MULTI-SOURCE AND SOURCE-PRIVATE CROSS-DOMAIN LEARNING FOR VISUAL RECOGNITION

by

Qucheng Peng

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

Indianapolis, Indiana

May 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Lingxi Li, Chair

Department of Electrical and Computer Engineering

Dr. Zhengming Ding

Department of Computer Science, Tulane University

Dr. Qingxue Zhang

Department of Electrical and Computer Engineering

Dr. Brian King

Department of Electrical and Computer Engineering

Approved by:

Dr. Brian King

To 2019 – 2022

ACKNOWLEDGMENTS

I want to thank Dr. Ding first who serves as my supervisor for nearly two years. With his selfless instructions, I finished my fist paper in the field of Artificial Intelligence and got tickets in the PhD applications. PhD positions in AI area are very competitive. Without his help, it is a mission impossible for me to get admitted and achieved scholarships. During the journey, senior students Haifeng and Taotao in Dr. Ding's group also help me a lot, and Haifeng gave me a lot of advice for the offer decision. Thank them too. In the coming fall semester, I will continue my study as a PhD student.

I want to thank Dr. Salama, Dr. King, Dr. Its, and Dr. Ding for their references. Recommendation is always an important part for the application. Without their strong and positive references, it's hard for me to receive these offers.

I want to thank Dr. Li, Dr. Ding, Dr. Zhang and Dr. King for becoming a part of the defense committee. I understand that all of them are very busy and it is very kind of them to participate in the thesis defense.

I want to thank Dr. Li, Dr. King and Ms. Tucker for helping me deal with a lot of registration affairs in the past two years.

Special thanks to all the things encourage me and stay with me, like music from Cyndi Wang, Adele and JT. Due to the COVID-19, all the things have become different since 2020 but finally I'm on the road best left behind.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

x	sample
y	label
$\hat{y}$	pseudo label
$\mathcal{S}$	source domain
$\mathcal{T}$	target domain
$\mathcal{L}$	loss
$\mathbb{E}$	expectation
$\mathcal{F}$	Fourier transform
$\mathcal{F}^{-1}$	inverse Fourier transform
$\mathcal{A}$	amplitude
$\mathcal{P}$	phase

ABBREVIATIONS

EC-MSA	Enhanced Consistency Multi-Source Adaptation
FTA-FDA	Fourier Transform-Assisted Federated Domain Adaptation
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
MMD	Maximum Mean Discrepancy
FFT	Fast Fourier Transform
t-SNE	t-distributed Stochastic Neighbor Embedding

ABSTRACT

Domain adaptation is one of the hottest directions in solving annotation insufficiency problem of deep learning. General domain adaptation is not consistent with the practical scenarios in the industry. In this thesis, we focus on two concerns as below.

First is that labeled data are generally collected from multiple domains. In other words, multi-source adaptation is a more common situation. Simply extending these single-source approaches to the multi-source cases could cause sub-optimal inference, so specialized multi-source adaptation methods are essential. The main challenge in the multi-source scenario is a more complex divergence situation. Not only the divergence between target and each source plays a role, but the divergences among distinct sources matter as well. However, the significance of maintaining consistency among multiple sources didn't gain enough attention in previous work. In this thesis, we propose an Enhanced Consistency Multi-Source Adaptation (EC-MSA) framework to address it from three perspectives. First, we mitigate feature-level discrepancy by cross-domain conditional alignment, narrowing the divergence between each source and target domain class-wisely. Second, we enhance multi-source consistency via dual mix-up, diminishing the disagreements among different sources. Third, we deploy a target distilling mechanism to handle the uncertainty of target prediction, aiming to provide high-quality pseudo-labeled target samples to benefit the previous two aspects. Extensive experiments are conducted on several common benchmark datasets and demonstrate that our model outperforms the state-of-the-art methods.

Second is that data privacy and security is necessary in practice. That is, we hope to keep the raw data stored locally while can still obtain a satisfied model. In such a case, the risk of data leakage greatly decreases. Therefore, it is natural for us to combine the federated learning paradigm with domain adaptation. Under the source-private setting, the main challenge for us is to expose information from the source domain to the target domain while make sure that the communication process is safe enough. In this thesis, we propose a method named Fourier Transform-Assisted Federated Domain Adaptation (FTA-FDA) to alleviate the difficulties in two ways. We apply Fast Fourier Transform to the raw data and transfer only the amplitude spectra during the communication. Then frequency space

interpolations between these two domains are conducted, minimizing the discrepancies while ensuring the contact of them and keeping raw data safe. What's more, we make prototype alignments by using the model weights together with target features, trying to reduce the discrepancy in the class level. Experiments on Office-31 demonstrate the effectiveness and competitiveness of our approach, and further analyses prove that our algorithm can help protect privacy and security.

1. INTRODUCTION

1.1 Overview of General Domain Adaptation

Recent decades witness that deep learning has gained incredible success in multiple areas such as computer vision from [1] and natural language processing from [2]. [1] proposes a model called ResNet which greatly improve the performance of image classification task, while [2] provides a pretty much strong representation model named BERT that benefits almost all the downstream tasks in the natural language processing area. The common key element of their success is the availability of large-scale labeled data. The pretrain of ResNet is based on Imagenet ([3]), while that of BERT depends on the WordPiece embeddings with a 30,000 token vocabulary ([4]).

In practical situations, however, it is not easy to acquire abundant labeled data. After all, it is hard to invest so much on data processing like Google since manual annotation is time-consuming and expensive. Having limited labels or even no label is a very common phenomenon for most of us. Another common assumption we get accustomed is that the data we need to inference share the same distribution with the labeled data. However, this assumption is easy to be distorted. Consider a situation that we have labeled data and unlabeled data which are generated under different circumstances such as styles, angles, cameras or light but share the same label space. If we simply ignore the distribution shifts between them, the inference will not be effective enough.

In such cases, domain adaptation is one of the solutions to dealing with two issues we mentioned above. In our setting, source domain is a collection of data which own labels while target domain includes data that have no label information. Domain adaptation aims to adapt models by transferring knowledge from the source domain to the target domain so that even without label information from the target domain, models can still work well on the target classification tasks ([5]). Actually, it is a specific type of transfer learning methods.

Based on the theoretical work from [6], [7], a large amount of single-source domain adaptation approaches have been proposed, which only explore labeled data from one single domain to infer the data from target domain. There exist two major groups of solutions with different strategies to mitigate the cross-domain shift. The first branch is metric-based

method, which hopes to find a metric to measure and minimize the discrepancy between the source domain and the target domain. MMD by [8], DDC by [9], D-CORAL by [10], DAN by [11] and DCC by [12] are representative jobs for this branch. While the second branch is adversarial approach, which regards the source and target domain as two competitors in a min-max game, and the discrepancy becomes smaller during the competition. Prior work comes from RevGrad by [13], MCD by [5], CyCADA by [14] and DRANet by [15]. Although these two paradigms consider domain adaptation from different views, their common goal is to minimize the domain discrepancy explicitly or implicitly.

Till now, we have talked about the necessity of research on domain adaptation and the progress of general domain adaptation. In the next three sections, we will first introduce some related work of general domain adaptation, then turn to multi-source domain adaptation and federated domain adaptation. To be concrete, we will discuss why we need these two types of varied adaptations, and how we can improve them in this thesis.

1.2 Related Work of General Domain Adaptation

Motivated by the seminal deductions from [6] and [7], a large number of methods have been proposed. Under certain assumptions, model's target error can be bounded by its source error and the divergence between the source and target domains. Metric-based and GAN-based approaches are the two major routes in the single-source domain adaptation. Metric-based method tends to measure the discrepancy between the source and target domain explicitly. [11] uses maximum mean discrepancy, while [9] applies deep domain confusion and [10] utilizes correlation relation. [16] uses this paradigm to solve heterogeneous domain adaptation. [17] makes jointly clustering and discrimination for alignment together with contrastive learning. The GAN-based method origins from [18]. It builds a min-max game for two players related to the source and target domains so as to minimize the discrepancy implicitly during the competition. [13] adopts domain to confuse the two players, while [5] uses classifier discrepancy as the objective. [19] also applies this paradigm but pay attention to a new kind of domain called neuromorphic vision sensing domain. With the invention of [20], [21] proposed CyCADA, considering this problem from the pixel level and creating

a reconstruction process between these two domains. Besides, there are other ideas that prove to work. [22], [23] and [24] attempt to alignment these two domains class-wisely on feature-level. Mix-up is also a popular augmentation technique, as [25] adds dual mix-up and [26] applies domain mix-up.

1.3 Motivation and Objective of Multi-Source Domain Adaptation

Multi-source domain adaptation dedicates to transfer the knowledge from multiple source domains to one target domain with unlabeled data. It fits the actual application scenarios better. For example, when we gather a lot of medical images for the use of segmentation, the different environments from different hospitals ensure that these images come from different domains ([27]). Another instance origins from analysis on distinct mobile devices in this era, which different customer habits lead to label distribution shifts ([28]). Therefore, multi-source domain adaptation gains more and more attention in the community recently.

The main challenge in the multi-source problem is a more complex divergence situation among multiple domains. Not only the divergence between the target and each source domain, but also the divergences among distinct source domains should be considered. There are also two major technical routes for the multi-source problem. First is the statistic matching method, aiming to aligning different domains statistically. M3SDA by [29], MCC by [30] and MSCLDA by [31] belong to this type. Second is the GAN-based approach, trying to build min-max games for each source-target pair so that the discrepancies can be reduced implicitly. DCTN by [32], MDAN by [33] and MSDTR by [34] apply this strategy. The common strategy of these two routes is aligning from two perspectives. We need to align each pair of source and target while ensure the consistency among all the sources in order to get an ideal prediction on target. Besides, ensemble strategy is also an significant part for multi-source adaptation. Popular methods like simple average from [35], distance weighting from [24] and knowledge distilling from [36] are representative work.

Although prior work is excellent and instructive, the potential weaknesses could not be ignored, and that's the reason why we propose our new method and write this thesis. First is a lack of suitable conditional alignment methods. Current class-wise methods tend to

cause big shifts in one single mini-batch and make the model not robust enough in a single batch training. Second, the consistency among multiple sources should be reinforced. Since we have so many source domains to be considered along with their pairwise discrepancies, a balanced situation will largely improve the model's performance, especially when we are applying a simple averaging ensemble strategy. Third is the noise inside the target's pseudo-labels. These noisy samples will directly impact the formation of our model. To address these issues above, we propose our approach named Enhanced Consistency Multi-Source Adaptation (short for EC-MSA) to enhance consistency from different aspects. We conclude our contributions from following perspectives:

- First, we use an adaptive conditional alignment strategy by making iterative update on centroids of each domain. This method could resist noise from single sample and avoid big shifts in one mini-batch.
- Second, we enhance the consistency among distinct source classifiers via target data augmentation, which ensure more agreement among different classifiers.
- Finally, we filter the low-confident target samples by adopting a target distilling mechanism. It will purify the target domain and thus help the previous two parts.

1.4 Motivation and Objective of Federated Domain Adaptation

Federated domain adaptation aims to transfer knowledge from the source domain to the target domain while keep the privacy and security of raw data. To be concrete, we need to keep the source and target data stored locally so that they cannot be fed into the neural network at the same time. It follows the trend in the industry that more and more users concern about the security of their private data and worry about the potential leakage. Besides, there are more and more countries and regions signing the regulations that protect the data privacy and security of citizens. Therefore, it is really necessary to focus on this topic.

The main challenge in the federated domain adaptation is the conflict between information exposure and protection. We hope to expose more source information to the target

client without transferring raw data, but also want to decrease the risk of data leakage during the training process. Generally there exist two technique routes. First is the non source-free adaptation, like [37], [38] and [27]. In this paradigm, sample-based information like gradient will be sent to the target domain while the raw data are excluded from the communication. Besides, we need to ensure that the recovery of the raw data from the sample-based information is impossible. Second is the source-free adaptation, like [39], [40] and [41]. Under this setting, only model parameters are allowed in the communications and sample-based source information is completely forbidden.

Prior work is really enlightening to us while the weaknesses are still need to be addressed. For the non source-free algorithms, people still worry about the potential recovery based on sample-wise information. In fact, [42] has proved that it is possible to reconstruct the raw source data with gradient leakage attacks if source's gradients are transferred to the target client as FADA ([37]) does. For the source-free algorithms, it always owns a more loose empirical risk bound than the non source-free versions according to the theoretical deductions from [43]. It's intuitive to us because sample-wise information is not allowed and the alignment will become harder. In practice, the source-free setting treats the adaptation problem as one supervised learning process on source and one self-supervised process on target, which lacks an explicit connection between these two domains. To alleviate these difficulties, we propose our method called Fourier Transform-Assisted Federated Domain Adaptation (FTA-FDA) and show our contributions in the following three folds:

- First, we separate the raw images in the frequency space and transit only a small proportions of the amplitude spectra. Frequency space interpolation is applied to the alignments between the source and target domains.
- Second, we conduct prototype alignments with the help of model's weights, ensuring the source-private setting and reducing class-wise discrepancies between the two domains.
- Finally, we evaluate our method on the popular dataset, and design a series of experiments to exhibit the security of it.

1.5 Organization

In this section, we make an introduction to the themes of this thesis. The rest chapters of the paper are organized in the following. Chapter 2 includes the background knowledge of the thesis that may help people understand the thesis better. In Chapter 3, we present our novel multi-source approach named Enhanced Consistency Multi-Source Adaptation (short for EC-MSA) in an end-to-end fashion, along with all the components of the model and some theoretical analyses. Besides, we also provide the experiments related to EC-MSA on several benchmark datasets in Chapter 3, together with the implementation details and result comparisons, followed by some empirical analyses. In Chapter 4, we present our novel source-private approach named Fourier Transform-Assisted Federated Domain Adaptation (short for FTA-FDA), along with all the elements of the model. Besides, we also provide the experiments related to FTA-FDA on the benchmark dataset in Chapter 4. What's more, we also talk about the effectiveness of security protection in this chapter. In the end, Chapter 5 concludes the previous parts of the thesis and then gives some hints on the future work for our topics.

2. BACKGROUND

2.1 Backbone

Although the foundations of domain adaptation come from traditional methods like feature selection ([44]), distribution adaptation ([45]) and subspace learning ([46]), no one can ignore the great evolution that deep learning brings. It is probably due to the reason that deep learning could extract more expressive transferable representations from the deep neural network ([47]).

However, the use of deep neural network could cause another problem. That is, the progress of a model is just achieved by a more complex model, instead of a more suitable adaptation or a better-defined loss function. In such a case, the core objective of domain adaptation problem will be blurred.

Therefore, scholars in this community form a consensus that certain types of backbones should be applied to avoid the unfair situation mentioned above. The most popular backbones are AlexNet ([48]) and ResNet ([1]).

2.1.1 AlexNet

AlexNet origins from the paper [48]. It is a specific type of Convolutional Neural Network and just combine several tricks like Rectified Linear Unit (ReLU) from [49] and Dropout from [50], but received an excellent test error improvement on the ImageNet LSVRC-2012 task. Gradually, it becomes a popular backbone to use in the domain adaptation community and lots of work has applied it to demonstrate the effectiveness.

Figure 2.1 (image from [1]) shows the architecture of AlexNet. Generally we tend to keep all the layers except the final fully-connected layer. The last layer will be replaced according to the number of classes.

Two diagrams are followed in the community. First is to use the extracted features after certain layers. For example, [47] uses the output of the last but one layer as features. This kind of method will feed these extracted features to some fully-connected layers and fine-tune on them directly. Since very simple network is adopted after feature extraction, the training

[1]↑https://en.wikipedia.org/wiki/AlexNet

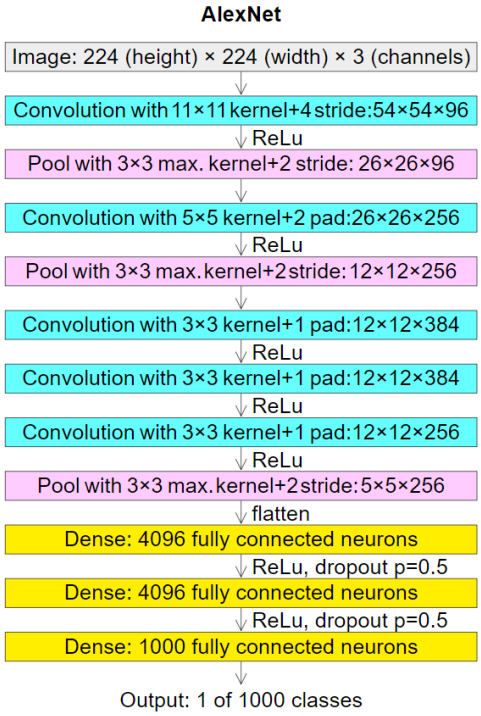


Figure 2.1. AlexNet. (Image from https://en.wikipedia.org/wiki/AlexNet) (Best viewed in color.)

process can save a lot of time. However, it might limit the potential of a new method as the training model is too shallow and cannot meet the end-to-end requirement, which is pretty much common in the industry.

Another is to use the whole network, as most work does. For all the things prior to the linear layers, we can fix them or assign a very small learning rate to them. For the fully connected layers, we can continue to use the original version or replace them with a more complex version, with four layers or even more. In such a case, the raw data can be used directly so that the training can be shaped in an end-to-end fashion. However, it will certainly take more time and use more resource.

2.1.2 ResNet

ResNet is proposed by [1]. Apart from the tricks from [48], ResNet adds residual blocks to the network so that the degradation problem in the deep network can be alleviated.

Table 2.1. Architecture of ResNet. (Image from [1])

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

For the supervised tasks like ImageNet LSVRC and COCO, ResNet has surpassed AlexNet completely. As for domain adaptation tasks, generally ResNet performs better than AlexNet when equipped with the same methods. Therefore, ResNet has a wider use than AlexNet in the community.

Table 2.1 (image from [1]) shows 5 branches of ResNet named as ResNet-18, ResNet-34, ResNet-50, ResNet-101 and ResNet-152. In some papers you can see the utilization of ResNet-18 and ResNet-34, but since they are relatively simple and do not have the bottleneck architectures, they are not very common. As for ResNet-152, it *does* have the most complicated structure among all these architectures while doesn't show any advantages, thus is hardly seen in all the scholars' work. The mainstream branches are ResNet-50 and ResNet-101. ResNet-50 tends to be used in simpler tasks and ResNet-101 is more suitable to difficult tasks, especially some tasks related to semantic segmentation.

For ResNet-50 and ResNet-101, we also have two diagrams. First is the feature extraction. The outputs after the adaptive average pooling layer are regarded as the features. Same as AlexNet, we feed these features to some fully-connected layers and fine-tune on them directly. The pros and cons are the same as mentioned in the previous subsection.

For using the whole network, things are quite different because we have two kind of protocols. One is to replace the fully-connected layers as AlexNet. The other is to replace

the part starting from the adaptive average pooling layer, as [35] does. The advantages and disadvantages are the same as mentioned in the previous subsection so we don't spend more time on that.

2.2 Distance Function

Although deep neural network can improve the model's performance greatly, it doesn't exert on the perspective of adaptation, which is what distance functions can help. In this section, we will talk about the popular distance functions in the community. All of them offer an empirical estimation of the discrepancy between different domains.

2.2.1 KL Divergence

KL divergence ([51]) is short for Kullback-Leibler divergence. It has been widely used in some work such as [52] and [53]. We assume that the distributions for two domains are $P(x)$ and $Q(x)$, and the probability space is $\mathcal{X}$, then KL divergence is denoted as:

$$\mathcal{D}_{KL}(PQ) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}. \tag{2.1}$$

2.2.2 JS Divergence

JS divergence ([54]) is short for Jensen-Shannon divergence. It is on the basis of KL divergence. We still use P and Q to represent two domains, while $M = \frac{1}{2}(P + Q)$, then the JS divergence is:

$$\mathcal{D}_{JS}(P||Q) = \frac{1}{2}[\mathcal{D}_{KL}(P||M) + \mathcal{D}_{KL}(Q||M)] \tag{2.2}$$

Compared with KL divergence, JS divergence is a more symmetric distance function. Work like [55] has applied it as a part of the loss function. Some self-supervised papers like [56] also adopt this metric.

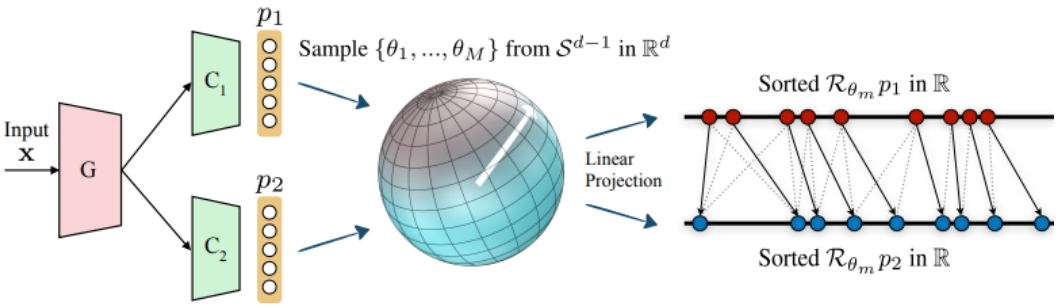


Figure 2.2. Sliced Wasserstein Distance. (Image from [59]) (Best viewed in color.)

2.2.3 Sliced Wasserstein Distance

Wasserstein distance ([57]) first appeared in the community when [58] came out. Then [59] extended it to the field of domain adaptation as sliced wasserstein distance. Generally, we use wasserstein-1 version, with another name called Earth-Mover. Assume γ is the joint distribution of P and Q as we used previously, and $\Pi(P, Q)$ is the set of all joint distributions, then we denote the wasserstein distance as:

$$\mathcal{WD}(P, Q) = \inf_{\gamma \in \Pi(P,Q)} \mathbb{E}_{\substack{(x_1,x_2)\in\gamma \\ x_1 \in P, x_2 \in Q}} \|x_1 - x_2\| \tag{2.3}$$

Sliced wassertein distance as shown in Figure 2.2 (image from [59]) is extended by it. Here we use a linear projection $\mathcal{R}_\theta$ to P and Q and sum the results over the sphere:

$$\mathcal{SWD}(P, Q) = \int_{\mathcal{S}^{d-1}} \mathcal{WD}(\mathcal{R}_\theta P, \mathcal{R}_\theta Q) d\theta \tag{2.4}$$

Compared with KL divergence and JS divergence, sliced wasserstein distance is much more sensible, especially when the distributions are supported by low dimensional manifolds. This is because probability space's underlying geometry properties are taken into consideration.

2.2.4 Mutual Information

Mutual information ([60]) is a popular estimation tool in the area of self-supervised learning. As more and more work in the domain adaptation begins to view the task in a similar perspective, it is necessary to get exposed to this distance function. Mutual information aims to measure the similarity between different distributions. Assume distributions for two domains are $P(x)$ and $Q(x)$, and γ is the joint distribution of P and Q. We can conclude:

$$\mathcal{MI}(P, Q) = \sum_{x_1 \in P} \sum_{x_2 \in Q} \gamma(x_1, x_2) \log \frac{\gamma(x_1, x_2)}{P(x_1)Q(x_2)} \tag{2.5}$$

Generally, we hope to maximize mutual information between different domains so that the domain-specific information can be disentangled like [61] does. It also plays a significant role in the federated and source-free domain adaptation like [62] and [39].

2.2.5 Maximum Mean Discrepancy

Maximum Mean Discrepancy ([63]) is one of the most popular distance functions in the domain adaptation community. We use $\phi(\cdot)$ to map multiple samples to the Reproducing Kernel Hilbert Space (RKHS), which is represented as $\mathcal{H}$. The whole expression is denoted as:

$$\mathcal{MMD}(P, Q) = \|\mathbb{E}_{x_1 \in P}[\phi(x_1)] - \mathbb{E}_{x_2 \in Q}[\phi(x_2)]\|^2_{\mathcal{H}} . \tag{2.6}$$

This distance function is based on the adaptation of marginal distribution. Following this, conditional adaptation ([64]) and joint adaptation ([47]) are proposed. If we represent the label space as $\mathcal{C}$, then the conditional Maximum Mean Discrepancy can be denoted as:

$$\mathcal{CMMD}(P, Q) = \mathbb{E}_{c \in \mathcal{C}} \|\mathbb{E}_{x_1 \in P^c}[\phi(x_1)] - \mathbb{E}_{x_2 \in Q^c}[\phi(x_2)]\|^2_{\mathcal{H}} . \tag{2.7}$$

As for joint adaptation, it combines the marginal version and the conditional version:

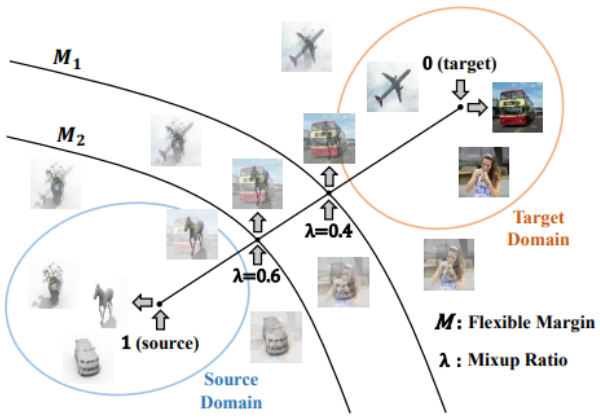


Figure 2.3. Details of Domain Mix-up. (Image from [26]) (Best viewed in color)

$$\mathcal{JMMD}(P,Q) = \|\mathbb{E}_{x_1\in P}[\phi(x_1)] - \mathbb{E}_{x_2\in Q}[\phi(x_2)]\|^2_{\mathcal{H}} + \mathbb{E}_{c\in\mathcal{C}} \|\mathbb{E}_{x_1\in P^c}[\phi(x_1)] - \mathbb{E}_{x_2\in Q^c}[\phi(x_2)]\|^2_{\mathcal{H}}. \tag{2.8}$$

2.3 Data Augmentation

Data augmentation is very effective for both supervised and unsupervised learning due to the fact that it can improve the generalization ability of models. When it comes to the domain adaptation problem, things become a little different. Two types of augmentations are adopted. One is to just assist the self-supervised process, such as pseudo-labeling. The other is designed for the domain adaptation task specifically. In this section, we will place extra emphasis on the second type and several methods will be introduced.

2.3.1 Domain Mix-up

Domain mix-up was first proposed by [26] and then also utilized by [65]. The core idea is to mix up two domains so that the adversarial process can learn the implicit transferable information better.

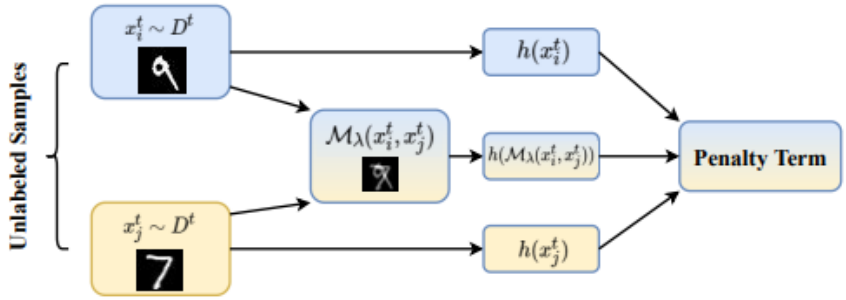


Figure 2.4. Details of Dual Mix-up. (Image from [25]) (Best viewed in color)

It is shown as Figure 2.3 (image from [26]). During the mix-up procedure, mixed samples and soft labels with mix-up ratios are created, together with a flexible margin. There exists two levels of domain mix-up: pixel-level domain mix-up and feature-level mix-up. They are all built to lead the model to behave linearly among distinct domains.

2.3.2 Dual Mix-up

Dual mix-up ([25]) makes data augmentation on pixel level but conducts categorical mix-up and domain mix-up. The details are shown in Figure (2.4) (image from [25]).

For domain mix-up, it serves the same function as mentioned above, which encourage a more linear behavior among multiple domains. What's more, the dual mix-up could also improve the consistency among different predictions. There exists other mix-up work like [66], but we won't spend more time on it.

2.3.3 Adversarial Feature Augmentation

Adversarial feature augmentation is proposed by [67]. According to the authors, it is the first time that GAN ([18]) has been used to do data augmentation on the feature level. The details are displayed in Figure 2.5 (image from [67]).

It can be observed that Step 1 and Step 2 both apply the augmentation by GAN. Random noise is added and then augment features from different domains. The GAN architecture

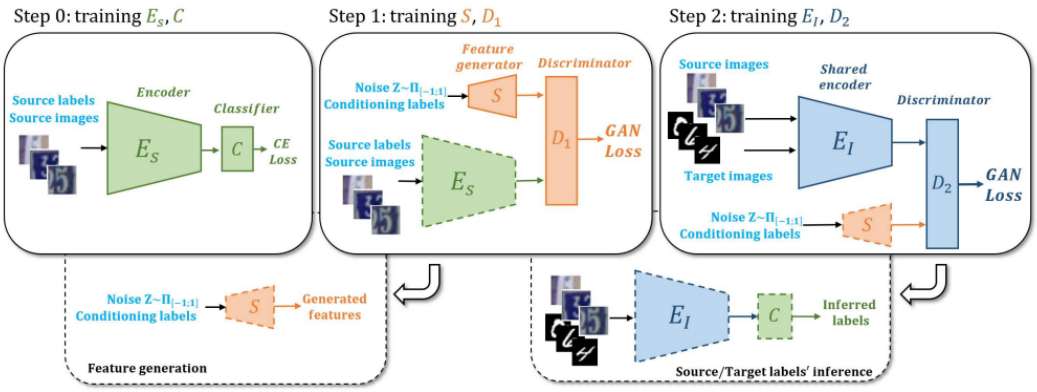


Figure 2.5. Details of Feature Augmentation. (Image from [67]) (Best viewed in color)

can help the model learn target features which are indistinguishable from the source, thus improve the whole adaptation process.

2.4 Federated Learning

Federated learning is really a hot topic nowadays, especially when we are in the times of Mobile Internet. Everyone cares more about leakage and illegal use of personal information, and the government pays more attention to the regulations about these issues. In such a case, a specific machine learning paradigm that can help protect security and privacy is very necessary, and that is federated learning. [68] is the first paper which proposed this concept completely. From the famous survey [28] in this field , we can get the definition of federate learning. It is a machine learning setting where multiple clients (for example, multiple cellphones or laptops) collaborate in solving a machine learning problem. During the process, we always need to ensure that raw data of each client are in a state of locally storage and these raw data are forbidden to be exchanged or transferred all the time. What's more, a large number of empirical experiments ([42]) show that federated learning can largely reduce computational complexity, thus save time and money for the academia and the industry.

Generally, there exist two kinds of federated learning, centralized and decentralized. Figure 2.6 (image from [69]) exhibits the situation of centralized version, where a central server will orchestrate all the training and communicate with all the clients without getting

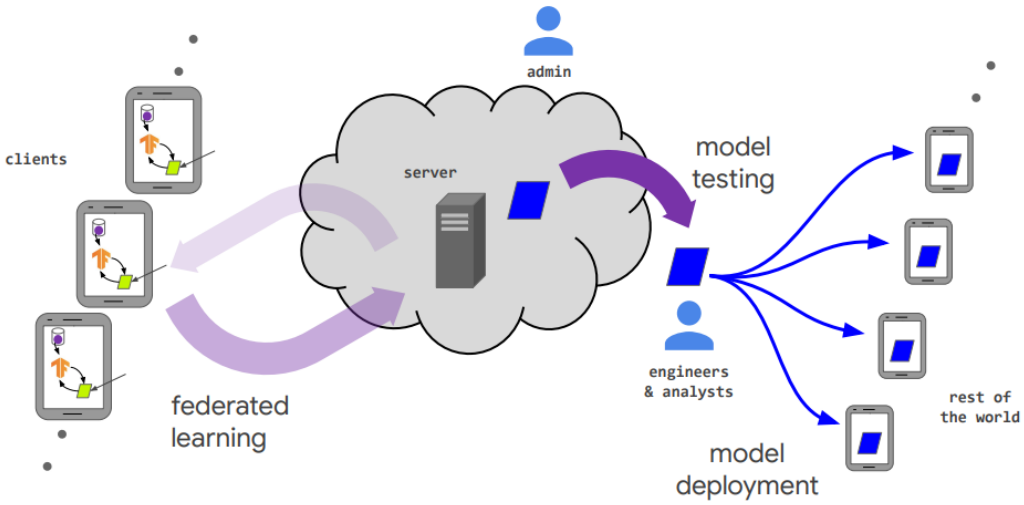


Figure 2.6. Federated Learning. (Image from [69]) (Best viewed in color)

in touch with any raw data. The decentralized version is a little different. It follows a peer-to-peer topology, which means no central server will be implemented, along with a connectivity graph that could be dynamic. Work like [42] applies this structure. Here we mainly focus on the centralized one.

Following what Figure 2.6 shows, we demonstrate the workflow of centralized federated learning, which includes six steps:

- **Problem setting:** First, the Machine Learning system should decide a machine learning problem to be solved.
- **Client preparation:** Next, we need to choose several clients to join the life-cycle. These clients will be instrumented with local data and initial models if exist. Commonly not all the clients will be available in one epoch.
- **Proxy data simulation:** This step is optional. In fact, for some federated learning approaches it can be skipped. Since raw data from any clients are not allowed to be exchanged or transferred, using proxy data could be a good choice. Here we can generate proxy data for each client using methods like dataset distillation ([70]), frequency space interpolation ([27]) and GAN ([41]).

- **Training:** Then we can start the training process. Sometimes we need to transfer the proxy data to each client. Then two options can be selected. First is that we can use just one model and different clients leveraging different variations or parameters of one single model, like [42] does. Second is that we try distinctive models for multiple clients, then try to combine them with different strategies like averaging ([68]) or voting ([71]).

- **Evaluation:** After the task have gained sufficient leverages, we need to analyze the effectiveness of all the candidates and select the best model. The selection standard could be either the candidates' performances on the data center's dataset or the combined performances on all the clients' local data.

- **Deployment:** In the end, one model has been selected. Then this model will be launched to all the clients that will be activated in the next epoch.

2.5 Fourier Transform

Fourier Transform is a useful tool for the synthesis and decomposition of signals. ([72]) With this tool, we can project signals from the time space to the frequency space, thus more properties of the signals can be found. The basic transforms between these two spaces are:

$$X(\mathrm{j}\omega) = \int_{-\infty}^{+\infty} x(t)\mathrm{e}^{-\mathrm{j}\omega t}dt, \tag{2.9}$$

and

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} X(\mathrm{j}\omega)\mathrm{e}^{\mathrm{j}\omega t}d\omega. \tag{2.10}$$

This version is generally called Continuous-Time Fourier Transform (short for CTFT). Equation 2.9 is the Fourier transform, which converts time space signal to the frequency space. Equation 2.10 is the inverse Fourier transform, which converts frequency space signal to the time space.

CTFT is the foundation for different kinds of transforms but that's not enough. Let's consider a situation that we need to deal with signals or data in the time space on computers. Generally, it's very hard for machine to compute on continuous data. ([73]) Therefore, it's

important to develop a transform that focus on discrete data in the time space. We sample from $x(t)$ to get the discrete time series as $x[n]$ and then the transforms become:

$$X(\mathrm{e}^{\mathrm{j}\omega}) = \sum_{-\infty}^{+\infty} x[n]\mathrm{e}^{-\mathrm{j}\omega n}, \tag{2.11}$$

and

$$x[n] = \frac{1}{2\pi}\int_{2\pi} X(\mathrm{e}^{\mathrm{j}\omega})\mathrm{e}^{\mathrm{j}\omega n}d\omega. \tag{2.12}$$

These two equations are called Discrete-Time Fourier Transform pair (short for DTFT). Equation 2.11 is the discrete Fourier transform, which converts the time space series $x[n]$ to the frequency space. Equation 2.12 is the inverse discrete Fourier transform, which converts the spectrum $X(\mathrm{e}^{\mathrm{j}\omega})$ to the time space.

After sampling from the time space, it's very natural for us to extend it to the frequency space. Of course, we also need machine's help when facing problems from the frequency space. If N samples will be selected evenly from the frequency space, the interval would be $\frac{2\pi}{N}$. First, we will introduce a useful notation as:

$$W_N = \mathrm{e}^{-\mathrm{j}(2\pi/N)}. \tag{2.13}$$

Then the Discrete Fourier Transform pair (short for DFT) can be represented as:

$$X[k] = \begin{cases} \sum_{n=0}^{N-1} x[n]W_N^{kn}, & 0 \le k \le N-1, \\ 0, & otherwise, \end{cases} \tag{2.14}$$

and

$$x[n] = \begin{cases} \frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn}, & 0 \le n \le N-1, \\ 0, & otherwise. \end{cases} \tag{2.15}$$

Similar as previous, Equation 2.14 is the analysis equation that transfers time space signals to the frequency space, while Equation 2.15 is the synthesis equation that transfers frequency space information to the time space.

Nowadays, the actual operations that most machines conduct are DFTs, but a more advanced algorithm named Fast Fourier Transform (short for FFT) has been implemented while keep the same results. ([74]) It can greatly reduce the time complexity of DFT from $O(N^2)$ to $O(\log N)$, thus much more friendly when computers are required, and many frameworks like PyTorch ([75]) and software like MATLAB ([76]) apply it as the default algorithm for all kinds of Fourier transforms. We are not going to talk about the details of how it makes improvements, but the core idea is that it's just a fast version of DFT.

3. ENHANCED CONSISTENCY MULTI-SOURCE DOMAIN ADAPTATION

3.1 Preliminaries and Motivations

For a typical multi-source domain adaptation problem, there are N well-labeled source domains, where $\mathcal{S}_{\mathrm{j}} = \{(x_{\mathrm{i}}^{s_{\mathrm{j}}}, y_{\mathrm{i}}^{s_{\mathrm{j}}})\}_{\mathrm{i}=1}^{n_{s_{\mathrm{j}}}}$ represents the dataset of source domain j with $n_{s_{\mathrm{j}}}$ labeled samples. What's more, we assume a source domain space $\mathcal{S} = \{1, 2, ..., N\}$. Besides, there exists a target domain dataset $\mathcal{T} = \{x_{\mathrm{i}}^{t}\}_{\mathrm{i}=1}^{n_t}$ that includes n_t unlabeled samples. These $N+1$ domains share the same label space but lie in different distributions. The label space is denoted as $\mathcal{D} = \{1, 2, ..., K\}$, while K is the number of classes. In such a case, the goal is to seek a good classification model that could achieve a high accuracy on the target domain.

For multi-source domain adaptation problem, plenty data will generalize the model better intuitively. However, they also bring a more complicated divergence situation. That is, not only the divergence between source and target domains brings challenges, but also the divergence across multiple source domains matters as well. Prior work like [13] and [10] shows that a lack of concentrating on the shifts among distinct source domains like the single-best and source-combine standards tends to lead a sub-optimal solution. Therefore, a specialized multi-source domain adaptation method is necessary and valuable in performance improvement.

3.2 Related Work

Theoretical analyses from [77] and [21] provide solid foundations for research under this topic. Similar to the single-source problem, there also exist two mainstream routes: GAN-based route and statistic matching route. The GAN-based method builds min-max games for each source and target domain. [32] uses domain discriminators to construct their model. [33] adds a known-unknown discrimination to enhance the min-max game. Recent work from [78] gets instructions from CycleGAN [20] and tries to establish cycle-consistency between each source and target domain. Statistic matching strategy aims at finding an explicit metric to represent discrepancies among multiple domains. [29] used moment matching to

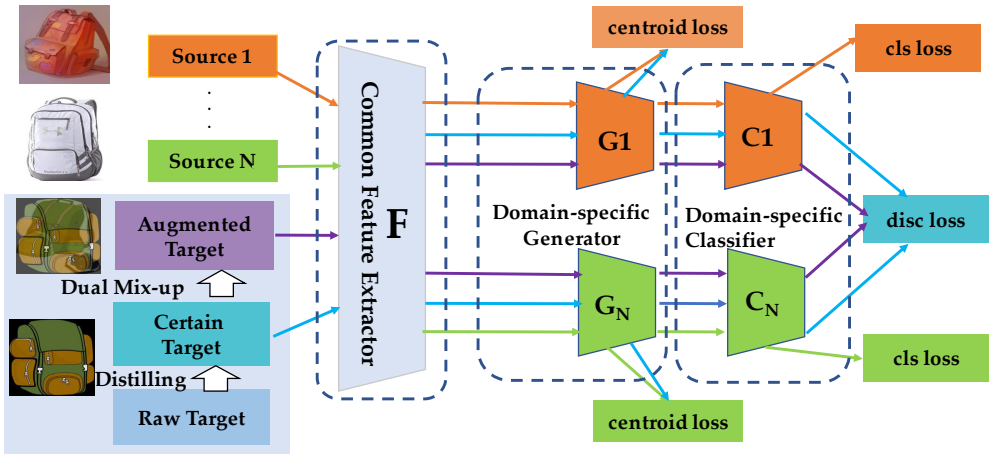


Figure 3.1. Overview of our proposed Enhanced Consistency Multi-Source Adaptation Network (EC-MSA), which contains one shared feature extractor F, N domain-specific generators $\{G_1\}, \cdots, \{G_N\}$ and source classifiers $\{C_1\}, \cdots, \{C_N\}$. Sample images are chosen from distinct domains of the Office-Home dataset. (Best viewed in color.)

align different domains. [30] proposes minimum class confusion to measure the discrepancy. [31] does alignment on domain-level and class-level simultaneously. [79] and [80] both follow this paradigm along with attention module. Besides, using federated learning like [39], [42] is a popular research now, and other strategies like implicit alignment [81] and source distilling [24] also gain much attention. Ensemble methods for the final inference is also an important part to delve into for multi-source adaptation. [35] applies an average strategy, while [24] tackles it with a distance-based weighting approach. [36] uses knowledge distilling to deal with the ensemble part.

Our model EC-MSA adopts the statistic strategy. Differently, we adopt a multiple source-target conditional alignment to capture the source-specific knowledge for target learning. Besides, we first introduce dual mix-up regularization to multi-source problem and also add a target distilling mechanism.

3.3 Method

3.3.1 Framework Overview

We propose our method EC-MSA. The architecture of our model is shown in Fig 3.1, which consists of two modules, i.e., the feature extractor group and the classifier group.

Specifically, the feature extractor group is composed of one shared extractor and K domain-specific extractors. We denote the shared one as $F(\cdot)$. The shared one is used for the first stage's primitive alignment, intending to map all the data into a relatively domain-invariant space and accelerate the training process at the same time. It drops most of the extra information that is not helpful for the following steps and reduces the scale of the data largely.

The domain-specific extractors are represented as $\{C_{\mathrm{j}}(\cdot)\}_{\mathrm{j}=1}^{N}$. They are aim to produce domain-specific features with pairs of source and target data. The information coming from domain-specific extractors is very helpful for feature-level centroid alignment work and we will discuss it later. One thing that is unique for our method is that we create separate corridors for source and target in each extractor. To speak concretely, we make the batch normalization processes unshared for data from distinct domain. In such a case, we can ensure the function of being domain specific for extractors of this group.

The classifier group contains N classifiers corresponding to distinct source domains denoted as $\{C_{\mathrm{j}}(\cdot)\}_{\mathrm{j}=1}^{N}$. Usually, the outputs of this group are utilized to calculate classification loss with cross-entropy as:

$$\mathcal{L}_{cls} = \sum_{\mathrm{j}=1}^{N} \mathbb{E}_{(x_{\mathrm{i}}^{s_{\mathrm{j}}}, y_{\mathrm{i}}^{s_{\mathrm{j}}}) \in S_{\mathrm{j}}} \mathcal{L}_{ce}(C_{\mathrm{j}}(G_{\mathrm{j}}(F(x_{\mathrm{i}}^{s_{\mathrm{j}}}))), y_{\mathrm{i}}^{s_{\mathrm{j}}}), \tag{3.1}$$

where $\mathcal{L}_{ce}(\cdot)$ represents the cross-entropy loss function. Commonly categorical regularization is utilized on the output probabilities as prior work does. The reason is that target samples near decision boundaries are more likely to be mis-classified. In other words, the disagreement on different probabilities may lead to worse predictions on these *hard* target samples, especially when we use the average strategy within the classifier group for the final verifica-

tion of the target dataset. Thus, we propose to use the regularized penalty item shown as:

$$\mathcal{L}_{cr} = \mathbb{E}_{x_{\mathrm{i}}^t \in \mathcal{T}}[\mathbb{E}_{\substack{\mathrm{j} \neq p \\ \mathrm{j}, p \in \mathcal{S}}} \mathbf{Dis}(C_{\mathrm{j}}(G_{\mathrm{j}}(F(x_{\mathrm{i}}^t))), C_p(G_p(F(x_{\mathrm{i}}^t))))], \tag{3.2}$$

where $\mathbf{Dis}(\cdot, \cdot)$ represents a canonical $L1$-Norm function. j and p correspond to different source domains.

3.3.2 Adaptive Cross-Domain Alignment

Although class-wise views are quite popular in the domain adaptation community, it is still not easy to align on class-aware sub-domains, especially when facing the multi-source setting owing to two main difficulties.

The first one is that data in one single batch is uncertain. To be concrete, data of distinct classes may not appear at the similar frequencies as the entire dataset, and a lack of data belonging to some classes in one batch is possible. These two phenomena will always cause ill-conditioned class-wise sub-domains and an unstable distribution in the batch thus be harmful to the following alignment work. Although there exists excellent prior work on single-source like [82] and [83], it is still hard to transfer these methods to multi-source problems in our experiments. It is due to the properties of the multi source problem considering so many domains to be considered and a more complex divergence situation. Therefore, doing alignment on the global centroids of all the categories with a moving average strategy like [23] might be a better option, and our results validate its efficiency.

The next to be mentioned is that the target data is unlabeled so we have to use pseudo-labels for class-aware alignment, but several poor-predicted samples in a single batch will influence the class-aware sub-domains a lot. In such a case, it is important to suppress the noisy labels conveyed in the target samples. Our moving average strategy for generating global centroids can play a big role in improving it, as most target samples can be classified correctly and the centroids will not shift a lot even if the wrong-labeled samples exist. However, adaptive cross-domain alignment is just a partial job of what we do. Target distilling mechanism will be applied and this part will be discussed later.

Our class-wise alignment is adaptively operated. Initially, we obtain the global centroids for source j domain ($\forall \mathrm{j} \in S$) as $\{\bar{O}_k^{s_\mathrm{j}}\}_{k=1}^K$ and for target domain as $\{\bar{O}_k^t\}_{k=1}^K$. For the to-be-updated global centroids, we denote them as $\{O_k^{s_\mathrm{j}}\}_{k=1}^K$ and $\{O_k^t\}_{k=1}^K$ correspondingly.

For each mini-batch, the temporary centroids are computed for the source j and target domain using the labels and pseudo-labels and denoted as $\{\hat{O}_k^{s_\mathrm{j}}\}_{k=1}^K$ and $\{\hat{O}_k^t\}_{k=1}^K$. Then we use the moving average strategy to update the global centroids for both the source j ($\forall \mathrm{j} \in S$) and target domain. This strategy could ensure that the old global centroids memorize the information from the past, while the temporary ones provide new knowledge to update without adding too much noise and imbalance:

$$O_k^{s_\mathrm{j}} = \alpha \hat{O}_k^{s_\mathrm{j}} + (1-\alpha)\bar{O}_k^{s_\mathrm{j}}, \tag{3.3}$$

$$O_k^t = \alpha \hat{O}_k^t + (1-\alpha)\bar{O}_k^t, \tag{3.4}$$

where α is a hyper-parameter, set to be 0.3 by default. After several iterations, the global centroids will become relatively stable and no class will be disregarded. They can reflect the actual distribution of the dataset, and this makes sense for the following step.

With the updated centroids, we can do the alignment work to handle the domain shifts across multiple source domains with target one. Maximum Mean Discrepancy (MMD) ([63]) is the metric to estimate the discrepancy between two domains, which is expressed as:

$$\mathcal{L}_{fd} = \sum_{\mathrm{i}=1}^{N} \mathbb{E}_{k\in\mathcal{D}} \left\| \phi(O_k^{s_\mathrm{j}}) - \phi(O_k^t) \right\|_2, \tag{3.5}$$

where $\phi(\cdot)$ denotes a mapping that could project the samples to Reproducing Kernel Hillbert Space(RKHS).

3.3.3 Enhanced Consistency via Dual Mix-up

Original data alone from the target domain are not sufficient for categorical information extraction, so data augmentation is very necessary. Dual mix-up is a very popular method in the community. Work from [25] has proven it to be efficient empirically and [66] shows its

effectiveness theoretically. It can not only smooth the model's outputs so as to encourage the model to have a relatively more strict linear behavior, but also guarantee consistent predictions in the data distributions.

We rearrange the original mini-batch to get a new batch, then assign interpolations between them to form the augmented batch for further training. Provided that x_r^t is the original sample in the initial batch and x_q^t is the corresponding sample at the same location of the new batch after permutation, we get the mix-up one:

$$\tilde{x}_{\mathrm{i}}^t = \mathcal{M}_\lambda(x_r^t, x_q^t) = \lambda x_r^t + (1-\lambda)x_q^t, \tag{3.6}$$

where λ is following a beta distribution $\mathrm{Beta}(\mu, \mu)$ and we set μ as 0.2 by default.

In this sense, we explore the augmented samples to enhance the target prediction consistency over our proposed method EC-MSA as:

$$\mathcal{L}_{dr} = \mathbb{E}_{x_{\mathrm{i}}^t \in T}[\mathbb{E}_{\substack{\mathrm{j}\neq p \\ \mathrm{j}, p \in S}} \mathbf{Dis}(C_{\mathrm{j}}(G_{\mathrm{j}}(F(\tilde{x}_{\mathrm{i}}^t))), C_p(G_p(F(\tilde{x}_{\mathrm{i}}^t))))]. \tag{3.7}$$

Different from [25], we enforce the outputs after passing distinct classifiers to be similar instead of comparing the results between the cases of mixup on the inputs or the outputs. That is because we are dealing with multi-source problems, and the divergence between multiple source domains is the major obstacle.

3.3.4 Pseudo-Label Based Target Distilling

Till now, our current model still suffers from target prediction uncertainty. For the class-wise centroid alignment process, pseudo-labeled sample is used to cluster target data, so the precision of target prediction matters a lot. Although using centroids could partially help it, it is limited since negative samples still play a role in the generation of centroids. For the two regularization processes, poor-predicted samples may transfer harmful categorical information to the model and enlarge the divergence between multiple source domains even more. In such a case, we begin to consider instance-aware distilling to extract qualified target data and exclude poor-predicted target data for training on the basis of the target

samples' output probabilities [84]. And the distilling mechanism will function prior to all the previous processes we have discussed so as to ensure the purity of the target samples fed into the network.

The distillation will be put into effort not at the beginning but after a certain amount of iterations. The reason is that the model is not stable and performing well at the very start, and it may contribute to a high error rate in distilling suitable samples. In our model, the distillation starts at the half of the total number iterations of the whole training process. Specifically, we define the output probability of the sub-network related to the source j domain ($\forall \mathrm{j} \in S$) as $\boldsymbol{p}_{\mathrm{i}}^{\boldsymbol{t}} = C_{\mathrm{j}}(G_{\mathrm{j}}(F(x_{\mathrm{i}}^t)))$.Then, this probability can be transformed into a K-digit vector as $\boldsymbol{p}_{\mathrm{i}}^{\boldsymbol{t}} = \left[p_{\mathrm{i},1}^t, p_{\mathrm{i},2}^t, ..., p_{\mathrm{i},K}^t\right]^\top$.

After these preparations, we filter the target samples with lower confidence by building a target sample selector as:

$$\mathrm{Distill}(x_{\mathrm{i}}^t) = 1[\max_{k \in \mathcal{D}} p_{\mathrm{i},k}^t > \beta] \cdot x_{\mathrm{i}}^t, \tag{3.8}$$

where k is the class, and $\beta \in (0, 1)$ is a threshold to distill a target sample x_{i}^t. Because $1(\cdot)$ is an indicator function, if the content inside the indicator function is true, then x_{i}^t will be distilled for the training processes. Otherwise, this sample will be dropped out. For simplicity, we choose $\beta = 0.5$ to ensure that this target sample has a higher probability belonging to class k than not belonging to this class. Although the selector is very straightforward, it is empirically useful by observation during the training process.

3.3.5 Overall Objective

We have already listed all the loss functions for the training process and could formulate it as a total loss function as:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \gamma(\mathcal{L}_{fd} + \mathcal{L}_{cr} + \mathcal{L}_{dr}), \tag{3.9}$$

where γ is a trade-off hyper-parameter to balance different components in the overall loss function.

3.3.6 Theoretical Insight

Let $\mathcal{H}$ be a hypothesis with a VC dimension of d and m is the size of a mini-batch. Since it is a theoretical analysis for a multi-source domain adaptation method, we need to define domain weights for different source domains and sample ratios for these sources in one single batch. In such a case, we construct a vector $\boldsymbol{\alpha} = \Big[\alpha_1, \alpha_2, .., \alpha_N\Big]^\top$ to be the domain weight vector and another vector $\boldsymbol{\beta} = \Big[\beta_1, \beta_2, .., \beta_N\Big]^\top$ to be the source sample ratio vector. These two vectors should always satisfy $\sum\limits_{\mathrm{j}=1}^{N} \alpha_\mathrm{j} = 1$ and $\sum\limits_{\mathrm{j}=1}^{N} \beta_\mathrm{j} = 1$. For $\mathrm{j} \in \{1, .., N\}$, we assume Q_j to be a set of samples that will be put into a mini-batch with size $\beta_\mathrm{j} m$ from source domain $\mathcal{S}_\mathrm{j}$. The corresponding labeling function is denoted as f_j. We also denote $\hat{h} = \arg\min_{h \in \mathcal{H}} \hat{\epsilon_\alpha}(h)$ and $h^*_\mathcal{T} = \arg\min_{h \in \mathcal{H}} \epsilon_\mathcal{T}(h)$. Till now, we could introduce the general bound for a typical multi-source domain adaptation problem as [7] does. For $\forall \delta \in (0, 1)$, with a confidence of at least $1 - \delta$:

$$\begin{aligned} \epsilon_\mathcal{T}(\hat{h}) \leq \epsilon_\mathcal{T}(h^*_\mathcal{T}) + \frac{2}{N}\sqrt{(\sum_{\mathrm{j}=1}^{N} \frac{\alpha_\mathrm{j}^2}{\beta_\mathrm{j}})(\frac{d\log(2m) - \log(\delta)}{2m})} \\ + \sum_{\mathrm{j}=1}^{N} \alpha_\mathrm{j}(2\lambda_\mathrm{j} + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_\mathrm{j}, \mathcal{T})), \end{aligned} \tag{3.10}$$

where $\lambda_\mathrm{j} = \min_{h \in \mathcal{H}}\{\epsilon_\mathcal{T}(h) + \epsilon_\mathrm{j}(h)\}$.

For our method EC-MSA, we choose an average ensemble strategy. In other words, all data instances are equally weighted and the weights for all training errors of different domains are the same. Therefore, we deduce a specific bound, still for $\forall \delta \in (0, 1)$, with a confidence of at least $1 - \delta$:

$$\begin{aligned} \epsilon_\mathcal{T}(\hat{h}) \leq \epsilon_\mathcal{T}(h^*_\mathcal{T}) + \frac{2}{N}\sqrt{(\frac{d\log(2m) - \log(\delta)}{2m})} \\ + \frac{1}{N}\sum_{\mathrm{j}=1}^{N} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_\mathrm{j}, \mathcal{T})) + \frac{2}{N}\sum_{\mathrm{j}=1}^{N} \lambda_\mathrm{j}. \end{aligned} \tag{3.11}$$

Now we need to switch the attention to the right side of this inequality. The first term and the third one depends mainly on the hypothesis space, while the second term is determined

by the batch size m and the confidence gate δ. Under these conditions, we seek to minimize the fourth term as following:

$$\begin{aligned}
\lambda_{\mathrm{j}} &= \min_{h\in\mathcal{H}} \epsilon_{\mathcal{T}}(h) + \epsilon_{\mathrm{j}}(h) \\
&= \min_{h\in\mathcal{H}} \epsilon_{\mathcal{T}}(h, f_{\mathcal{T}}) + \epsilon_{\mathrm{j}}(h, f_{\mathrm{j}}) \\
&\leq \min_{h\in\mathcal{H}} \epsilon_{\mathcal{T}}(h, f_{\mathrm{j}}) + \epsilon_{\mathcal{T}}(f_{\mathrm{j}}, f_{\mathcal{T}}) + \epsilon_{\mathrm{j}}(h, f_{\mathrm{j}}) \\
&\leq \min_{h\in\mathcal{H}} \epsilon_{\mathcal{T}}(h, f_{\mathrm{j}}) + \epsilon_{\mathcal{T}}(f_{\mathrm{j}}, f_{\hat{\mathcal{T}}}) + \epsilon_{\mathcal{T}}(f_{\hat{\mathcal{T}}}, f_{\mathcal{T}}) + \epsilon_{\mathrm{j}}(h, f_{\mathrm{j}}).
\end{aligned} \tag{3.12}$$

The first term and the last term on the inequality's right side represent the differences between h and f_{j}. Generally, it's not difficult to find such an $h \in \mathcal{H}$ to approximate f_{j}, so here we mainly focus on the rest two terms.

For $\epsilon_{\mathcal{T}}(f_{\mathrm{j}}, f_{\hat{\mathcal{T}}})$, we can use such a way to approximate it under the background of deep learning:

$$\begin{aligned}
\epsilon_{\mathcal{T}}(f_{\mathrm{j}}, f_{\hat{\mathcal{T}}}) &= \mathbb{E}_{x\in\mathcal{T}} f_{\mathrm{j}}(x) - f_{\hat{\mathcal{T}}}(x) \\
&= \mathbb{E}_{x\in\mathcal{T}} G_{\mathrm{j}} \circ F(x) - G_{\hat{\mathcal{T}}} \circ F(x).
\end{aligned} \tag{3.13}$$

By doing class-wise centroid alignment, for any given class $k \in \mathcal{D}$, we wish to have target features in the same class to be similar with source j centroid of class k, hence this will help to reduce this item effectively.

As for $\epsilon_{\mathcal{T}}(f_{\hat{\mathcal{T}}}, f_{\mathcal{T}})$, it measures the differences between the real hypothesis h and the empirical one $\hat{h}$. In other words, the precision of the pseudo-labels is reflected on this term. By using a target sample distilling mechanism, we can restrict it more than previous methods, which tend to ignore this item.

3.4 Experiments and Analyses

We evaluate our method by performing experiments on three standard benchmarks including **Office-31**, **Office-Home** and **ImageCLEF-DA**, and compare it with state-of-the-art multi-source domain adaption methods to our knowledge. Then certain empirical analyses have been applied to demonstrate the effectiveness of our model EC-MSA.

3.4.1 Datasets

3.4.1.1 Office-31

Office-31 ([85]) is a standard benchmark for domain adaptation in computer vision, containing 4,652 images and 31 categories from three different domains: 2,817 images in the *Amazon* (**A**) domain from amazon.com, 498 images in the *Webcam* (**W**) domain taken by web camera and 795 images in the *DSLR* (**D**) domain from digital SLR camera. These three domains are under different settings and images in each domain are unbalanced. Here we try to evaluate three multi-source domain adaptation tasks: {**A**, **D**}→**W**; {**A**, **W**}→**D**; {**D**, **W**}→**A**.

3.4.1.2 ImageCLEF-DA

ImageCLEF-DA[1] comes from the ImageCLEF 2014 domain adaptation challenge. It includes three domains, with each one made up of 600 images and 12 categories: *Caltech-256* (**C**), *ImageNet ILSVRC 2012* (**I**) and *Pascal VOC 2012* (**P**). It is a very balanced dataset. Three tasks are needed for this dataset: {**C**, **I**} → **P**; {**C**, **P**} → **I**; {**I**, **P**} → **C**.

3.4.1.3 Office-Home

Office-Home ([86]) is a more challenging dataset for the multi-source domain adaptation problem, consisting of 15588 images with 65 classes from four different domains: 2,427 images in the *Artistic* (**Ar**) domain, 4,365 images in the *Clip-Art* (**Cl**) domain, 4,439 images in the *Product* (**Pr**) and 4,357 images in the *Real-World* (**Rw**). All the images are from the office and home setting. We build four tasks to test our method by leaving-one-out as the target domain: {**Ar**, **Cl**, **Pr**}→**Rw**; {**Ar**, **Cl**, **Rw**}→**Pr**; {**Ar**, **Pr**, **Rw**}→**Cl**; {**Cl**, **Pr**, **Rw**}→**Ar**.

[1]↑https://www.imageclef.org/2014/adaptation

Table 3.1. Multi-Source Domain Adaptation Accuracy(%) on Office-31 Dataset

Standards	Method	D	W	A	Avg
Single Best	Resnet	99.3	96.7	62.5	86.2
	DDC	98.2	95.0	67.4	86.9
	DAN	99.5	96.8	66.7	87.7
	D-CORAL	99.7	98.0	65.3	87.7
	RevGrad	99.1	96.9	68.2	88.1
	RTN	99.4	96.8	66.2	87.5
Source Combine	DAN	99.6	97.8	67.6	88.3
	D-CORAL	99.3	98.0	67.1	88.1
	RevGrad	99.7	98.1	67.6	88.5
Multi-Source	DCTN	99.3	98.2	64.2	87.2
	MFSAN	99.5	98.5	72.7	90.2
	SImpAl$_{50}$	99.2	97.4	70.6	89.0
	KD3A	99.8	98.7	71.0	89.8
	MSCLDA	99.8	**98.8**	73.7	90.8
	MSDTR	99.7	98.3	75.2	91.1
	DECISION	99.6	98.4	75.4	91.1
	Ours	**100**	98.3	**76.8**	**91.7**

3.4.2 Implementation Details

3.4.2.1 Network Settings

We use PyTorch ([75]) to implement the network with ResNet50 ([1]) as the backbone. Our network architecture is almost the same as [35]. The shared part of the feature extractor group inherits completely from ResNet50, and the domain-specific part is composed of the structure (conv(1×1), conv(3×3), conv(1×1)), the same for all source domains. After that, the number of channels are reduced to 256 connected to the classifier group, and each member in this group is just a single fully-connected layer. The difference is that in the domain-specific part, we apply separate batch normalization layers for source and target data, ensuring these layers to be specific for each domain.

Table 3.2. Multi-Source Domain Adaptation Accuracy(%) on Image-CLEF Dataset

Standards	Method	P	C	I	Avg
Single Best	Resnet	74.8	91.5	83.9	83.4
	DDC	74.6	91.1	85.7	83.8
	DAN	75.0	93.3	86.2	84.8
	D-CORAL	76.9	93.6	88.5	86.3
	RevGrad	75.0	96.2	87.0	86.1
	RTN	75.6	95.3	86.9	85.9
Source Combine	DAN	77.6	93.3	92.2	87.7
	D-CORAL	77.1	93.6	91.7	87.5
	RevGrad	77.9	93.7	91.8	87.8
Multi-Source	DCTN	75.0	95.7	90.3	87.0
	MFSAN	79.1	95.4	93.6	89.4
	SImpAl$_{50}$	77.5	93.3	91.0	87.3
	KD3A	79.0	95.3	93.2	89.2
	MSCLDA	79.5	95.9	**94.3**	89.9
	Ours	**79.8**	**96.5**	**94.3**	**90.3**

3.4.2.2 Parameter Settings

There are several parameters related to the optimizer to be considered. We use mini-batch Stochastic Gradient Descent (SGD) ([87]) together with a momentum of 0.9 and a weight decay of 0.0005. However, the learning rate annealing strategy is not applied as [13] does but a fixed value $\eta = 0.01$. The reason why we keep the learning rate unchanged is that we use centroids to compute the main discrepancy, and these centroids remain stable during the training, while the non-class-wise alignment uses the varied mini-batch data directly, and a dynamic learning rate is more helpful to adapt the optimization process. Same as [35], we set the learning rate of the shared generator part $\eta_0 = 0.1\eta$. The only dynamic parameter in our method γ in Equation (3.9) is formulated as $\gamma = \frac{2}{1+\exp(-\theta p)} - 1 \in [0, 1)$, where θ is fixed to be 10 and p is the training completion percentage rising from 0 to 1 linearly. With this expression, we can control γ to increase from 0 to near 1 gradually during the training. Further analysis about parameter sensitivity and selection will be presented later.

Table 3.3. Multi-Source Domain Adaptation Accuracy(%) on Office-Home Dataset

Standards	Method	Ar	Cl	Pr	Rw	Avg
Single Best	ResNet	65.3	49.6	79.7	75.4	67.5
	DDC	64.1	50.8	78.2	75.0	67.0
	DAN	68.2	56.5	80.3	75.9	70.2
	D-CORAL	67.0	53.6	80.3	76.3	69.3
	RevGrad	67.9	55.9	80.4	75.8	70.0
Source Combine	DAN	68.5	59.4	79.0	82.5	72.4
	D-CORAL	68.1	58.6	79.5	82.7	72.2
	RevGrad	68.4	59.1	79.5	82.7	72.4
Multi-Source	MFSAN	72.1	62.0	80.3	81.8	74.1
	SImpAl$_{50}$	70.8	56.3	80.2	81.5	72.2
	MADAN	66.8	54.9	78.2	81.5	70.4
	K3DA	70.7	62.1	81.1	82.8	74.2
	MSCLDA	71.6	61.4	79.9	80.4	73.4
	DECISION	74.5	59.4	**84.4**	83.6	75.5
	MSDTR	73.8	64.6	81.6	83.5	75.9
	Ours	**75.7**	**65.3**	84.1	**84.0**	**77.3**

In the inference stage, the average weights are assigned to distinct classifiers. It is due to the fact that we have strong $L1$ regularizations in the model so that the agreement among multiple classifiers is ensured.

3.4.3 Baselines

We compare with other methods from three mainstream standards. First is the single best reporting the best single-source domain adaptation results. Here **ResNet** ([1]), **DAN** ([11]), **DDC** ([9]), **D-CORAL** ([10]), **RevGrad** ([13]) and **RTN** ([88]) are utilized. Second is the source combine, which combines data from all the source domains into one domain and then do transfer work. This standard includes **DAN** ([11]), **D-CORAL** ([10]) and **RevGrad** ([13]). Final part is the multi-source domain adaptation methods including **DCTN** ([32]), **MFSAN** ([35]), **SImpAl$_{50}$** ([81]), **MADAN** ([78]), **MSCLDA** ([31]), **MSDTR** ([34]), **KD3A** ([42]) and **DECISION** ([39]).

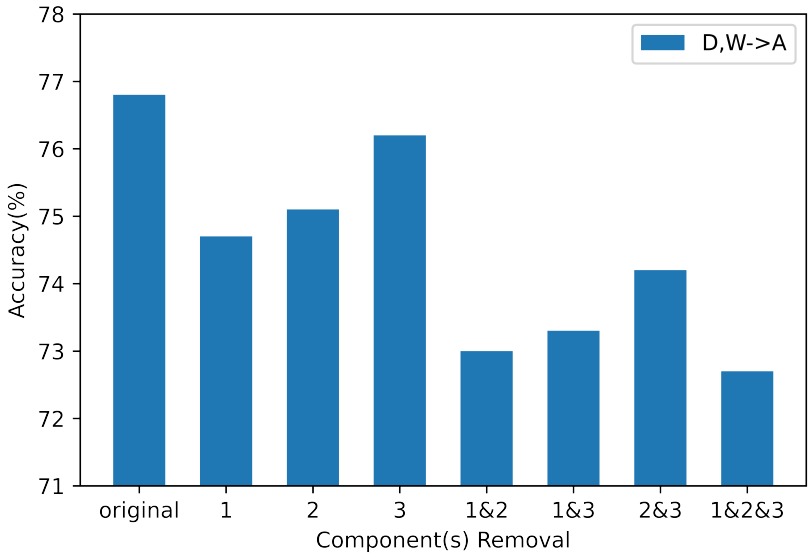


Figure 3.2. Ablation study of our algorithm on task {D,W}→A from Office dataset. There are three components in our model: 1. centroid alignment, 2. mix-up regularization and 3. pseudo-label based distillation. We remove certain component(s) for every situation. (Best viewed in color.)

Note that we *do* keep close watch on the development of the new methods from the multi-source standard, but some classic methods ([29]) or latest methods ([89]) are excluded because of using different backbones and datasets and it is not suitable to make comparisons under distinct settings. All the compared baselines use ResNet50 ([1]) as the backbone and follow the general protocol similar as [35].

3.4.4 Results and Analysis

The comparison results with the referred baselines on these three datasets are reported in Tables 3.1, 3.2 and 3.3. From the results, we have the following observations.

First of all, our proposed method EC-MSA yields state of the art results on almost all the tasks and proves to be effective. In terms of average accuracy, our method outperforms all the other baselines. For Office-31 experiments, the average improvement is about 0.6%. Significantly, we improve the most difficult task {D, W} → A by 1.4%. As for the other two

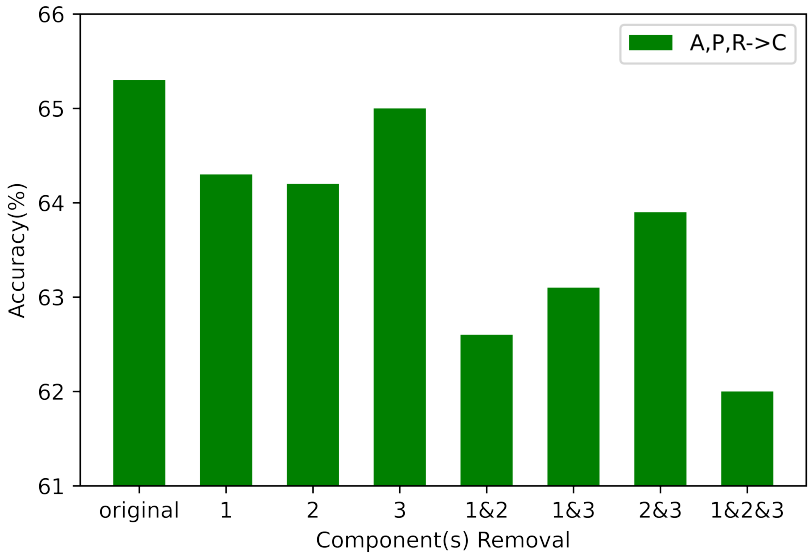


Figure 3.3. Ablation study of our algorithm on task {Ar,Pr,Rw}→Cl from Office-Home dataset. (Best viewed in color.)

tasks, due to the fact that D is very close to W, it is hard to make enough progress, so EC-MSA looks similar to other multi-source methods. For Image-CLEF, the average accuracy has been increased by 0.4%. What needs to be mentioned is that our method becomes the first one from the multi-source standard to surpass RevGrad from the single best standard on the task {I, P} → C, which demonstrate that multi-source standard is more practical for multi-source adaptation even if the single-source method is strong enough. On average our method EC-MSA doesn't improve too much as the other two datasets. This is because Image-CLEF is a relatively simple dataset with just 12 classes. Thus, it is not challenging for the classifier to discriminate even if the model is not well-designed. When comparing with other methods on Office-Home, we maintain a lead of 1.4% on average, and achieve a lead of 1.2% on the task {Cl, Pr, Rw} → Ar. Since Office-Home is the largest-scale dataset with more categories among the three, it leaves enough space for EC-MSA to show its advantages. Generally speaking, our model achieves the best average results among all domain adaptation situations we proposed. For all the single tasks, the best results are achieved in almost all

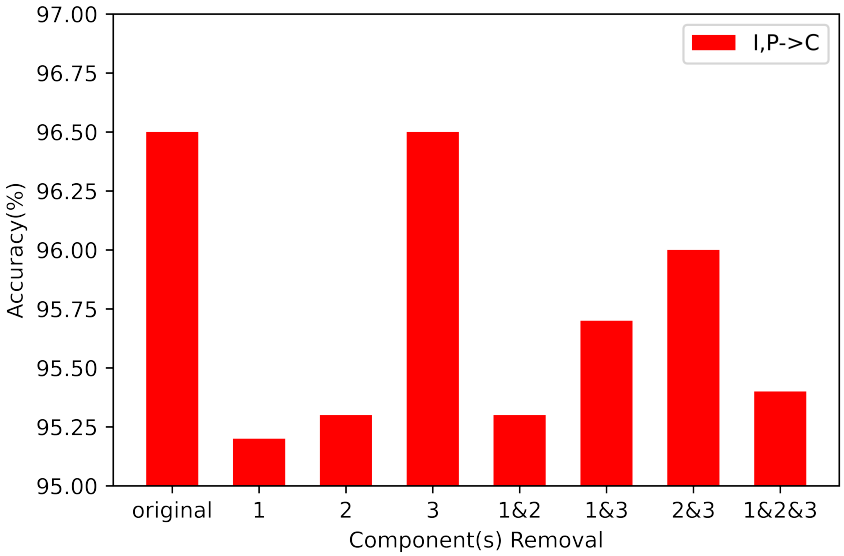
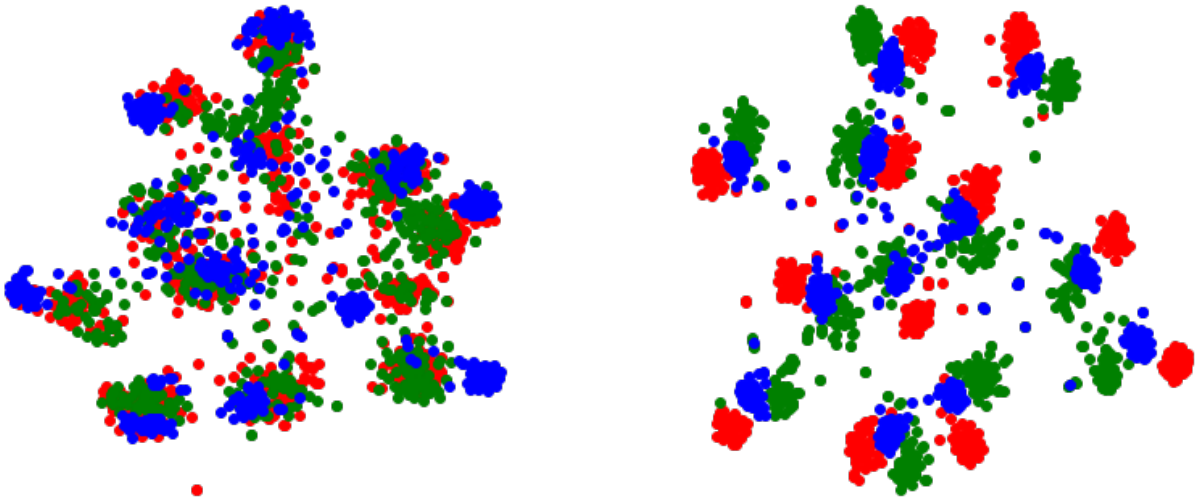


Figure 3.4. Ablation study of our algorithm on task {I,P}→C from ImageCLEF dataset. (Best viewed in color.)

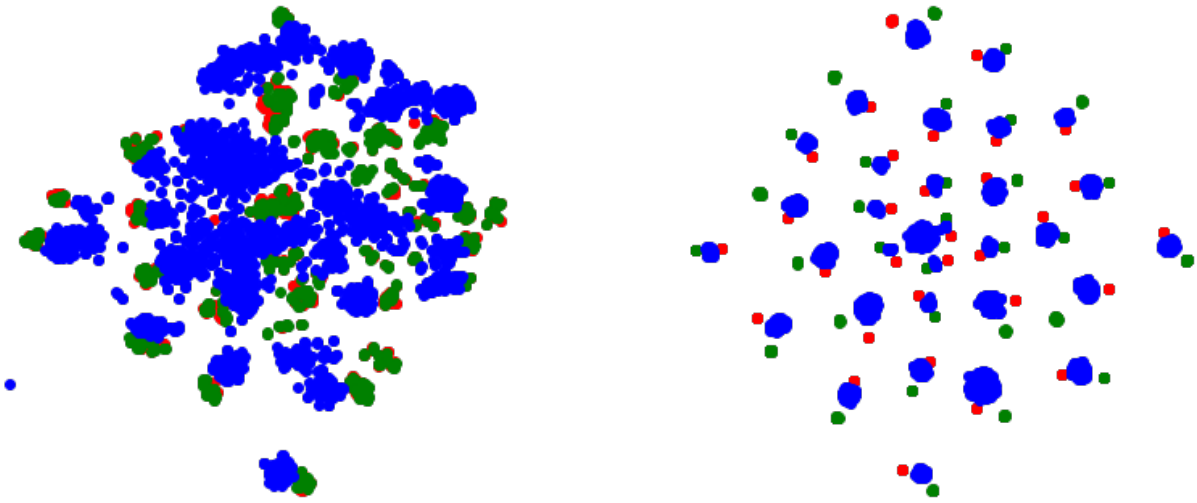
the experiments except the task {A, D} → W and {Ar, Cl, Rw} → Pr. But EC-MSA still gets relatively good results on these two tasks and the disparities are small enough to be accepted. These results show the effectiveness of our proposed model in solving multi-source domain adaption.

Moreover, methods belonging to the single-best standard tend to have the worst performance among three standards on average. It reflects that data from multiple sources can always benefit the training process and the inference stage. Therefore, it is necessary in the practical scenarios to collect data with distinct sources and do research on specialized multi-source domain adaptation. Besides, methods subordinating to the multi-source standard perform better than models from the source combine standard. That is because domain shift also exists across distinct source domains, and simply regarding all the sources as the same will confuse the classifier. In such a case, it is essential to develop networks supporting multiple sources with multiple substructures, which is what EC-MSA attempts to do.



(a) I,P→C before adaptation (b) I,P→C after adaptation

Figure 3.5. Visualization with t-SNE on task I,P→C from ImageCLEF-DA dataset (best viewed in color). **a**: t-SNE before adaptation. **b**: t-SNE after adaptation.



(a) D,W→A before adaptation (b) D,W→A after adaptation

Figure 3.6. Visualization with t-SNE on task D,W→A from Office-31 dataset (best viewed in color). **a**: t-SNE before adaptation. **b**: t-SNE after adaptation.

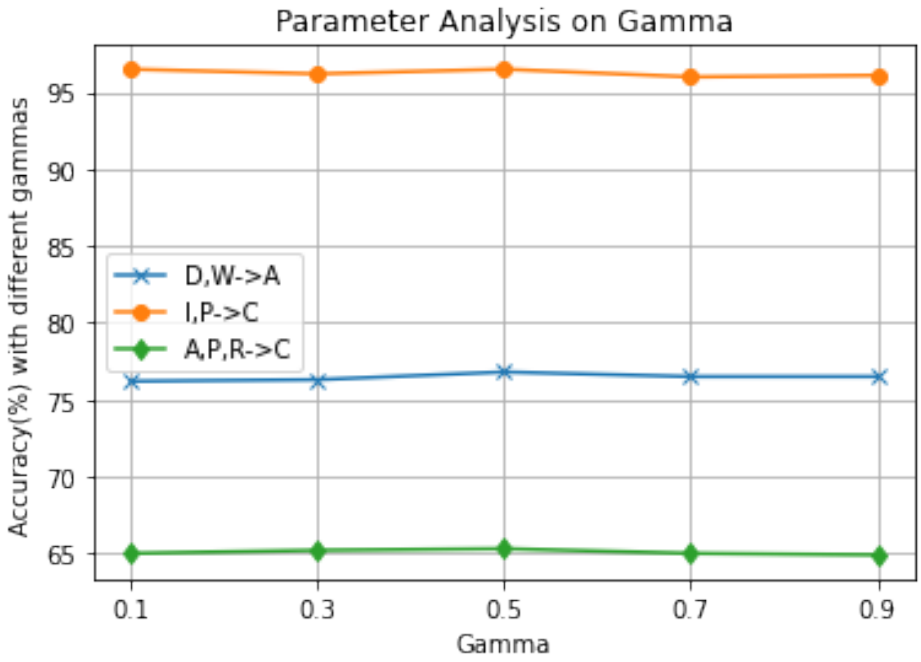


Figure 3.7. Parameter analysis of our algorithm on γ. (Best viewed in color.)

3.4.5 Empirical Analysis

3.4.5.1 Ablation Study

We choose one typical task from each dataset to do ablation test. There are three components in our model: 1) centroid alignment, 2) mix-up regularization and 3) pseudo-label based distillation. Hence there exist eight combinations for our ablation study: original, without 1, without 2, without 3, without 1&2, without 1&3, without 2&3 and without 1&2&3. Three tasks are selected for this analysis as {D, W} $\rightarrow$ A in Figure 3.2, {Ar, Pr, Rw} $\rightarrow$ Cl in Figure 3.3, and {I, P} $\rightarrow$ C in Figure 3.4.

From these bar charts, we can see that the removal of any of these three variants will cause a reduction to the accuracies directly. What's more, it is clear that centroid alignment and dual mix-up play a more prominent role than distillation, and centroid alignment outweighs dual mix-up in improving the model's performance. As for the task {I, P} $\rightarrow$ C (Figure 3.4), the distilling process seems not functional. It may be due to the fact that the tasks

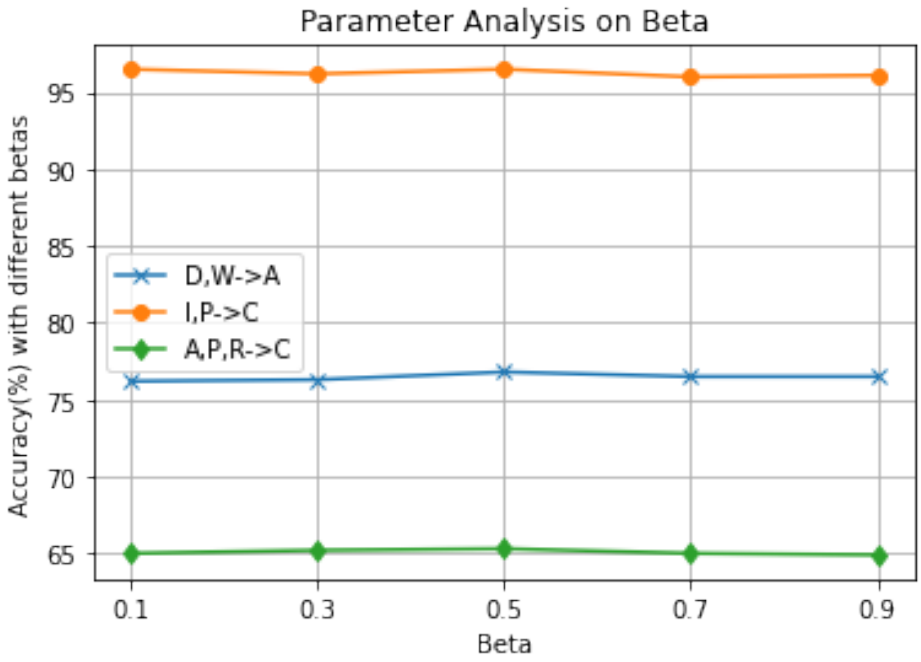


Figure 3.8. Parameter analysis of our algorithm on β. (Best viewed in color.)

in the ImageCLEF-DA dataset only consider 12 categories, which is relatively small. Hence it is much easier for a target sample to get a concentrated distribution and go through the selector. In such a case, most of the target examples are distilled and the pseudo-label based selector doesn't serve as a filter successfully. But from the other two tasks we can see, when the dataset has a larger label space, it will definitely help the model to improve the accuracy.

3.4.5.2 Embedding Visualization

To further understand the alignment of distribution, we visualize features produced by the feature extractor group. Here t-SNE ([90]) is used, which is a very popular data visualization tool. For the source domain data, we use the results from the corresponding feature extractor. But for the target data, we take the average embedding derived from distinct outputs. Two tasks {I, P}→C and {D, W}→A are chosen to verify our model and the results are shown in Figure 3.5 and Figure 3.6 separately. We use the ResNet ([1]) from the single best standard to compare. It can be seen that our model make the features more discriminative and

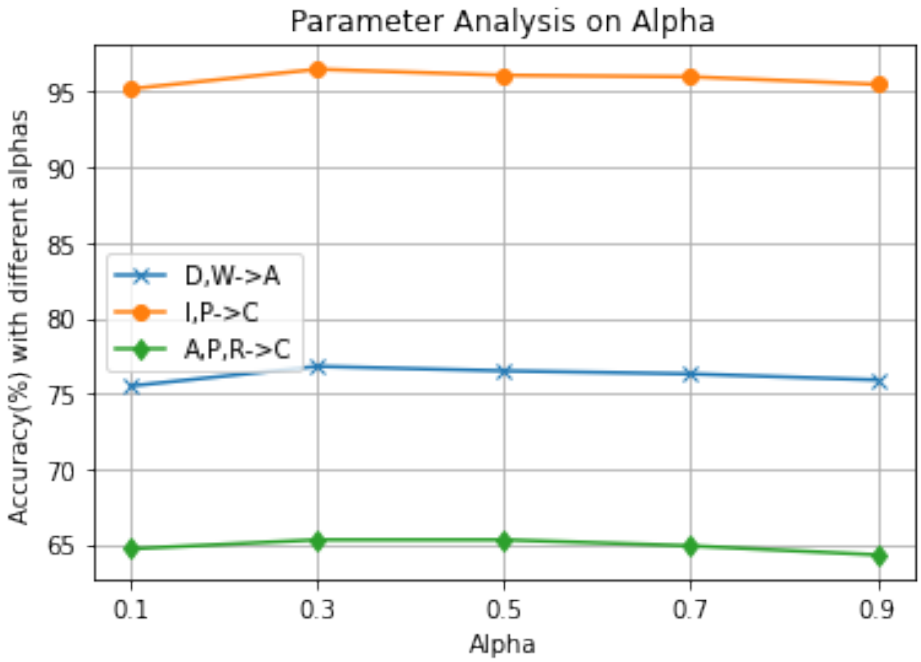


Figure 3.9. Parameter analysis of our algorithm on α. (Best viewed in color.)

separate better than the non-adapted situation. Besides, we can see that for each cluster, data from three domains connect tightly with each other after adaptation, especially for the task {D, W}→A (Figure 3.6). It can also illustrate the efficiency of our work on enhancing consistency among distinct domains and agreement on corresponding classifiers.

3.4.5.3 Parameter Analysis

We conduct experiments to investigate the sensitivity of our method to three hyper-parameters γ, β and α. γ is the balance parameter for the total loss function, while β is the threshold for the target distilling mechanism and α is the ratio of iterative moving average process for the centriods' computation. The results are displayed in Figure 3.7 for γ, Figure 3.8 for β and Figure 3.9 for α. For the trade-off parameter γ in our objective function, we test $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. Here β is fixed as 0.5 and α is fixed as 0.3. For the threshold parameter in the distilling mechanism, we sample the values in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$, while set γ to be 0.5 and α to be 0.3. For the moving average ratio parameter α, we use the same

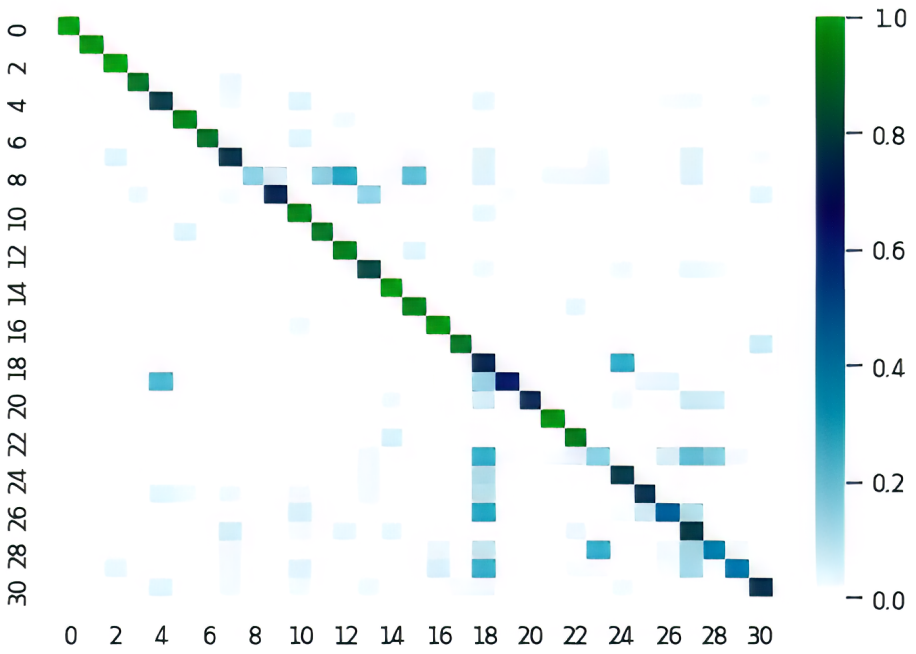


Figure 3.10. Confusion matrix of the predicted results of our algorithm on task D,W→A from ImageCLEF dataset. (Best viewed in color.)

range as above while choose the default value of the other two as 0.5. We can see that our method is not very sensitive to parameters. By observation, 0.5 for γ, 0.5 for β and 0.3 for α should be the best choice. Besides, we can see that the curves are relatively stable, indicating that EC-MSA is robust enough so that slight modifications on hyper-parameters won't have an obvious impact on its final performances.

3.4.5.4 Confusion Matrices Visualization

We choose two typical tasks: {D,W}→A and {I,P}→C. In our experiments they could achieve accuracies of 76.8% and 96.5% separately. Then two confusion matrices have been built to show the effectiveness, Figure 3.10 for the task {D,W}→A and Figure 3.11 for the task {I,P}→C. In each confusion matrix of Figure 3.10 and Figure 3.11, the row represents the predicted results while the column denotes the ground truth. The color band on the right side of each matrix illustrates the percentage of class i regarded as class j in the target

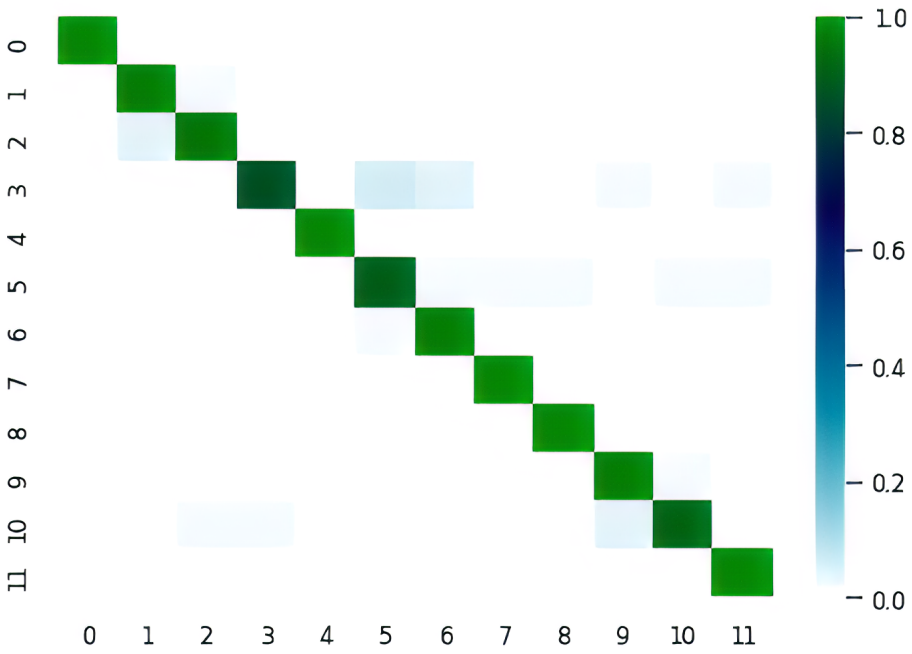


Figure 3.11. Confusion matrix of the predicted results of our algorithm on task I,P→C from ImageCLEF dataset dataset. (Best viewed in color.)

domain. If the ratio is higher, than the color should be brighter. As for the task {I,P}→C, since it's a relatively simple task, we can see that the bright blocks are concentrated on the diagonal. For the other one, the task itself is more difficult and the diagonal blocks are not so bright. But we still can find that the predictions are even, with just two out of 31 categories are under 0.4. It shows the efficiency of our class-wise alignment strategy with centroids.

3.4.6 Conclusion

In this chapter, we introduce our multi-source domain adaptation approach EC-MSA, trying to improve it in three ways. By using centroid alignment, dual mix-up and target distilling, we obtain a better model, which can be proved by the following analyses theoretically and empirically.

4. FOURIER TRANSFORM-ASSISTED FEDERATED DOMAIN ADAPTATION

4.1 Preliminaries and Motivations

For a typical federated domain adaptation problem, there is one source domain, where $\mathcal{S} = \{(x_{\mathrm{i}}^s, y_{\mathrm{i}}^s)\}_{\mathrm{i}=1}^{n_s}$ represents the dataset of source domain with n_s labeled samples. Besides, there exists a target domain dataset $\mathcal{T} = \{x_{\mathrm{i}}^t\}_{\mathrm{i}=1}^{n_t}$ that includes n_t unlabeled samples. The source and target domains share the same label space but lie in different distributions. The label space is denoted as $\mathcal{D} = \{1, 2, ..., K\}$, while K is the number of classes. In such a case, the goal is to seek a good classification model that could achieve a high accuracy on the target domain. Since it's federated domain adaptation, any communications between the raw data of these two domains are completely forbidden.

The motivation of our work is that we hope to bridge these two domains so as to make alignments and reduce discrepancies even under the federated setting. Work from [62] did receive a great success, but this kind of hypothesis transfer learning setting seems to treat federated domain adaptation as one supervised learning problem on source domain and one self-supervised learning problem on target domain. In such a case, an explicit process of knowledge transfer or domain adaptation doesn't exhibit. Although the exchange or transfer of raw data is forbidden, there still exist other approaches that might help the source and target domain to get in touch, and frequency domain's assistance could be one of the choices. That's the motivation of our work.

4.2 Related Work

Due to the emphasis on privacy and security, the community pays more and more attention to the federated learning, which is a machine learning setting that ensures the local storage of raw data. From [37], federated learning started to connect with domain adaptation. This paper replaces the original data exposure with gradient, together with dynamic attention mechanisms and feature disentanglement to enhance the knowledge transfer between

different domains. There also exists work like [27] that exposes frequency space information instead of raw data among different domains.

Source-free adaptation is another paradigm for federated domain adaptation. It regards the domain adaptation problem as a hypothesis transfer problem. [62] proposes source-free domain adaptation, in case which sample-based exposures are completely forbidden and only model parameters can be exchanged among multiple clients. Besides, this paper introduced mutual information maximization ([91]) to deal with hypothesis transfer. Follow [62], a lot of work has been proposed, like [39] which extends [62] to the multi-source problem. [92] combines source-free domain adaptation together with attention mechanism. It also adopts local structure clustering to assist knowledge transfer. [40] applies contrastive learning to the federated adaptation problem, and also uses generated prototypes to make alignments. [41] utilizes GAN to generate pseudo source data so that it can help align between source and target. [93] uses model's weights as source prototypes to solve the source-free situation.

Here we adopt the first paradigm, which sample-based information is allowed to communicate among different domains while raw data's communications are totally forbidden. Our model FTA-FDA uses Fast Fourier Transformation to transfer the image information to the frequency space and only part of the amplitude images are utilized.

4.3 Method

4.3.1 Framework Overview

Here we still apply ResNet50 ([1]) as the backbone, while the last fully connected layer is replaced by two fully connected blocks together with tricks like ReLU ([49]) and Dropout ([50]). The overall two stages have been arranged as most federated work does. First, we train a model on the labeled source data. Then the model will be moved forward to the client of target domain, together with information from the frequency spectra of the source domain. Based on these, the training on the target domain starts and after certain iterations we get the final model. One thing needs to be mentioned is that in the source training stage, all parts of the network will be optimized and updated, but in the target training stage, the last fully-connected block will be fixed while the rest can still be optimized and updated.

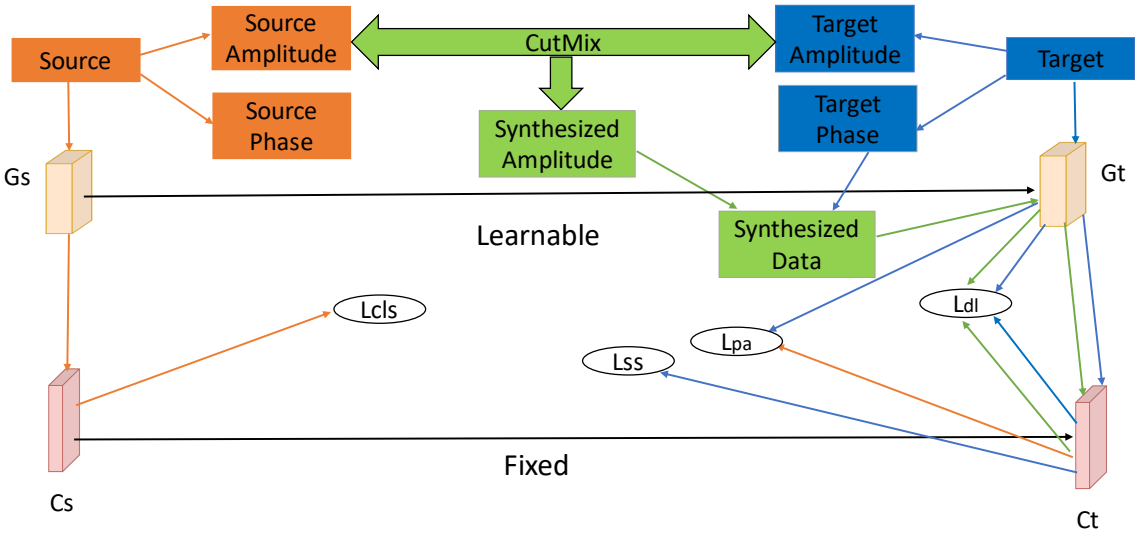


Figure 4.1. Overview of our proposed Fourier Transform-Assisted Federated Domain Adaptation (FTA-FDA), which contains the source model $f_s = G_s \circ C_s$ and the target model $f_t = G_t \circ C_t$. The left side is operated on the source client while the right side is operated on the target client. (Best viewed in color.)

Figure 4.1 shows the overall process of our method FTA-FDA. The left side is about training on source and the right side is about training on target. All the things passing the central line is related to the communication between these two clients.

4.3.2 Source Model Training

In this section, we will talk about the training stage on source domain. The most common technique to deal with labeled data is the cross-entropy loss:

$$\mathcal{L}_{cls} = -\mathbb{E}_{(x_i^s, y_i^s)\in\mathcal{S}} \sum_{k\in D} y_i^s \log softmax(f_s(x_i^s)) \tag{4.1}$$

Here f_s is the model we want to learn, and y_i^s is the one-hot encoded version of the label. Since the training stages of source and target are relatively separate, it is very important to keep the model's ability of generalization and avoid overfitting greatly on the supervised

learning, especially when we fix certain parts of the model for the target training stage. Therefore, label smoothing ([94]) is really helpful here. Assume a smoothed label like:

$$l_{\mathrm{i}}^{s} = \frac{\alpha}{K} + (1-\alpha) y_{\mathrm{i}}^{s}, \tag{4.2}$$

where α is the smoothing parameter and usually set to be 0.1. Then we can deduce the new version of the loss used on source data as:

$$\mathcal{L}_{cls} = -\mathbb{E}_{(x_{\mathrm{i}}^{s}, y_{\mathrm{i}}^{s}) \in \mathcal{S}} \sum_{k \in D} l_{\mathrm{i}}^{s} \log softmax(f_s(x_{\mathrm{i}}^{s})). \tag{4.3}$$

4.3.3 Self-Supervised Target Model Training

Before connecting the two domains, some prior work is necessary with the assistance of self-supervised learning methods. Here we introduce two kinds of loss functions, information maximization and pseudo supervised learning.

For a better explanation of the following content, we import the notation of one target sample's output probability as:

$$p_{\mathrm{i}}^{t} = softmax(f_t(x_{\mathrm{i}}^{t})), \tag{4.4}$$

where f_t is the model used on the target domain. After that, the information maximization item can be represented as:

$$\mathcal{L}_{IM} = -\mathbb{E}_{x_{\mathrm{i}}^{t} \in \mathcal{T}} \sum_{k \in D} p_{\mathrm{i}}^{t} \log p_{\mathrm{i}}^{t} + \sum_{k \in D} \bar{p}^{t} \log \bar{p}^{t}. \tag{4.5}$$

Here the first item of Equation 4.5 aims at minimizing the entropy for each sample, while the second item tends to promote the diversity of the whole batch. $\bar{p}^{t}$ is the average of all the output probabilities in one single batch.

For the pseudo supervised learning part, it is quite simple. What differs our work from [62] is that we do not utilize the deep clustering strategy for acquiring the pseudo labels on the target domain, because from our empirical experiments, we observe that this strategy

is harmful to the model's performance. Here the predictions of previous models are used directly as:

$$\mathcal{L}_{ps} = -\mathbb{E}_{(x_{\mathrm{i}}^t, \hat{y}_{\mathrm{i}}^t)\in\hat{\mathcal{T}}} \sum_{k\in D} \hat{y}_{\mathrm{i}}^t \log p_{\mathrm{i}}^t, \tag{4.6}$$

where $\hat{y}_{\mathrm{i}}^t$ is the pseudo label and $\hat{T}$ is the pseudo target domain that includes pseudo labels. We combine Equation 4.5 and Equation 4.6 and get the self-supervised loss as:

$$\mathcal{L}_{ss} = \mathcal{L}_{IM} + \mathcal{L}_{ps}. \tag{4.7}$$

4.3.4 Frequency Domain Interpolation

Fourier Transform has been introduced to the field of domain adaptation since [95], then [27] applied it under the federated learning setting. These two papers focus on the perspectives of semantic segmentation, while our work aims at improving visual object recognition.

First, we introduce the standard format of Fourier transform. Assume an input sample as $x_{\mathrm{i}}^t \in \mathbb{R}^{C\times H\times W}$, where C, H and W correspond to channel numbers, height and width. Different from what we refer to in the background chapter, here it transfers data from space domain to frequency domain:

$$\mathcal{F}_{\mathrm{i}}^t = \sum_{h=0}^{H-1}\sum_{w=0}^{W-1} x_{\mathrm{i}}^t \exp\left[-\mathrm{j}2\pi(\frac{h}{H}u + \frac{w}{W}v)\right]. \tag{4.8}$$

Here we ensure the one-to-one correspondence between channels of different domains, while x_{i}^t is a function of (h, w), the unit of an RGB image, and $\mathcal{F}_{\mathrm{i}}^t$ is a function of (u, v), the unit of an frequency space image.

Of course, we cannot transfer the frequency domain image to a different client because it can be completely recovered as the original space domain image. Fortunately, the frequency space image can be further decomposed as:

$$\mathcal{F}_{\mathrm{i}}^t = \mathcal{A}_{\mathrm{i}}^t \exp\left(\mathcal{P}_{\mathrm{i}}^t\right) \tag{4.9}$$

where $\mathcal{A}_{\mathrm{i}}^t \in \mathbb{R}^{C\times H\times W}$ is the amplitude spectrum and $\mathcal{P}_{\mathrm{i}}^t \in \mathbb{R}^{C\times H\times W}$ is the phase spectrum.

According to the work from [95] and [27], the amplitude spectrum reflects the low-level distributions like style and the high-level semantics like object shape are stored in the phase spectrum. Since our task is domain adaptation, we hope that the model could transfer the style information from the source domain to the target domain, while keep the object shape relatively stable. Therefore, it's natural for us to choose amplitude part as the information that will be transferred.

Even if only the amplitude spectrum will be transferred, people may still worry about the leak of client's information during the process. We try to alleviate this issue in two ways. One is that we will show that the amplitude and phase spectra are neither discriminative nor transferable when we train on a deep neural network, which is represented later. The other is that we just crop a very little ratio of the amplitude spectrum, which includes very little information from one source sample. Assume the ratio is α, then we build a mask as:

$$\mathcal{M} = 1(h, w) \in [\frac{1-\alpha}{2}H : \frac{1+\alpha}{2}H, \frac{1-\alpha}{2}W : \frac{1+\alpha}{2}W]. \tag{4.10}$$

After that, the interpolation can be represented as:

$$\mathcal{A}_{\mathrm{i}}^{t \to s} = \mathcal{M} * \mathcal{A}_{\mathrm{j}}^{s} + (1 - \mathcal{M}) * \mathcal{A}_{\mathrm{i}}^{t}, \tag{4.11}$$

where $\mathcal{A}_{\mathrm{j}}^{s}$ is a random spectrum from the source domain. Here we set $\alpha = 0.1$ and only the selected area will be transferred, which occupies only 1% of the whole amplitude spectrum. You can also regard the whole interpolation as CutMix ([96]), like what Figure 4.1 shows.

People may consider a big loss of information in the target sample during the interpolation. This can also alleviate by FFTshift ([97]), which is a very common operation in the frequency domain. FFTshift moves all the zeros to the center of the spectrum, and the replacement is also focused on the center of the spectrum. After interpolation, inverse FFTshift is able to move the spectrum back to the original status. Therefore, most semantic information remains the same after interpolation.

Till now, we can get the synthesized sample by inverse Fourier Transform:

$$x_{\mathrm{i}}^{t \to s} = \mathcal{F}^{-1}(\mathcal{A}_{\mathrm{i}}^{t \to s}, \mathcal{P}_{\mathrm{i}}^{t}). \tag{4.12}$$

After obtaining the synthesized sample, we can compute the discrepancy between these two domains. Denote the hypothesis as $f_t = G_t \circ C_t$, where G_t is the generator and C_t is the classifier, then the discrepancy loss is:

$$\mathcal{L}_{dl} = \left\| \mathbb{E}_{x_{\mathrm{i}}^t \in T}[\phi(G_t(x_{\mathrm{i}}^t))] - \mathbb{E}_{x_{\mathrm{i}}^{t\to s} \in (T\to S)}[\phi(G_t(x_{\mathrm{i}}^{t\to s}))] \right\|_{\mathcal{H}}^2 + \mathbb{E}_{x_{\mathrm{i}}^t \in T} \left\| C_t(G_t(x_{\mathrm{i}}^t)) - C_t(G_t(x_{\mathrm{i}}^{t\to s})) \right\|. \tag{4.13}$$

The first item is Maximum Mean Discrepancy and the second item is a $L1$-norm regularizer that assists the first item.

4.3.5 Prototype Alignment

In the previous section, we make alignments between two domains on a sample-level. Generally, class-wise alignment can help improve the performance as well. Since it's a source-private adaptation, we cannot access the raw source data. Inspired by [93], we use the weights of the last layer as a substitute of the source prototypes because this layer is fixed during the second stage's training and contains information completely from the source domain. By making comparisons between the weights and the features, we can reduce the discrepancies in a class manner. Here we still use Maximum Mean Discrepancy as the distance metric:

$$\mathcal{L}_{pa} = \left\| \mathbb{E}_{(x_{\mathrm{i}}^t, \hat{y}_{\mathrm{i}}^t) \in \hat{T}}[\phi(G_t(x_{\mathrm{i}}^T; \hat{y}_{\mathrm{i}}^t))] - \mathbb{E}[\phi(W_s(\hat{y}_{\mathrm{i}}^t))] \right\|_{\mathcal{H}}^2. \tag{4.14}$$

Pseudo labels have been applied in the loss function, which differs our approach from [93]. W_s are all the weights of the last fully-connected layer, and we build pairs with x_{i}^T according to the pseudo labels. Empirical results show that our model FTA-FDA outperforms [93] by an obvious margin, which will be discussed in the next chapter.

4.3.6 Overall Objective

We conclude all the loss functions mentioned before to indicate how the models are trained. For the models from the source client:

$$f_s = G_s \circ C_s = \arg\min_{f_s} \mathcal{L}_{cls}. \tag{4.15}$$

Here G_s represents the part that is still learnable in the next stage, and C_s corresponds to the part that will be fixed. For the models from the target client:

$$G_t = \arg\min_{G_t} \mathcal{L}_{ss} + \mathcal{L}_{dl} + \mathcal{L}_{pa}. \tag{4.16}$$

Finally we get the model as $f_t = G_t \circ C_s$.

4.4 Experiments and Analyses

4.4.1 Dataset and Implementation

We have stated a lot about the datasets and implementation in the previous chapter, so here a simple version will be exhibited here. Here **Office-31** is applied for the evaluation, and ResNet50 is still our backbone. We try to evaluate six domain adaptation tasks: {**A**→**D**}; {**A**→**W**}; {**D**→**A**}; {**D**→**W**}; {**W**→**A**}; {**W**→**D**}.

As mentioned before, the last fully-connected layer is replaced by two fully-connected blocks. The last fully-connected block will be fixed when training on the target data. Besides, we don't have too many hyper-parameters except one that relates to the frequency domain interpolation, which will be discussed later.

4.4.2 Baselines and Comparisons

Here the comparisons include two categories of domain adaptation approaches, the source-access methods and the source-private methods. For the first category, we choose **ResNet** ([1]), **DANN** ([13]), **SAFN** ([98]), **CDAN** ([99]), **SRDC** ([100]), **BNM** ([101]) and **MCC** ([30]). For the second category, we choose **FADA** ([37]), **SFDA** ([102]), **SHOT** ([62]), **SDDA** ([103]), **SoFA** ([104]), **CPGA** ([40]), **VAKDT** ([41]), **Proto-DA** ([93]), **MA** ([105]) and **A2Net** ([106]).

The comparisons of results are shown in Table 4.1. In the source-access part, the best accuracies are highlight with underlines, and in the source-private part, the best accuracies

Table 4.1. Federated Domain Adaptation Accuracy(%) on Office-31 Dataset

Standards	Method	A $\rightarrow D$	A $\rightarrow W$	D $\rightarrow A$	D $\rightarrow W$	W $\rightarrow A$	W $\rightarrow D$	Avg
Source-Access	Resnet	68.9	68.4	62.5	96.7	60.7	99.3	76.1
	DANN	79.7	82.0	68.2	96.9	67.4	99.1	82.2
	SAFN	90.7	90.1	73.0	98.6	70.2	99.8	87.1
	CDAN	92.9	94.1	71.0	98.6	69.3	100	87.7
	BNM	90.3	91.5	70.9	98.5	71.6	100	87.1
	MCC	95.6	95.4	72.6	98.6	73.9	100	89.4
	SRDC	95.8	95.7	76.7	99.2	77.1	100	90.8
Source-Private	FADA	90.3	88.2	72.0	98.7	70.8	99.9	86.7
	SFDA	92.2	91.1	71.0	98.2	71.2	99.5	87.2
	SDDA	85.3	82.5	66.4	99.0	67.7	99.8	83.5
	SoFA	73.9	71.7	53.7	96.7	54.6	98.2	74.8
	SHOT	94.0	90.1	74.7	98.4	74.3	99.9	88.6
	CPGA	94.4	94.1	76.0	98.4	76.6	99.8	89.9
	VAKDT	89.9	91.8	73.9	98.7	72.0	99.9	87.7
	MA	92.7	93.7	75.3	98.5	**77.8**	99.8	89.6
	A2Net	94.5	94.0	**76.7**	**99.2**	76.1	**100**	90.1
	Ours	**95.6**	**95.6**	**76.7**	99.0	76.4	**100**	**90.6**

are highlight with bold format. From the table, we can see that our method reaches state of the art in most tasks and average under the source-private setting. When compared with the best method SRDC under the source-access setting, our method is still very competitive.

4.4.3 Ablation Study

In this work, there are two variants that could have an impact on the performance, whether to use the frequency domain interpolation and whether to do prototype alignment. We denote them as variant 1 (v1) and variant 2 (v2) separately and add ResNet as a baseline to show the effectiveness of each component. The results are shown in Table 4.2.

From Table 4.2, we can see that the two components all play important roles in improving the performance. It looks that frequency domain interpolation outperforms prototype alignment in our method.

Table 4.2. Ablation Study Accuracy(%) of FTA-FDA on Office-31 Dataset

Method	A→ D	A→ W	D→ A	D→ W	W→ A	W→ D	Avg
Resnet	68.9	68.4	62.5	96.7	60.7	99.3	76.1
w\$ov1\&v2$	94.0	90.1	74.7	98.4	74.3	99.9	88.6
w\$ov1$	94.1	94.5	73.0	98.8	74.2	99.8	89.1
w\$ov2$	95.0	95.0	74.9	**99.0**	74.1	**100**	89.7
Ours	**95.6**	**95.6**	**76.7**	**99.0**	**76.4**	**100**	**90.6**

4.4.4 Parameter Sensitivity Study

As mentioned before, the most important parameter we concerned is the ratio of the amplitude spectrum that will be applied in the interpolation. The larger the ratio is, the higher the risk will be brought to the communication processes. Here we choose a set ranging from 0 to 1 as $\{0, 0.1, 0.4, 0.7, 1\}$. The results are represented in Table 4.3.

Table 4.3. Parameter Study Accuracy(%) of FTA-FDA on Office-31 Dataset

Crop Ratio	A→ D	A→ W	D→ A	D→ W	W→ A	W→ D	Avg
0	94.1	94.5	73.0	98.8	74.2	99.8	89.1
0.1	95.6	**95.6**	**76.7**	99.0	76.4	**100**	**90.6**
0.4	**96.0**	95.0	76.4	98.8	76.2	99.8	90.4
0.7	95.0	95.3	75.9	99.0	**76.6**	**100**	90.3
1	95.1	95.5	76.0	**99.1**	76.2	99.8	90.3

From the table above, we can see that our method isn't sensitive to the ratio values since the performance is relatively stable. Besides, when $ratio = 0.1$, the results are competitive enough as it achieves state of the art in most tasks and the average scene. It also proves that a little ratio of amplitude spectrum is enough to introduce domain-specific information. After taking privacy and security into account, 0.1 should be the best choice.

4.5 Conclusion

In this chapter, we introduce our source-private domain adaptation approach FTA-FDA. Our major contributions are Fourier transform-assisted space domain interpolation and prototype alignment. Further analyses show that our method is effective and competitive. Special study about privacy and security indicates that we achieve our goal.

5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this thesis, we focus on two varied settings of unsupervised domain adaptation for object recognition. The two alternatives are multi-source domain adaptation and source-private domain adaptation. One assumes that we have labeled source data from multiple domains while the other assumes that the raw data in the source client cannot be exposed to the target client.

For the multi-source adaptation, we propose a novel method named Enhanced Consistency Multi-Source Adaptation (EC-MSA) which attempts to enhance multi-source consistency for cross-domain learning. The conditional sub-domain alignment technique via centroids is applied for the multi-source problem, which largely narrows the divergence between each pair of the source domain and the target domain. We construct centroids iteratively with a moving average strategy for all the classes of each domain and align on these centroids instead of data in one batch. By using this method, we alleviate two common issues that limit the performance of conditional alignment: the uncertainty of data in one single batch and the negative impact of poor-predicted target samples. Then, we adopt a special data augmentation strategy called dual mix-up to enforce the consistency among different source domains along with distinct classifiers. The original batch has been rearranged to form a new one. Then we assign interpolation between these two batches so as to achieve an augmented batch. By making comparisons between the probability outputs of the original batch and that of the augmented batch, we can encourage the model to have a more strict linear behavior while ensure consistent predictions in the data distribution for the multi-source scenario. What's more, we introduce a pseudo-label based target distilling mechanism to purify the target samples, ensuring that the low-confident data will not have too much impact during the training process. A confidence gate has been built on the basis of the highest digit of the output probability. Though it looks simple, it is very useful by observation. We offer detailed illustrations on the datasets and the implementations with PyTorch. Experiments on three different domain adaptation scenarios prove that our proposed approach could achieve state-of-the-art performance under this topic. Further analyses testify the efficiency and stability

of our proposed model EC-MSA. Ablation study, embedding visualization, parameter analysis and confusion matrices visualization have been applied to demonstrate the properties of our model.

For the source-private adaptation, we propose a novel method named Fourier Transform-Assisted Federated Domain Adaptation (FTA-FDA). It aims at solving the conflict between source information exposure and source data security. Here we use frequency domain interpolation to balance these two concerns. Raw image data from the source client are first transferred to the frequency domain by Fast Fourier Transform (FFT), then we crop only a small proportions of their amplitude spectra to communicate with the target client. After certain interpolations, we build synthesized source images and make alignments with the target images. What's more, prototype alignment is also applied in order to improve the model's performance. We extract the weights of the last fully-connected layer and make conditional alignments with target features. Although this technique looks simple, it is very helpful. Experiments on Office-31 prove the effectiveness and competitiveness of FTA-FDA. Further analyses testify the stability of our proposed approach, especially the research on the security and privacy shows that our method FTA-FDA completely meets the criteria of source-private domain adaptation.

5.2 Future Work

Although we have made significant improvements on the multi-source domain adaptation task and the source-private domain adaptation task, there always exists future work that needs to be done from distinct perspectives to further explore this area.

In the multi-source setting, we can improve the algorithm in three ways. For the conditional alignment, centroids alignment is really computationally expensive and drives the whole training process very slowly. Maybe a new method can be put to achieve good results while save the resource used. For the augmentation strategy, dual mix-up is a very simple way and makes a very shallow interpolation between two batches. It is possible to find a more effective augmentation strategy for the domain adaptation task. Besides, further theoretical work is needed. Though prior work from [66] has shown that dual mix-up is beneficial to

model's generalization, the mechanism of how it helps the domain adaptation remains blank. For the distilling mechanism, a more complex and effective method can be expected. Apart from output probability, approaches based on other confidence gate should be designed to achieve a better distilling function.

In the source-private setting, our algorithm FTA-FDA is able to be improved in two ways. For the frequency domain interpolation, now we can only exhibit that it is beneficial to the adaptation tasks empirically. Further theoretical work to illustrate why this strategy works is necessary. Besides, in FTA-FDA only transfer source amplitude spectra are transferred to the target client. Maybe transferring target amplitude spectra to the source client or even bi-directional transferring can lead better results. For the prototype alignment, though we achieve more advanced performances than the work from [93], the loss value for prototype alignment is relatively large due to the dissimilarity of weights and features. Maybe a assistant neural network can help improve it. Also, theoretical insight for this technique can be a good topic for future work.

REFERENCES

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

[2] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[4] Y. Wu, M. Schuster, Z. Chen, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

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

[6] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, *et al.*, "Analysis of representations for domain adaptation," *Advances in neural information processing systems*, vol. 19, p. 137, 2007.

[7] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1, pp. 151–175, 2010.

[8] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2272–2281.

[9] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[10] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European conference on computer vision*, Springer, 2016, pp. 443–450.

[11] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*, PMLR, 2015, pp. 97–105.

[12] G. Li, G. Kang, Y. Zhu, Y. Wei, and Y. Yang, "Domain consensus clustering for universal domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9757–9766.

[13] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*, PMLR, 2015, pp. 1180–1189.

[14] J. Hoffman, E. Tzeng, T. Park, *et al.*, "Cycada: Cycle-consistent adversarial domain adaptation," in *International conference on machine learning*, PMLR, 2018, pp. 1989–1998.

[15] S. Lee, S. Cho, and S. Im, "Dranet: Disentangling representation and adaptation networks for unsupervised cross-domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 252–15 261.

[16] H. Wu, H. Zhu, Y. Yan, J. Wu, Y. Zhang, and M. K. Ng, "Heterogeneous domain adaptation by information capturing and distribution matching," *IEEE Transactions on Image Processing*, vol. 30, pp. 6364–6376, 2021.

[17] W. Deng, Q. Liao, L. Zhao, *et al.*, "Joint clustering and discriminative feature alignment for unsupervised domain adaptation," *IEEE Transactions on Image Processing*, vol. 30, pp. 7842–7855, 2021.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[19] A. Chadha and Y. Andreopoulos, "Improved techniques for adversarial discriminative domain adaptation," *IEEE Transactions on Image Processing*, vol. 29, pp. 2622–2637, 2019.

[20] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[21] J. Hoffman, M. Mohri, and N. Zhang, "Algorithms and theory for multiple-source adaptation," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 8256–8266.

[22] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *International Conference on Machine Learning*, PMLR, 2015, pp. 1718–1727.

[23] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *International conference on machine learning*, PMLR, 2018, pp. 5423–5432.

[24] S. Zhao, G. Wang, S. Zhang, *et al.*, "Multi-source distilling domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 12 975–12 983.

[25] Y. Wu, D. Inkpen, and A. El-Roby, "Dual mixup regularized learning for adversarial domain adaptation," in *European Conference on Computer Vision*, Springer, 2020, pp. 540–555.

[26] M. Xu, J. Zhang, B. Ni, *et al.*, "Adversarial domain adaptation with domain mixup," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 6502–6509.

[27] Q. Liu, C. Chen, J. Qin, Q. Dou, and P.-A. Heng, "Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1013–1023.

[28] P. Kairouz, H. B. McMahan, B. Avent, *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[29] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1406–1415.

[30] Y. Jin, X. Wang, M. Long, and J. Wang, "Minimum class confusion for versatile domain adaptation," in *European Conference on Computer Vision*, Springer, 2020, pp. 464–480.

[31] K. Li, J. Lu, H. Zuo, and G. Zhang, "Multi-source contribution learning for domain adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[32] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin, "Deep cocktail network: Multi-source unsupervised domain adaptation with category shift," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3964–3973.

[33] A. S. Sebag, L. Heinrich, M. Schoenauer, M. Sebag, L. Wu, and S. Altschuler, "Multi-domain adversarial learning," in *ICLR 2019-Seventh annual International Conference on Learning Representations*, 2019.

[34] L. Zhou, M. Ye, D. Zhang, C. Zhu, and L. Ji, "Prototype-based multisource domain adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[35] Y. Zhu, F. Zhuang, and D. Wang, "Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5989–5996.

[36] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain adaptive ensemble learning," *IEEE Transactions on Image Processing*, vol. 30, pp. 8008–8018, 2021.

[37] X. Peng, Z. Huang, Y. Zhu, and K. Saenko, "Federated adversarial domain adaptation," in *International Conference on Learning Representations*, 2020.

[38] J. Liang, D. Hu, Y. Wang, R. He, and J. Feng, "Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[39] S. M. Ahmed, D. S. Raychaudhuri, S. Paul, S. Oymak, and A. 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*, 2021, pp. 10 103–10 112.

[40] Z. Qiu, Y. Zhang, H. Lin, *et al.*, "Source-free domain adaptation via avatar prototype generation and adaptation," *arXiv preprint arXiv:2106.15326*, 2021.

[41] Y. Hou and L. Zheng, "Visualizing adapted knowledge in domain transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 824–13 833.

[42] H. Feng, Z. You, M. Chen, *et al.*, "Kd3a: Unsupervised multi-source decentralized domain adaptation via knowledge distillation," in *Proceedings of the 38th International Conference on Machine Learning*, 2021, pp. 3274–3283.

[43] I. Kuzborskij and F. Orabona, "Stability and hypothesis transfer learning," in *International Conference on Machine Learning*, PMLR, 2013, pp. 942–950.

[44] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proceedings of the 2006 conference on empirical methods in natural language processing*, 2006, pp. 120–128.

[45] Y. Zhang, "A survey of unsupervised domain adaptation for visual recognition," *arXiv preprint arXiv:2112.06745*, 2021.

[46] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2960–2967.

[47] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *International conference on machine learning*, PMLR, 2017, pp. 2208–2217.

[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[49] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[50] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[51] F. Pérez-Cruz, "Kullback-leibler divergence estimation of continuous distributions," in *2008 IEEE international symposium on information theory*, IEEE, 2008, pp. 1666–1670.

[52] F. Zhuang, X. Cheng, P. Luo, S. J. Pan, and Q. He, "Supervised representation learning with double encoding-layer autoencoder for transfer learning," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 2, pp. 1–17, 2017.

[53] Z. Meng, J. Li, Y. Gong, and B.-H. Juang, "Adversarial teacher-student learning for unsupervised domain adaptation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 5949–5953.

[54] M. Menéndez, J. Pardo, L. Pardo, and M. Pardo, "The jensen-shannon divergence," *Journal of the Franklin Institute*, vol. 334, no. 2, pp. 307–318, 1997.

[55] J. Jiang, X. Wang, M. Long, and J. Wang, "Resource efficient domain adaptation," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2220–2228.

[56] E. Englesson and H. Azizpour, "Generalized jensen-shannon divergence loss for learning with noisy labels," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[57] L. Rüschendorf, "The wasserstein distance and approximation theorems," *Probability Theory and Related Fields*, vol. 70, no. 1, pp. 117–129, 1985.

[58] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, PMLR, 2017, pp. 214–223.

[59] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 285–10 295.

[60] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical review E*, vol. 69, no. 6, p. 066 138, 2004.

[61] X. Xie, J. Chen, Y. Li, L. Shen, K. Ma, and Y. Zheng, "Mi2gan: Generative adversarial network for medical image domain adaptation using mutual information constraint," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2020, pp. 516–525.

[62] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," in *International Conference on Machine Learning*, PMLR, 2020, pp. 6028–6039.

[63] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. 25, pp. 723–773, 2012.

[64] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf, "Domain adaptation with conditional transferable components," in *International conference on machine learning*, PMLR, 2016, pp. 2839–2848.

[65] J. Na, H. Jung, H. J. Chang, and W. Hwang, "Fixbi: Bridging domain spaces for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1094–1103.

[66] V. Verma, A. Lamb, C. Beckham, *et al.*, "Manifold mixup: Better representations by interpolating hidden states," in *International Conference on Machine Learning*, PMLR, 2019, pp. 6438–6447.

[67] R. Volpi, P. Morerio, S. Savarese, and V. Murino, "Adversarial feature augmentation for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5495–5504.

[68] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282.

[69] P. Kairouz, H. B. McMahan, B. Avent, *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[70] T. Wang, J.-Y. Zhu, A. Torralba, and A. A. Efros, "Dataset distillation," *arXiv preprint arXiv:1811.10959*, 2018.

[71] Y. Zhu, X. Yu, Y.-H. Tsai, F. Pittaluga, M. Faraki, Y.-X. Wang, *et al.*, "Voting-based approaches for differentially private federated learning," *arXiv:2010.04851*, 2020.

[72] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals Systems (2nd Ed.)* USA: Prentice-Hall, Inc., 1996, ISBN: 0138147574.

[73] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd. USA: Prentice Hall Press, 2009, ISBN: 0131988425.

[74] W. T. Cochran, J. W. Cooley, D. L. Favin, *et al.*, "What is the fast fourier transform?" *Proceedings of the IEEE*, vol. 55, no. 10, pp. 1664–1674, 1967.

[75] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.

[76] MATLAB, *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2010.

[77] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," *Advances in Neural Information Processing Systems*, vol. 21, 2008.

[78] S. Zhao, B. Li, P. Xu, X. Yue, G. Ding, and K. Keutzer, "Madan: Multi-source adversarial domain aggregation network for domain adaptation," *International Journal of Computer Vision*, pp. 1–26, 2021.

[79] Y. Wang, Z. Zhang, W. Hao, and C. Song, "Attention guided multiple source and target domain adaptation," *IEEE Transactions on Image Processing*, vol. 30, pp. 892–906, 2020.

[80] Y. Zuo, H. Yao, and C. Xu, "Attention-based multi-source domain adaptation," *IEEE Transactions on Image Processing*, vol. 30, pp. 3793–3803, 2021.

[81] N. Venkat, J. N. Kundu, D. K. Singh, A. Revanur, *et al.*, "Your classifier can secretly suffice multi-source domain adaptation," in *NeurIPS*, 2020.

[82] Y. Zhu, F. Zhuang, J. Wang, *et al.*, "Deep subdomain adaptation network for image classification," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 4, pp. 1713–1722, 2020.

[83] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2200–2207.

[84] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah, "In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning," in *International Conference on Learning Representations*, 2020.

[85] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*, Springer, 2010, pp. 213–226.

[86] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5018–5027.

[87] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[88] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," *Advances in Neural Information Processing Systems*, vol. 29, pp. 136–144, 2016.

[89] H. Wang, M. Xu, B. Ni, and W. Zhang, "Learning to combine: Knowledge aggregation for multi-source domain adaptation," in *European Conference on Computer Vision*, Springer, 2020, pp. 727–744.

[90] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[91] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual information analysis," in *International Workshop on Cryptographic Hardware and Embedded Systems*, Springer, 2008, pp. 426–442.

[92] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui, "Generalized source-free domain adaptation," in *International Conference on Computer Vision*, 2021, pp. 8978–8987.

[93] K. Tanwisuth, X. Fan, H. Zheng, *et al.*, "A prototype-oriented framework for unsupervised domain adaptation," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[94] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" *Advances in neural information processing systems*, vol. 32, 2019.

[95] Y. Yang and S. Soatto, "Fda: Fourier domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4085–4095.

[96] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6023–6032.

[97] M. Abdellah, S. Saleh, A. Eldeib, and A. Shaarawi, "High performance multi dimensional (2d/3d) fft-shift implementation on graphics processing units (gpus)," in *2012 Cairo International Biomedical Engineering Conference (CIBEC)*, IEEE, 2012, pp. 171–174.

[98] R. Xu, G. Li, J. Yang, and L. Lin, "Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1426–1435.

[99] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," *Advances in neural information processing systems*, vol. 31, 2018.

[100] H. Tang, K. Chen, and K. Jia, "Unsupervised domain adaptation via structurally regularized deep clustering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8725–8735.

[101] S. Cui, S. Wang, J. Zhuo, L. Li, Q. Huang, and Q. Tian, "Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3941–3950.

[102] Y. Kim, D. Cho, K. Han, P. Panda, and S. Hong, "Domain adaptation without source data," *arXiv:2007.01524*, 2020.

[103] V. K. Kurmi, V. K. Subramanian, and V. P. Namboodiri, "Domain impression: A source data free domain adaptation method," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 615–625.

[104] H.-W. Yeh, B. Yang, P. C. Yuen, and T. Harada, "Sofa: Source-data-free feature alignment for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 474–483.

[105] R. Li, Q. Jiao, W. Cao, H.-S. Wong, and S. Wu, "Model adaptation: Unsupervised domain adaptation without source data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9641–9650.

[106] H. Xia, H. Zhao, and Z. Ding, "Adaptive adversarial network for source-free domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9010–9019.