (in *Purple*) and others as known classes (thus $|\mathscr{C}_s| = 3$). We first train the $(|\mathscr{C}_s|+1)$-way classifier which contains 4 linear layers with the normal cross entropy loss on samples of known categories, and then evaluate it on all classes. Upper part of Fig. 5.1 (*right*) shows that the samples of the unknown category (*Purple*) are misclassified as there are only 3 prediction regions for 3 known categories. As shown in lower part of Fig. 5.1 (*right*) that there are 4 prediction regions (3 known + 1 unknown categories), after training on 2 CE losses the classifier can detect samples of unknown category which is unseen before. We attach a demo video to show the difference between training the $(|\mathscr{C}_s|+1)$-way classifier with only standard CE loss and those 2 CE losses.

An intuitive understanding of the proposed method is that, we can split the $(|\mathscr{C}_s|+1)$-way classification into 2 levels: 1) if we check the prediction $p(x_i)$ we would say x_i has to belong to category y_i; 2) if we check the prediction $\hat{p}(x_i)$ we would say that x_i is impossible to belong to all other categories except the potential unknown categories.

Since in Eq. 5.1 the output score of unknown category (last dimension) will always rule other non-ground-truth categories, we call the last dimension of the classifier head as *OneRing* dimension and our model as *OneRing*. In the experimental section, we will show that our *OneRing* model trained on source data can be directly deployed to open-set recognition and open-set single domain generalization.

5.3.3 Target Adaptation: One Ring to Bind All Categories without the Source

Our source-pretrained *OneRing* model is empowered with the ability to recognition unknown classes in the target domain. We further posit that it can easily be adapted to target domains where domain shift and unknown categories exist. The key part is to rectify the wrong predictions due to the domain shift. We propose to simply use entropy minimization, which is widely used in DA [73, 84, 111, 113, 124], to achieve adaptation with only a slight but indispensable modification:

$$\mathcal{L}_{target} = \frac{bs}{\hat{n}_{k_{all}}} \mathbb{E}_{\bar{y}_i \in \mathcal{C}_s} \mathcal{L}_{ent}(p(x_i)) + \frac{bs}{\hat{n}_{u_{all}}} \mathbb{E}_{\bar{y}_i \in \mathcal{C}_u} \mathcal{L}_{ent}(p(x_i)) \tag{5.2}$$

which is computed in the **mini-batch** (bs denotes batch size), and $\bar{y}_i$ is the predicted label, $\hat{n}_{k_{all}}$ is the number of samples in the *whole dataset* which are predicted as *known* category $\mathcal{C}_s$, $\hat{n}_{u_{all}}$ is the number of those predicted as *unknown* category $\mathcal{C}_u$ also in the *whole dataset*. Here $\frac{bs}{\hat{n}_{k_{all}}} = \frac{N_t}{\hat{n}_{k_{all}}} \times \frac{bs}{N_t}$ (similar for $\frac{bs}{\hat{n}_{u_{all}}}$), where $N_t = \hat{n}_{k_{all}} + \hat{n}_{u_{all}}$ and $\frac{N_t}{\hat{n}_{k_{all}}}$ is the reciprocal of the known/unknown category ratio (a prior information according to the predictions). The reason to deploy these weights is to balance the two entropy terms, and $\frac{bs}{N_t}$ is a scale factor. [‡] With this simple objective, the source model can be adapted to the target domain under domain and category shift efficiently.

Augmented with Attracting-and-Dispersing. Since our *OneRing* method can equip models to efficiently detect unknown classes, it can be used as a baseline to be combined with methods in close-set source-free DA. Here we integrate our method with a simple state-of-the-art SFDA method Attracting-and-Dispersing (AaD) [158], which is introduced in Ch. 3, note AaD can not directly tackle the open-partial domain adaptation setting. AaD has an objective with only 2 dot product terms: $\mathcal{L}_{dis}$ for discriminability and $\mathcal{L}_{div}$ for diversity, more details can be found in AaD paper. The

[‡] Instead of using the predictions over the whole dataset to compute known-unknown ratio, we can also use prediction of current mini-batch for approximation (thus N_t will be replaced by bs, and similar for $\hat{n}_{u_{all}}$ and $\hat{n}_{k_{all}}$), in the experiment we empirically found these two different estimation manners lead to almost the same results.

resulting objective is:

$$\begin{aligned}\mathcal{L}_{target+} = & \frac{bs}{\hat{n}_{k_{all}}} \mathbb{E}_{\bar{y}_i \in \mathscr{C}_s} [\mathcal{L}_{ent}(p(x_i)) + \mathcal{L}_{dis} + \mathcal{L}_{div}] \\ & + \frac{bs}{\hat{n}_{u_{all}}} \mathbb{E}_{\bar{y}_i \in \mathscr{C}_u} [\mathcal{L}_{ent}(p(x_i)) + \mathcal{L}_{dis}]\end{aligned} \tag{5.3}$$

where we do not deploy the diversity term for samples predicted as an unknown class since there is only one single unknown class.

5.4 Experiments

Here we provide quantitative results and analyses related to open-set single domain generalization and source-free open-partial domain adaptation.

5.4.1 Datasets

Open-set Single Domain Generalization. For OS-SDG the model is trained on source data and evaluated on target data containing both known and unknown categories, but here domain shift exists between source and target domains. We use the following benchmarks just as CrossMatch [174]: 1) **Office31** [110] has 31 classes with 3 different domains: amazon (A), dslr (D) and webcam (W). The 10 classes shared by Office-31 and Caltech-256 [38] will be used as source categories. Then the last 11 classes in alphabetical order along with the 10 source categories will be used as target categories. Following CrossMatch, we only adopt A as the source domain, since D and W contain a relatively small amount of samples. 2) **Office-Home** [141] has 4 domains: Artistic (A), Clip Art (C), Product (P), and Real-World (R) with 65 categories. In alphabetic order, the first 15 classes are adopted as source categories. And all classes are used as target categories. 3) **PACS** [63] has 4 domains: Art Paint, Cartoon, Sketch, and Photo. It has 7 categories. Of these, 4 classes (dog, elephant, giraffe, and guitar) will be used as source categories and all classes will be used as target categories. For Office-Home and PACS, the model will be trained on one domain and evaluated on all remaining domains.

Source-free Univeral Domain Adaptation. For SF-OPDA, the model is trained on the source domain first, then adapted to the target domain without access to any source data. Here both the source and target domains have their private categories and the target domain has some unknown categories. We evaluate our method on several benchmarks following the same setting as previous work in OPDA [111, 113, 161]:

1) **Office-31** shares 10 classes with Caltech-256 which will be used as the common categories. Then the next 10 classes in alphabetical order will be source private, and the remaining classes will be target private. 2) **Office-Home** The first 10 classes in alphabetical order are shared between domains, and the next 5 categories will be source private, and the remaining classes are target private. 3) **VisDA** (VisDA-C 2017) [102] The 6 classes out of 12 classes will be the shared categories, and source and target domain both have 3 private classes. 4) **DomainNet** [101] DomainNet is one of the largest domain adaptation benchmarks with around 0.6 million images. Following previous works, we will use 3 domains: Painting (P), Real (R), and Sketch (S). We will use the first 150 classes as shared categories, the next 50 classes are source private and the remaining 145 as target private. The number of source, target and shared categories is described in the title of each Table.

Table 5.2: Accuracy (%) on **Office-31** dataset using ResNet-18. **Open-set Single Domain Generalization** where $|\mathscr{C}_s| = 10$, $|\mathscr{C}_t| = 21$, $|\mathscr{C}_s \cap \mathscr{C}_t| = 10$. All other results are from [174].

Metric	ERM	+CM [174]	ADA	+CM [174]	MEADA	+CM [174]	***OneRing*-S**
Acc	79.8	78.3	80.1	78.6	**80.3**	79.0	67.3
UNK	27.0	37.6	25.2	34.5	25.1	41.1	**77.0**
OS*	85.1	82.4	85.6	83.0	**85.8**	82.8	66.3
H	40.7	51.1	38.7	48.5	38.6	54.7	**71.3**

Table 5.3: Accuracy (%) on **Office-Home** using ResNet-18. **Open-set Single Domain Generalization** where $|\mathscr{C}_s| = 15$, $|\mathscr{C}_t| = 65$, $|\mathscr{C}_s \cap \mathscr{C}_t| = 15$. Other results are copied from [174].

	Artistic		*Clipart*		*Product*		*Real World*		**Average**	
	OS	H	OS	H	OS	H	OS	H	OS	H
ERM [56]	65.0	31.1	64.1	35.8	60.5	36.3	66.6	33.9	64.1	34.3
ERM+CM [174]	65.5	52.9	63.4	50.5	58.0	47.3	67.8	52.6	63.7	50.8
ADA [142]	68.3	32.9	65.1	42.1	60.5	34.7	67.0	34.9	65.2	36.2
ADA+CM [174]	66.3	46.7	62.6	49.3	58.7	47.5	66.8	50.5	63.6	48.5
MEADA [166]	68.3	33.3	65.3	42.1	60.4	35.7	67.0	34.7	65.0	36.4
MEADA+CM [174]	65.9	52.3	62.9	48.9	58.4	45.3	67.1	50.8	63.6	49.6
***OneRing*-S**	63.5	**58.0**	63.2	**59.2**	57.0	**55.3**	63.1	**57.8**	61.7	**57.6**

Table 5.4: Accuracy (%) on **PACS** dataset using ResNet-18. **Open-set Single Domain Generalization** where $|\mathscr{C}_s| = 4$, $|\mathscr{C}_t| = 7$, $|\mathscr{C}_s \cap \mathscr{C}_t| = 4$. Other results are copied from [174].

	Art Paint		*Cartoon*		*Sketch*		*Photo*		**Average**	
	OS	H	OS	H	OS	H	OS	H	OS	H
ERM [56]	62.2	38.9	55.3	41.0	39.2	28.9	38.3	35.7	48.8	36.1
ERM+CM [174]	63.5	44.9	57.6	48.3	38.5	30.4	42.5	41.6	50.5	41.3
ADA [142]	62.5	39.0	56.4	41.6	39.0	26.9	40.3	38.1	49.6	36.4
ADA+CM [174]	64.3	42.4	60.4	51.8	42.5	35.2	43.8	**42.8**	52.8	**43.0**
MEADA [166]	62.4	38.9	56.1	41.3	38.9	26.4	39.9	38.2	49.3	36.2
MEADA+CM [174]	62.6	41.9	60.0	51.4	41.5	35.8	43.5	41.6	51.9	42.7
***OneRing*-S**	54.3	**48.1**	54.5	**58.1**	34.8	**36.5**	32.8	29.4	44.1	**43.0**

5.4.2 Model Details and Evaluation

For all setting, we directly adopt the prediction of our *OneRing* model, without using any extra process for unknown category detection. To ensure fair comparison with previous methods, our method is based on the original code released by OPDA method OVANet [113] (modified for OS-SDG and SF-OPDA).

For OS-SDG, we train our *OneRing* model on source with Eq. 5.1 and directly evaluate on the target. For SF-OPDA, after finishing source training with Eq. 5.1, we will adapt the source pretrained model to target domain without using source data. Only on the very large DomainNet under SF-OPDA setting we found that our method had difficulties converging. Therefore, we applied a two-phase training on the source data. In the first phase, we train with the standard CE loss. Then after convergence, we add the second CE loss for a few epochs. For all experiments under SF-OPDA setting, the *OneRing* classifier is fixed during target adaptation. When augmented with AaD [158], we set the hyperparameter K in $\mathscr{L}_{dis}$ same as AaD, and β in $\mathscr{L}_{div}$ as 1. *We use the predictions in current mini-batch to estimate the known/unknown ratio in Eq. 5.2, since it does not require access to the whole dataset, and we will show it achieves similar results as using the one over whole dataset.*

For OS-SDG, we will report average per-class accuracy over known categories (OS^*), unknown class accuracy (*UNK*) and harmonic mean (H) between OS^* and *UNK*. For SF-OPDA, we will mainly report the harmonic mean, as all previous methods did, and also the average per-class accuracy over all categories (OS) on Office-31. Note for OS-SDG and SF-OPDA, the model is expected to have high performance on both known and unknown accuracy, which should result in a high harmonic mean (H). As pointed out by ROS [9], OS is not a reasonable evaluation metric and can be quite high

Table 5.5: Accuracy (%) on **Office-31** and **VisDA** dataset using ResNet-50. **open-partial domain adaptation** where for *Office-31*: $|\mathscr{C}_s| = 20$, $|\mathscr{C}_t| = 21$, $|\mathscr{C}_s \cap \mathscr{C}_t| = 10$; and for *VisDA*: $|\mathscr{C}_s| = 9$, $|\mathscr{C}_t| = 9$, $|\mathscr{C}_s \cap \mathscr{C}_t| = 6$. The second highest H score is underlined. **SF** indicates whether source-free.

Office-31	**SF**	A2W		D2W		W2D		A2D		D2A		W2A		**Avg**		**VisDA**
		OS	H	OS	H	OS	H	OS	H	OS	H	OS	H	OS	H	H
OSBP [116]	✗	66.1	50.2	73.6	55.5	85.6	57.2	72.9	51.1	47.4	49.8	60.5	50.2	67.7	52.3	27.3
UAN [161]	✗	85.6	58.6	94.8	70.6	**98.0**	71.4	86.5	59.7	85.5	60.1	85.1	60.3	89.2	63.5	30.5
ROS [9]	✗	-	71.3	-	94.6	-	95.3	-	71.4	-	81.0	-	81.2	-	82.1	-
CMU [29]	✗	86.7	67.3	**96.7**	79.3	**98.0**	80.4	89.1	68.1	88.4	71.4	88.6	72.2	91.1	73.1	34.6
DCC [66]	✗	**91.7**	78.5	94.5	79.3	96.2	88.6	**93.7**	**88.5**	**90.4**	70.2	**92.0**	75.9	**93.1**	80.2	43.0
DANCE [111]	✗	-	71.5	-	91.4	-	87.9	-	78.6	-	79.9	-	72.2	-	80.3	4.4
OVANet [113]	✗	-	79.4	-	**95.4**	-	94.3	-	85.8	-	80.1	-	84.0	-	86.5	53.1
USFDA [58]	✓	-	79.8	-	90.6	-	81.2	-	85.5	-	83.2	-	88.7	-	84.8	-
87.4	-	**90.4**	-	87.0	58.3											
***OneRing*-S**		69.0	67.9	92.5	90.6	96.5	89.4	81.9	74.9	64.8	74.8	69.9	78.8	79.1	79.4	35.2
OneRing	✓	78.8	83.8	94.7	95.2	97.5	96.0	86.6	85.7	82.0	85.8	81.0	84.7	86.8	88.5	60.7
***OneRing*+**	✓	85.3	**85.4**	94.0	94.2	97.0	93.6	88.4	86.1	88.9	**90.7**	87.3	84.0	90.2	**89.0**	**66.1**

even when *UNK* is 0, since $OS = \frac{|\mathscr{C}_s|}{|\mathscr{C}_s|+1} \times OS^* + \frac{1}{|\mathscr{C}_s|+1} \times UNK$. In the following tables,

Table 5.6: **H-score** (%) on **Office-Home** dataset using ResNet-50 as backbone. **open-partial domain adaptation** where $|\mathscr{C}_s| = 15$, $|\mathscr{C}_t| = 60$, $|\mathscr{C}_s \cap \mathscr{C}_t| = 10$. The second highest H score is underlined. **SF** indicates whether source-free.

	SF	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	**Avg**
OSBP [116]	✗	39.6	45.1	46.2	45.7	45.2	46.8	45.3	40.5	45.8	45.1	41.6	46.9	44.5
UAN [161]	✗	51.6	51.7	54.3	61.7	57.6	61.9	50.4	47.6	61.5	62.9	52.6	65.2	56.6
CMU [29]	✗	56.0	56.9	59.1	66.9	64.2	67.8	54.7	51.0	66.3	68.2	57.8	69.7	61.6
DCC [66]	✗	58.0	54.1	58.0	**74.6**	70.6	77.5	64.3	**73.6**	74.9	81.0	**75.1**	80.4	70.2
DANCE [111]	✗	-	-	-	-	-	-	-	-	-	-	-	-	49.2
OVANet [113]	✗	62.8	75.6	78.6	70.7	68.8	75.0	71.3	58.6	80.5	76.1	64.1	78.9	71.8
***OneRing*-S**		55.7	72.4	79.6	64.6	65.3	74.6	65.9	51.5	77.9	72.1	57.8	75.0	67.7
OneRing	✓	63.3	72.4	81.0	68.8	67.2	74.6	73.3	60.8	80.9	78.1	63.9	76.7	71.8
***OneRing*+**	✓	**69.5**	**81.4**	**87.9**	73.2	**77.9**	**82.4**	**81.5**	68.6	**88.1**	**81.1**	70.5	**85.7**	**79.0**

Table 5.7: **H-score** (%) on **DomainNet** using ResNet-50 as backbone. **open-partial domain adaptation** where $|\mathscr{C}_s| = 200$, $|\mathscr{C}_t| = 295$, $|\mathscr{C}_s \cap \mathscr{C}_t| = 150$. The second highest H score is underlined. **SF** indicates whether source-free.

Method	**SF**	P2R	R2P	P2S	S2P	R2S	S2R	**Avg**
OSBP [116]	✗	33.6	33.0	30.6	30.5	30.6	33.7	32.0
DANCE [111]	✗	21.0	47.3	37.0	27.7	**46.7**	21.0	33.5
UAN [161]	✗	41.9	43.6	39.1	38.9	38.7	43.7	41.0
CMU [29]	✗	50.8	**52.2**	45.1	44.8	45.6	51.0	48.3
DCC [66]	✗	56.9	50.3	43.7	44.9	43.3	56.2	49.2
OVANet [113]	✗	56.0	51.7	**47.1**	47.4	44.9	57.2	50.7
***OneRing*-S**		**59.1**	42.9	43.8	35.5	39.5	52.9	45.6
OneRing	✓	57.9	52.0	46.5	**49.6**	44.1	**57.8**	**51.3**

we will denote our model trained with only source data as ***OneRing*-S**, model after target adaptation as ***OneRing***, and model augmented with AaD after target adaptation as ***One Ring*+**.

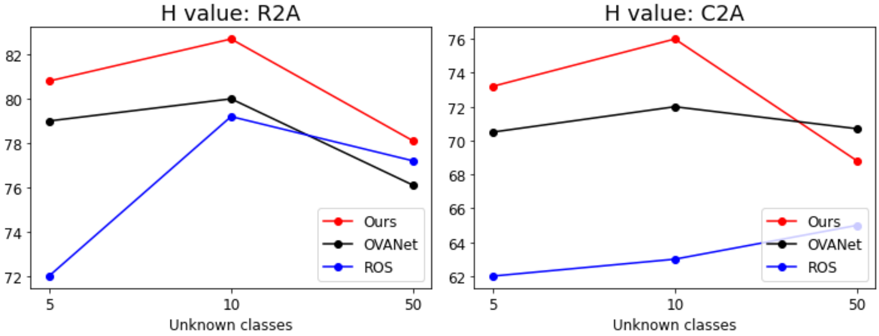

Figure 5.2: **H** value of open-partial domain adaptation on Office-Home. We vary the number of unknown classes as shown in the x axis. Here 'ours' denotes OneRing without being augmented with AaD, OVANet and ROS demand source data.

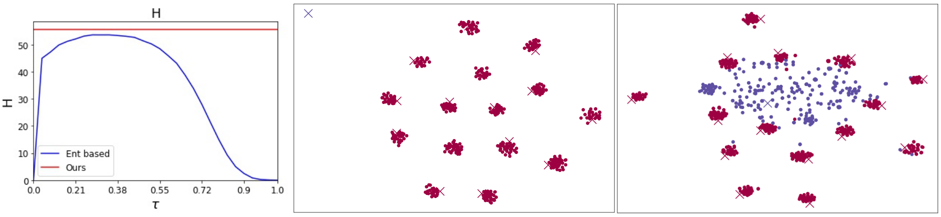

Figure 5.3: (**Left**) H value of our source model and entropy based rejection on A2C of Office-Home. t-SNE visualization of features with either only source known categories (**Middle**) or also with 10 source extra unknown categories (**Right**) from source model on *Artistic* of Office-Home, where the cross is the class prototype. The red denotes known classes while other for unknown class.

Table 5.8: Accuracy (%) on open-partial DA. **Results are from one random run.**

Office-Home																		
	Ar → Cl			Ar → Pr			Ar → Rw			Cl→ Ar			Cl → Pr			Cl → Rw		
	OS*	UNK	**H**	OS*	UNK	**H**	OS*	UNK	**H**	OS*	UNK	**H**	OS*	UNK	**H**	OS*	UNK	**H**
OneRing-S	42.9	79.3	55.7	75.7	69.5	72.4	91.7	70.3	79.6	52.9	82.1	64.4	60.0	71.7	65.3	75.2	74.0	74.6
OneRing	54.1	73.9	62.5	78.5	69.8	73.9	93.3	72.5	81.6	65.9	73.0	69.3	67.5	66.1	66.8	80.0	69.0	74.1
OneRing+	58.5	84.4	69.1	78.3	84.8	81.4	92.6	84.4	88.3	62.7	88.2	73.3	72.1	86.3	78.6	80.4	86.0	83.1

	Pr → Ar			Pr → Cl			Pr → Rw			Rw→ Ar			Rw → Cl			Rw → Pr			**Avg**		
	OS*	UNK	**H**	OS*	UNK	**H**	OS*	UNK	**H**	OS*	UNK	**H**	OS*	UNK	**H**	OS*	UNK	**H**	OS*	UNK	**H**
OneRing-S	55.9	80.2	65.9	38.6	77.1	51.5	86.9	70.5	77.9	70.4	73.9	72.1	46.6	76.0	57.8	82.6	68.7	75.0	65.0	74.4	67.7
OneRing	73.1	73.2	73.1	52.8	70.2	60.3	91.6	73.4	81.5	77.9	78.1	78.0	57.7	70.2	63.4	88.2	70.3	78.2	73.4	71.6	71.9
OneRing+	77.4	86.7	82.2	58.3	82.3	68.3	92.2	84.7	88.3	76.5	86.2	81.1	61.5	82.7	70.5	86.9	85.4	86.1	**74.8**	**85.2**	**79.2**

Table 5.9: Ablation study (R2C of Office-Home) on the proposed weight in the weighted entropy minimization. Results of OVANet are from our running based on their official code.

R2C	**OS***	**UNK**	**OS**	**H**
OVANet [113]	55.1	70.0	56.5	61.7
OneRing w/o weight in Eq.2	19.2	97.8	26.3	32.1
OneRing	57.8	71.6	59.1	63.9
OneRing+	61.5	82.7	63.4	70.5

Table 5.10: **H-score** (%) on **Office-Home** dataset using ResNet-50 as backbone. **Open-partial domain adaptation** where $|\mathscr{C}_s| = 15$, $|\mathscr{C}_t| = 60$, $|\mathscr{C}_s \cap \mathscr{C}_t| = 10$. The second highest H score is underlined. **SF** indicates whether source-free. * indicates using predictions over the *whole dataset* instead of mini-batch in Eq. 5.2.

	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	**Avg**
OVANet [113]	62.8	75.6	78.6	70.7	68.8	75.0	71.3	58.6	80.5	76.1	64.1	78.9	71.8
***OneRing*-S**	55.7	72.4	79.6	64.6	65.3	74.6	65.9	51.5	77.9	72.1	57.8	75.0	67.7
OneRing	63.3	72.4	81.0	68.8	67.2	74.6	73.3	60.8	80.9	78.1	63.9	76.7	71.8
OneRing*	60.9	72.1	80.9	67.7	66.0	73.7	73.1	60.4	81.4	77.7	63.4	78.2	71.3
***OneRing*+**	69.5	81.4	87.9	73.2	77.9	82.4	81.5	68.6	88.1	81.1	70.5	85.7	79.0
***OneRing**+**	70.1	82.5	88.9	75.1	80.1	83.0	82.5	64.6	89.3	81.0	66.4	86.0	**79.1**

Table 5.11: Open-partial DA on VisDA, results of OVANet are from our running based on their code.

VisDA	**source-free**	**OS***	**UNK**	**OS**	**H**
OVANet [113]	✗	60.5	46.4	58.5	52.5
OneRing-S		25.7	55.9	30.0	35.2
OneRing	✓	57.2	64.6	58.3	60.7
OneRing+	✓	65.5	66.8	65.7	66.1

5.4.3 Quantitative results

Open-set Single Domain Generalization. In Tab. 5.2-5.4, we show the results of our source model *OneRing*-S on Office-31, Office-Home and PACS. ERM [56], ADA [142] and MEADA [166] are methods originally designed for typical domain

Table 5.12: Open-set DA on Office-31 (VGG19), results (H) except ours are from OVANet.

Methods	*source-free*	A2D	A2W	D2A	D2W	W2D	W2A	Avg
OSBP [116]	✗	81.0	77.5	78.2	95.0	91.0	72.9	82.6
ROS [9]	✗	79.0	81.0	78.1	94.4	99.7	74.1	84.4
OVANet [113]	✗	89.5	84.9	89.7	93.7	85.8	88.5	88.7
OneRing	✓	91.0	84.5	90.1	96.0	93.7	90.1	**90.9**

Table 5.13: Closed-set DA on Office-31 (ResNet50), results (accuracy) except ours are from DANCE, and **results of OVANet are from our running based on their official code**.

Methods	*source-free*	A2W	D2W	W2D	A2D	D2A	W2A	Avg
ETN	✗	87.9	99.2	100	88.4	68.7	66.8	85.2
STA	✗	77.1	90.7	98.1	75.5	51.4	48.9	73.6
UAN	✗	86.5	97.0	100	84.5	69.6	68.7	84.4
DANCE	✗	88.6	97.5	100	89.4	69.5	68.2	85.5
OVANet	✗	88.1	97.0	99.1	88.6	68.8	67.0	84.8
OneRing	✓	89.0	97.3	100	89.0	70.1	68.5	**85.7**

generalization, CrossMatch (CM) [174] is plugged into these methods which empower them with the ability to detect unknown classes in the target domain with several complex modules, as well as generating unknown samples. While our *OneRing*-S is elegantly simple, the results show it can better detect open classes under domain shift compared to CM. Note, we have no module specifically for DG in *OneRing*-S. The fact that *OneRing*-S has better performance proves the efficacy of our method.

Source-free open-partial domain adaptation In Tab. 5.5-5.7, we show the results under open-partial DA setting where **SF** column indicates whether source-free. Note that our method does not need source data during target adaptation. As shown in the tables, our source model (*One Ring-S*) already achieves decent H performance. The simple *OneRing* with only entropy minimization already outperforms all other methods on all 4 benchmarks, adding AaD [158] into method as shown in Eq. 5.3 (*OneRing*+) can further improve the results significantly, leading to 0.5%, 5.4% and 7.2% improvement on Office-31, VisDA and Office-Home respectively, and it surpasses the current state-of-the-art OVANet by by 2.5%, 13% and 7.2% on these 3 benchmarks respectively. We also show the detailed results of OS*, UNK and H in Tab. 5.8.

5.4.4 Analysis

Compare One Ring with entropy based unknown rejection. We also show the results with entropy based unknown rejection, where a sample is predicted as unknown if the entropy (maximal normalized) of the prediction (*with normal classifier head*) is higher than a manually set threshold. Fig. 5.3 (*left*) shows the H value of *source pretrained model* on A2C task of Office-Home under open-partial DA setting, where the *x axis* denotes the threshold. Our source model gets better results without any extra effort.

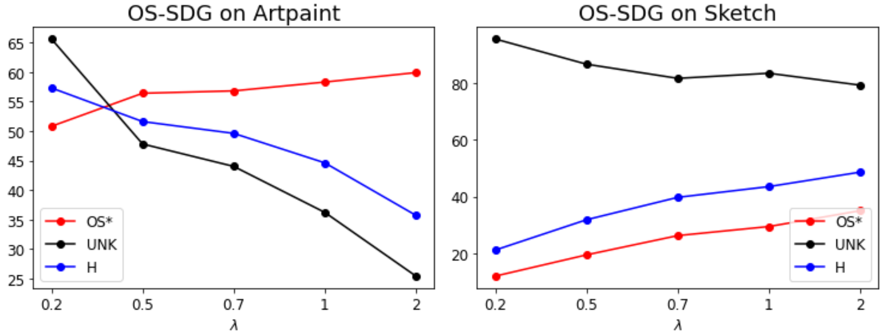

Figure 5.4: Results on PACS (**OS-SDG**), with different weight factors applied to the normal CE loss.

Trade-off between 2 CE losses. In this chapter, we show results where the two CE losses have equal weight, and hence our method does not have any hyperparameter. However, in Eq. 5.1, we can also multiply a weight factor to the standard CE loss as a trade-off. Intuitively, a smaller factor to the standard CE loss gives more weight to unknown-class recognition and vice verse. The results under OS-SDG setting in Fig. 5.4 verify this , where the *x axis* denotes the weight factor multiplied to the standard CE loss. As can be seen, this trade-off can be used to further improve results. However, for the sake of simplicity, and given the already good results, we choose not to optimize this parameter.

Visualization of features and class prototypes. In Fig. 5.3 (*Middle*), we visualize the source features and class prototypes (weights of *OneRing* classifier) from *source model* with t-SNE. The prototype of the unknown category is in the corner with no source features around it. In Fig. 5.3 (*Right*), we further visualize 10 extra unknown classes. It shows that those features of unknown categories will not cluster around any of the known classes, but they are close to the unknown prototype. This implies that the *OneRing* model can efficiently distinguish known and unknown categories.

Importance of weight in entropy minimization. We ablate the weights in entropy minimization in Eq. 5.2. If removing weights, the *OS**, *UNK* and *H* on R2C (Office-Home) will decrease. In Tab. 5.9, we report OS*, UNK, OS and H on VisDA under open-partial DA, we outperform OVANet on the metrics of both OS and H. It shows the deployed weights are important and effective to balance the two terms in Eq. 5.2.

Known/unknown ratio estimation through mini-batch or whole dataset. In Eq. 2, we have two choice to estimate the known/unknown ratio, which will be utilized to balance the 2 entropy terms. In Tab. 5.10, we show that these 2 different manners lead to almost the same results. Though there may exist some imbalance mini-batches which only contain few samples predicted as known or unknown, the results imply that the known/unknown ratio estimated by the mini-batch is enough to achieve decent performance. Note the Office-Home here is not a well balance (amount of samples per category) dataset, and also in the target domain the unknown categories (50) are much more than known (10).

Robustness to amount of unknown categories. In Fig. 5.2, we compare our source-free *OneRing* (without being augmented with AaD) to ROS [9] and OVANet [113] under OPDA setting with different amount of unknown categories from target domain. The results show that our method is robust to the amount of unknown categories.

Results with OS*, UNK and H on open-partial DA. In Tab. 5.11, we report OS*, UNK, OS and H on VisDA under open-partial DA, we outperform OVANet on the metrics of both OS and H.

Results on open-set DA. In Tab. 5.12, we report the results of several open-set or open-partial DA methods under open-set DA. Our method still gets the best performance.

Results on closed-set DA In Tab. 5.13, we report the results of several OPDA methods under closed-set DA, our method is still superior to other methods.

5.5 Conclusion

In this chapter, we first introduce a simple method with the proposed *OneRing* classifier head, it possesses strong ability to detect unknown categories from target data even no matter without or with domain shift after training with two simple cross entropy losses. Then, we further adapt the model to the target domain which contains unknown categories, with only weighted entropy minimization and no access to source data. In the experiment, we show that our method achieves good performance on open-set single domain generalization and source-free open-partial domain adaptation, which proves the effectiveness of our method.

6 Conclusions and Future Work

6.1 Conclusions

In this thesis, we have investigated a new paradigm of domain adaptation, called source-free domain adaptation, which aims to adapt the pretrained source model to a new unlabeled target domain without access to the labeled source data. We also studied situations where, after adaptation, the model is expected to not forget on the source domain. Finally, we investigated the case where the label spaces between different domains are not identical. For these various domain adaptation scenarios, we proposed corresponding solutions in this thesis:

- **Chapter 2: Neighborhood Reciprocal Clustering for Source-free Domain Adaptation.** In this chapter, we have introduced a source-free domain adaptation (SFDA) method by uncovering the intrinsic target data structure. We proposed to achieve adaptation by encouraging label consistency among local target features. We differentiate between nearest neighbors, reciprocal neighbors, and expanded neighborhood. Experimental results verified the importance of considering the local structure of the target features. Finally, our experimental results on both 2D image and 3D point cloud datasets testify to the efficacy of our method.
- **Chapter 3: Attracting and Dispersing for Source-free Domain Adaptation.** In this chapter, we proposed to tackle source-free domain adaptation by encouraging similar features in feature space to have similar predictions while dispersing predictions of dissimilar features in feature space, to simultaneously achieve feature clustering and cluster assignment. We introduced an upper bound to our proposed objective, resulting in two simple terms. Further, we showed that we can unify several popular domain adaptation, source-free domain adaptation, and contrastive learning methods from the perspective of discriminability and diversity. The approach is simple but achieves state-of-the-art performance on several benchmarks, and can also be extended to source-free open-set and partial-set domain adaptation.
- **Chapter 4: Generalized Source-free Domain Adaptation.** In this chapter, we exploited a new domain adaptation paradigm denoted as Generalized Source-free Domain Adaptation, where the learned model needs to have good performance

on both the target and source domains, with only access to the unlabeled target domain during adaptation. We proposed local structure clustering to keep local target cluster information in feature space, successfully adapting the model to the target domain without source domain data. We proposed sparse domain attention, which activates different feature channels for different domains, and is also utilized to regularize the gradient during target training to maintain source domain information. Experimental results testify the efficacy of our method.

- **Chapter 5: OneRing for Model Adaptation under Domain and Category Shift.** In this chapter, we first introduced a simple method with the proposed OneRing classifier head, which possesses a strong ability to detect unknown categories from target data even without or with domain shift after training with two simple cross entropy losses. Then, we further succeeded in adapting the model to the target domain, which contains unknown categories, with only weighted entropy minimization and no access to source data. In the experiment, we showed that our method achieved good performance on open-set single domain generalization and source-free open-partial domain adaptation, which proves the effectiveness of our method.

6.2 Future work

For future work, we are interested in extending the source-free domain adaptation to more general scenarios, for example, starting the source-free domain adaptation from a big foundation model that is already trained with huge amount of multimodality data. As the foundation model such as CLIP [107] is already able to recognize novel categories with the corresponding text prompt. Adapting this strong model to a new environment with only unlabeled data, the resulting model can achieve novel category discovery which not only rejects unseen classes but also distinguishes each novel categories. In this way, it will widen the application scenarios of the foundation model.

We are also interested in expanding the downstream tasks from classification to others, such as semantic/instance/panoptic segmentation, object detection, and tracking, and the data modality could also be point-cloud and RGB-depth data.

And as it is possible to collect a limited amount of labeled data in the real application, we will also consider introducing a few labeled data into the source-free domain adaptation setting, either by manually labeling or utilizing active learning techniques. With efficient use of the provided supervision of only a few labeled data, the model should further improve.

As most existing works in domain adaptation need to access all target data during the adaptation stage, we are interested in conducting source-free domain adaptation in an online manner, for example, fully test-time adaptation where each data point can only be accessed once. Successful test-time adaptation can reduce the computation burden, which is highly important for deployment in edge devices.

Publications

1. **Yang, S**., Wang, Y., Wang, K., Jui, S., van de Weijer, J. (2022). One Ring to Bring Them All: Towards Open-Set Recognition under Domain Shift. ***arXiv preprint arXiv:2206.03600***.

2. Wang, K., Wu, C., Bagdanov, A., Liu, X., **Yang, S**., Jui, S., van de Weijer, J. (2022). Positive Pair Distillation Considered Harmful: Continual Meta Metric Learning for Lifelong Object Re-Identification. (***BMVC 2022***)

3. **Yang, S**., Wang, Y., Wang, K., Jui, S., van de Weijer, J. (2022). Attracting and dispersing: A simple approach for source-free domain adaptation. (***NeurIPS 2022 spotlight***)

4. **Yang, S**., Wang, K., Herranz, L., van de Weijer, J. (2021). On implicit attribute localization for generalized zero-shot learning. (***IEEE Signal Processing Letters, 2021***)

5. **Yang, S**., Wang, Y., van de Weijer, J., Herranz, L., Jui, S. (2021). Exploiting the intrinsic neighborhood structure for source-free domain adaptation. (***NeurIPS 2021***)

6. **Yang, S**., Wang, Y., van de Weijer, J., Herranz, L., Jui, S. (2021). Generalized source-free domain adaptation. (***ICCV 2021***)

7. **Yang, S**., Wang, Y., van de Weijer, J., Herranz, L., Jui, S. (2020). Casting a BAIT for offline and online source-free domain adaptation. ***arXiv preprint arXiv:2010.12427.***

Bibliography

[1] Davide Abati, Jakub Tomczak, Tijmen Blankevoort, Simone Calderara, Rita Cucchiara, and Babak Ehteshami Bejnordi. Conditional channel gated networks for task-aware continual learning. In *CVPR*, pages 3931–3940, 2020.

[2] Idan Achituve, Haggai Maron, and Gal Chechik. Self-supervised learning for domain adaptation on point clouds. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 123–133, 2021.

[3] Sk Miraj Ahmed, Dripta S Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10103–10112, 2021.

[4] Kristen M Altenburger and Johan Ugander. Monophily in social networks introduces similarity among friends-of-friends. *Nature human behaviour*, 2(4):284–290, 2018.

[5] Dina Bashkirova, Dan Hendrycks, Donghyun Kim, Samarth Mishra, Kate Saenko, Kuniaki Saito, Piotr Teterwak, and Ben Usman. Visda-2021 competition universal domain adaptation to improve performance on out-of-distribution data, 2021.

[6] Roger Bermudez Chacon, Mathieu Salzmann, and Pascal Fua. Domain-adaptive multibranch networks. In *ICLR*, 2020.

[7] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

[8] Andreea Bobu, Eric Tzeng, Judy Hoffman, and Trevor Darrell. Adapting to continuously shifting domains. In *ICLR workshop*, 2018.

[9] Silvia Bucci, Mohammad Reza Loghmani, and Tatiana Tommasi. On the effectiveness of image rotation for open set domain adaptation. In *ECCV*, pages 422–438. Springer, 2020.

[10] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. *arXiv preprint arXiv:2102.03526*, 2021.

[11] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *ECCV*, pages 135–150, 2018.

[12] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation. In *CVPR*, pages 2985–2994, 2019.

[13] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *ICCV*, pages 5879–5887, 2017.

[14] Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. Adversarial reciprocal points learning for open set recognition. *IEEE TPAMI*, 2021.

[15] Guangyao Chen, Limeng Qiao, Yemin Shi, Peixi Peng, Jia Li, Tiejun Huang, Shiliang Pu, and Yonghong Tian. Learning open set network with discriminative reciprocal points. In *ECCV*, 2020.

[16] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6321–6330, 2019.

[17] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *ICML*, pages 1081–1090, 2019.

[18] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4794–4802, 2019.

[19] Jinwoo Choi, Gaurav Sharma, Samuel Schulter, and Jia-Bin Huang. Shuffle and attend: Video domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 678–695. Springer, 2020.

[20] Safa Cicek and Stefano Soatto. Unsupervised domain adaptation via regularized conditional alignment. In *ICCV*, pages 1416–1425, 2019.

[21] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. *CVPR*, 2020.

[22] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Fast batch nuclear-norm maximization and minimization for robust domain adaptation. *arXiv preprint arXiv:2107.06154*, 2021.

[23] Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *ICCV*, pages 9944–9953, 2019.

[24] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *NeurIPS*, 32, 2019.

[25] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. *ICCV*, 2021.

[26] Xinjie Fan, Qifei Wang, Junjie Ke, Feng Yang, Boqing Gong, and Mingyuan Zhou. Adversarially adaptive normalization for single domain generalization. In *CVPR*, pages 8208–8217, 2021.

[27] Qianyu Feng, Guoliang Kang, Hehe Fan, and Yi Yang. Attract or distract: Exploit the margin of open set. In *ICCV*, pages 7990–7999, 2019.

[28] Zeyu Feng, Chang Xu, and Dacheng Tao. Open-set hypothesis transfer with semantic consistency. *IEEE TIP*, 30:6473–6484, 2021.

[29] Bo Fu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Learning to detect open classes for universal domain adaptation. In *ECCV*, pages 567–583. Springer, 2020.

[30] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[31] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016.

[32] Huan Gao, Jichang Guo, Guoli Wang, and Qian Zhang. Cross-domain correlation distillation for unsupervised domain adaptation in nighttime semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9913–9923, 2022.

[33] Zongyuan Ge, Sergey Demyanov, and Rahil Garnavi. Generative openmax for multi-class open set classification. In *BMVC*, 2017.

[34] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *ICCV*, pages 5736–5745, 2017.

[35] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 597–613. Springer, 2016.

[36] Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. *NIPS*, 17, 2004.

[37] Ryan Gomes, Andreas Krause, and Pietro Perona. Discriminative clustering by regularized information maximization. In *NIPS*, 2010.

[38] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073. IEEE, 2012.

[39] Xiang Gu, Jian Sun, and Zongben Xu. Spherical space domain adaptation with robust pseudo-label loss. In *CVPR*, pages 9101–9110, 2020.

[40] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *ICLR*, 2021.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[42] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 749–757, 2020.

[43] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *ICML*, pages 1558–1567, 2017.

[44] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *ICML*, pages 2849–2858. PMLR, 2019.

[45] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. *NeurIPS*, 34, 2021.

[46] Zhizhong Huang, Jie Chen, Junping Zhang, and Hongming Shan. Exploring non-contrastive representation learning for deep clustering. *arXiv preprint arXiv:2111.11821*, 2021.

[47] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5001–5009, 2018.

[48] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[49] Mohammed Jabi, Marco Pedersoli, Amar Mitiche, and Ismail Ben Ayed. Deep clustering: On the link between discriminative models and k-means. *TPAMI*, 2019.

[50] Herve Jegou, Hedi Harzallah, and Cordelia Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *CVPR*, pages 1–8. IEEE, 2007.

[51] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, pages 9865–9874, 2019.

[52] Xiang Jiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei. Implicit class-conditioned domain alignment for unsupervised domain adaptation. *arXiv preprint arXiv:2006.04996*, 2020.

[53] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. *ECCV*, 2020.

[54] Taotao Jing, Hongfu Liu, and Zhengming Ding. Towards novel target discovery through open-set domain adaptation. In *ICCV*, pages 9322–9331, 2021.

[55] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[56] Vladimir Koltchinskii. *Oracle inequalities in empirical risk minimization and sparse recovery problems: École D'Été de Probabilités de Saint-Flour XXXVIII-2008*, volume 2033. Springer Science & Business Media, 2011.

[57] Shu Kong and Deva Ramanan. Opengan: Open-set recognition via open data generation. *ICCV*, 2021.

[58] Jogendra Nath Kundu, Naveen Venkat, and R Venkatesh Babu. Universal source-free domain adaptation. *CVPR*, 2020.

[59] Jogendra Nath Kundu, Naveen Venkat, Ambareesh Revanur, R Venkatesh Babu, et al. Towards inheritable models for open-set domain adaptation. In *CVPR*, pages 12376–12385, 2020.

[60] Jogendra Nath Kundu, Rahul Mysore Venkatesh, Naveen Venkat, Ambareesh Revanur, and R Venkatesh Babu. Class-incremental domain adaptation. *ECCV*, 2020.

[61] Trung Le, Tuan Nguyen, Nhat Ho, Hung Bui, and Dinh Phung. Lamda: Label matching deep domain adaptation. In *ICML*, pages 6043–6054. PMLR, 2021.

[62] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, pages 10285–10295, 2019.

[63] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, pages 5542–5550, 2017.

[64] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018.

[65] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *ICCV*, pages 1446–1455, 2019.

[66] Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. In *CVPR*, pages 9757–9766, 2021.

[67] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021.

[68] Lei Li, Ke Gao, Juan Cao, Ziyao Huang, Yepeng Weng, Xiaoyue Mi, Zhengze Yu, Xiaoya Li, and Boyang Xia. Progressive domain expansion network for single domain generalization. In *CVPR*, pages 224–233, 2021.

[69] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, pages 9641–9650, 2020.

[70] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *AAAI*, 2021.

[71] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6936–6945, 2019.

[72] Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPAMI*, 40(12):2935–2947, 2017.

[73] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. *ICML*, 2020.

[74] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *CVPR*, pages 16632–16642, 2021.

[75] Jian Liang, Dapeng Hu, Jiashi Feng, and Ran He. Umad: Universal model adaptation under domain and category shift. *arXiv preprint arXiv:2112.08553*, 2021.

[76] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *arXiv preprint arXiv:2012.07297*, 2020.

[77] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[78] Jian Liang, Yunbo Wang, Dapeng Hu, Ran He, and Jiashi Feng. A balanced and uncertainty-aware approach for partial domain adaptation. In *ECCV*, pages 123–140. Springer, 2020.

[79] Hong Liu, Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Qiang Yang. Separate to adapt: Open set domain adaptation via progressive separation. In *CVPR*, pages 2927–2936, 2019.

[80] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. In *NeurIPS*, 2021.

[81] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *CVPR*, pages 1215–1224, 2021.

[82] Mingsheng Long, Yue Cao, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Transferable representation learning with deep adaptation networks. *TPAMI*, 41(12):3071–3085, 2018.

[83] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *ICML*, 2015.

[84] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NIPS*, pages 1647–1657, 2018.

[85] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *ICCV*, pages 2200–2207, 2013.

[86] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, pages 136–144, 2016.

[87] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NIPS*, pages 6470–6479, 2017.

[88] Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi-Zhe Song, and Tao Xiang. Stochastic classifiers for unsupervised domain adaptation. In *CVPR*, pages 9111–9120, 2020.

[89] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the ECCV (ECCV)*, pages 67–82, 2018.

[90] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, pages 7765–7773, 2018.

[91] Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. Adagraph: Unifying predictive and continuous domain adaptation through graphs. In *CVPR*, pages 6568–6577, 2019.

[92] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo. Autodial: Automatic domain alignment layers. In *Proceedings of the IEEE international conference on computer vision*, pages 5067–5075, 2017.

[93] Marc Masana, Tinne Tuytelaars, and Joost van de Weijer. Ternary feature masks: zero-forgetting for task-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3570–3579, 2021.

[94] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *CVPR*, pages 1094–1103, 2021.

[95] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *ECCV*, 2018.

[96] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[97] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, pages 1717–1724, 2014.

[98] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2009.

[99] Yingwei Pan, Ting Yao, Yehao Li, Chong-Wah Ngo, and Tao Mei. Exploring category-agnostic clusters for open-set domain adaptation. In *CVPR*, pages 13867–13875, 2020.

[100] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *ICCV*, pages 754–763, 2017.

[101] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, pages 1406–1415, 2019.

[102] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

[103] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017.

[104] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *CVPR*, pages 12556–12565, 2020.

[105] Can Qin, Haoxuan You, Lichen Wang, C-C Jay Kuo, and Yun Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. *Advances in Neural Information Processing Systems*, 32:7192–7203, 2019.

[106] Danfeng Qin, Stephan Gammeter, Lukas Bossard, Till Quack, and Luc Van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *CVPR 2011*, pages 777–784. IEEE, 2011.

[107] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[108] Alexander Robey, George J Pappas, and Hamed Hassani. Model-based domain generalization. *NeurIPS*, 34:20210–20229, 2021.

[109] Simone Romano, James Bailey, Vinh Nguyen, and Karin Verspoor. Standardized mutual information for clustering comparisons: one step further in adjustment for chance. In *ICML*, pages 1143–1151, 2014.

[110] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226. Springer, 2010.

[111] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. *NeurIPS*, 33, 2020.

[112] Kuniaki Saito, Donghyun Kim, Piotr Teterwak, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Tune it the right way: Unsupervised validation of domain adaptation via soft neighborhood density. In *ICCV*, pages 9184–9193, 2021.

[113] Kuniaki Saito and Kate Saenko. Ovanet: One-vs-all network for universal domain adaptation. In *ICCV*, pages 9000–9009, 2021.

[114] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. *ICLR*, 2018.

[115] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pages 3723–3732, 2018.

[116] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *ECCV*, pages 153–168, 2018.

[117] Tim Salimans and Diederik P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *arXiv preprint arXiv:1602.07868*, 2016.

[118] Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *CVPR*, pages 8934–8943, 2019.

[119] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, pages 4548–4557. PMLR, 2018.

[120] Yuefan Shen, Yanchao Yang, Mi Yan, He Wang, Youyi Zheng, and Leonidas J Guibas. Domain adaptation on point clouds via geometry-aware implicits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7223–7232, 2022.

[121] Yuming Shen, Ziyi Shen, Menghan Wang, Jie Qin, Philip HS Torr, and Ling Shao. You never cluster alone. In *NeurIPS*, 2021.

[122] Yuan Shi and Fei Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In *Proceedings of the 29th International Coference on ICML*, pages 1275–1282, 2012.

[123] Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *ICLR*, 2022.

[124] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *ICLR*, 2018.

[125] Yang Shu, Zhangjie Cao, Chenyu Wang, Jianmin Wang, and Mingsheng Long. Open domain generalization with domain-augmented meta-learning. In *CVPR*, pages 9624–9633, 2021.

[126] Yu Shu, Yemin Shi, Yaowei Wang, Tiejun Huang, and Yonghong Tian. P-odn: Prototype-based open deep network for open set recognition. *Scientific Reports*, 2020.

[127] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *ICLR*, 2015.

[128] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, pages 9120–9132. PMLR, 2020.

[129] Peng Su, Shixiang Tang, Peng Gao, Di Qiu, Ni Zhao, and Xiaogang Wang. Gradient regularized contrastive learning for continual domain adaptation. *AAAI*, 2021.

[130] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, 2016.

[131] Xin Sun, Zhenning Yang, Chi Zhang, Guohao Peng, and Keck-Voon Ling. Conditional gaussian distribution learning for open set recognition. In *CVPR*, 2020.

[132] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *CVPR*, pages 8725–8735, 2020.

[133] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[134] Tsung Wei Tsai, Chongxuan Li, and Jun Zhu. Mice: Mixture of contrastive experts for unsupervised image clustering. In *ICLR*, 2021.

[135] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 7167–7176, 2017.

[136] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[137] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *ECCV*, pages 268–285. Springer, 2020.

[138] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7492–7501, 2022.

[139] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. In *ICLR*, 2022.

[140] Ramakrishna Vedantam, David Lopez-Paz, and David J Schwab. An empirical investigation of domain generalization with empirical risk minimizers. *NeurIPS*, 34, 2021.

[141] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5018–5027, 2017.

[142] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *NeurIPS*, 31, 2018.

[143] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, pages 9929–9939. PMLR, 2020.

[144] Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. Transferable attention for domain adaptation. In *AAAI*, volume 33, pages 5345–5352, 2019.

[145] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *ICCV*, pages 834–843, 2021.

[146] Jianlong Wu, Keyu Long, Fei Wang, Chen Qian, Cheng Li, Zhouchen Lin, and Hongbin Zha. Deep comprehensive correlation mining for image clustering. In *ICCV*, pages 8150–8159, 2019.

[147] Yuan Wu, Diana Inkpen, and Ahmed El-Roby. Dual mixup regularized learning for adversarial domain adaptation. *ECCV*, 2020.

[148] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018.

[149] Haifeng Xia, Handong Zhao, and Zhengming Ding. Adaptive adversarial network for source-free domain adaptation. In *ICCV*, pages 9010–9019, 2021.

[150] Renjun Xu, Pelen Liu, Liyan Wang, Chao Chen, and Jindong Wang. Reliable weighted optimal transport for unsupervised domain adaptation. In *CVPR*, pages 4394–4403, 2020.

[151] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*, October 2019.

[152] Guanglei Yang, Haifeng Xia, Mingli Ding, and Zhengming Ding. Bi-directional generation for unsupervised domain adaptation. In *AAAI*, pages 6615–6622, 2020.

[153] Lei Yang, Dapeng Chen, Xiaohang Zhan, Rui Zhao, Chen Change Loy, and Dahua Lin. Learning to cluster faces via confidence and connectivity estimation. In *CVPR*, pages 13369–13378, 2020.

[154] Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin. Learning to cluster faces on an affinity graph. In *CVPR*, pages 2298–2306, 2019.

[155] Shiqi Yang, Joost van de Weijer, Luis Herranz, Shangling Jui, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *NeurIPS*, 34, 2021.

[156] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Unsupervised domain adaptation without source data by casting a bait. *arXiv preprint arXiv:2010.12427*, 2020.

[157] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *ICCV*, pages 8978–8987, 2021.

[158] Shiqi Yang, Yaxing Wang, Kai Wang, Shangling Jui, et al. Attracting and dispersing: A simple approach for source-free domain adaptation. In *NeurIPS*, 2022.

[159] Shiqi Yang, Yaxing Wang, Kai Wang, Shangling Jui, and Joost van de Weijer. One ring to bring them all: Towards open-set recognition under domain shift. *arXiv preprint arXiv:2206.03600*, 2022.

[160] Shaokai Ye, Kailu Wu, Mu Zhou, Yunfei Yang, Sia Huat Tan, Kaidi Xu, Jiebo Song, Chenglong Bao, and Kaisheng Ma. Light-weight calibrator: a separable component for unsupervised domain adaptation. In *CVPR*, pages 13736–13745, 2020.

[161] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Universal domain adaptation. In *CVPR*, pages 2720–2729, 2019.

[162] Hongjie Zhang, Ang Li, Jie Guo, and Yanwen Guo. Hybrid models for open set recognition. In *ECCV*, pages 102–117. Springer, 2020.

[163] Yabin Zhang, Bin Deng, Kui Jia, and Lei Zhang. Label propagation with augmented anchors: A simple semi-supervised learning baseline for unsupervised domain adaptation. In *ECCV*, pages 781–797, 2020.

[164] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *CVPR*, pages 5031–5040, 2019.

[165] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *ICML*, pages 7404–7413, 2019.

[166] Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. *NeurIPS*, 33:14435–14447, 2020.

[167] Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10867–10875, 2021.

[168] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*, pages 1318–1327, 2017.

[169] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Learning placeholders for open-set recognition. In *CVPR*, pages 4401–4410, 2021.

[170] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *ECCV*, pages 561–578. Springer, 2020.

[171] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *ICLR*, 2021.

[172] Wenzhang Zhou, Dawei Du, Libo Zhang, Tiejian Luo, and Yanjun Wu. Multi-granularity alignment domain adaptation for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9581–9590, 2022.

[173] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *NeurIPS*, 33, 2020.

[174] Ronghang Zhu and Sheng Li. Crossmatch: Cross-classifier consistency regularization for open-set single domain generalization. In *ICLR*, 2022.

[175] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, pages 6002–6012, 2019.

[176] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the ECCV (ECCV)*, pages 289–305, 2018.