ABSTRACT

Title of Dissertation: PRACTICAL ROBUST LEARNING
UNDER DOMAIN SHIFTS

Luyu Yang
Doctor of Philosophy, 2022

Dissertation Directed by: Professor Abhinav Shrivastava
Department of Computer Science

Professor Larry S. Davis
Department of Computer Science

With the constantly upgraded devices, the data we capture is shifting with time. Despite the domain shifts among the images, we as humans can put aside the difference and still recognize the content. However, these shifts are a bigger challenge for machines. It is widely known that humans are naturally adaptive to the visual changes in the environment, without learning all over again. However, to make machines work in the changed environment we need new annotations from human. The fundamental question is: can we make machines as adaptive as humans?

In this thesis, we have worked towards addressing this question through advances in the study of robust learning under domain shifts via domain adaptation. Our goal is to facilitate the transfer of information of the machines while minimizing the need for human supervision.

To enable real systems with demonstrated robustness, the study of domain adaptation needs to move from ideals to realities. In current domain adaptation research, there are few ideals that

are not consistent with reality: i) The assumption that domains are perfectly sliced and that domain labels are available. ii) The assumption that the annotations from the target domain should be treated equally as those of the source domain. iii) The assumption that the samples of target domains are constantly accessible. In this thesis, we try to address the issue that true domain labels are hard to obtain, the target domain labels have better ways to exploited, and that in reality the target domain is often time-sensitive.

In the scope of problem settings, this thesis has covered the following scenarios with practical values. Unsupervised multi-source domain adaptation, semi-supervised domain adaptation and online domain adaptation. Three completed works are reviewed corresponding to each problem setting. The first work proposes an adversarial learning strategy that learns a dynamic curriculum for source samples to maximize the utility of source labels of multiple domains. The model iteratively learns which domains or samples are best suited for aligning to the target. The intuition is to force the adversarial agent to constantly re-measure the transferability of latent domains over time to adversarially raise the error rate of the domain discriminator. The method has removed the need of domain labels, yet it outperforms other methods on four well-known benchmarks by significant margins. The second work aims to address the problem that current methods have not effectively used the target supervision by treating source and target supervision without distinction. The work points out that the labeled target data needs to be distinguished from the source, and propose to explicitly decompose the task into two sub-tasks: a semi-supervised learning task in the target domain and an unsupervised domain adaptation task across domains. By doing so, the two sub-tasks can better leverage the corresponding supervision and thus yield very different classifiers. The third work is proposed in the context of online privacy, i.e. each online sample of the target domain is permanently deleted after processed. The proposed framework utilizes the labels from the public data and predicts on the unlabeled sensitive private data. To tackle the inevitable distribution shift from the public data to the private data, the work proposes a novel domain adaptation algorithm that directly aims at the fundamental challenge of this online setting–the lack of diverse source-target data pairs.

PRACTICAL ROBUST LEARNING UNDER DOMAIN SHIFTS

by

Luyu Yang

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2022

Advisory Committee:
Professor Abhinav Shrivastava, Advisor
Professor Larry S. Davis, Co-Advisor
Professor Joseph F. JaJa, Department Representative
Professor Ramani Duraiswami
Professor Judy Hoffman (Georgia Tech)

To my mother,

who carries my world.

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

First, I'd like to thank my advisor, Professor Abhinav Shrivastava for supporting me, providing me with the much-needed security behind my explorations. Without the precious freedom he has given me over the past three years, I would not have learned so much about research, and especially about myself. He has been a mentor for me, teaching me how face the ups and downs in research, and sometimes the inevitable failures with maturity. He is the anchor in the heart of many PhD students, it has been and will always be a pleasure to learn from this extraordinary being.

I want to also thank my co-advisor, Professor Larry Davis for giving me an invaluable opportunity to join the amazing computer science department, through which I have worked with many brilliant minds. I appreciate and cherish the guidance he has given me in my first year. During my PhD years, I have benefited deeply from the research community Larry established, we all carry the enlightening yet practical spirit that Larry has illustrated.

I would also like to thank Professor Judy Hoffman from Georgia Tech, who has showered me with so many genius ideas and insightful perspectives. Meanwhile, I appreciate Professor Joseph F. JaJa and Professor Ramani Duraiswami for agreeing to serve on my committee.

Much appreciation to Professor Rama Chellappa and his humor, to Professor Wei-Lun

Chao from Ohio State for being a role-model of striving for perfection, To Professor Kilian Q. Weinberger and Professor Bharath Hariharan from Cornell.

I cherish the experience of working with many brilliant collaborators throughout these years: Yan Wang, Shuaiyi Huang, Yu Shen, Zeyuan Chen, Ran Xu, Chetan Ramaiah and Ser-Nam Lim. I am especially grateful for the guidance from the senior students Yogesh Balaji and Mingfei Gao, for making the PhD life less challenging. I also benefit from dicussions with Kamal Gupta, Max Ehrich, Moustafa Meshry and Ahmed Taha, who help me to keep the sanity. Appreciations for the good time and good food with my other fun and amazing labmates.

My research is funded by Facebook AI, NSF (III-1618134, III- 1526012, IIS-1149882, IIS-1724282, TRIPODS-1740822, OAC-1934714, DMR-1719875), the Office of Naval Research DOD (N00014-17-1-2175), DARPA SAIL-ON (W911NF2020009). There is no collaboration between Facebook and the government agencies.

Finally, I owe my deepest gratitude to my family, my mother and father who support and respect my decisions. Especially my mother, who has buried all the worries she has and provided me with all the love and trust I need. Thank my best friend Bean who have pulled me through the hardest times and refilled me with laughter.

It is impossible to remember all, and I apologize to those whom I unintentionally left out.

God bless you all!

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

The way we create data is changing rapidly. Where cellphones were once only a mobile alternative to telephones, in the span of a decade, they have instead become a system that bridges the most personal wants and needs with the ubiquitous access to the internet. Indeed, the internet updates constantly to provide the newest feedback for every search. The result of searching "New York City", for example, now contains images from professional cameras, cellphones, drones, and even vehicle mounted cameras. Despite the domain shifts among the images, we as humans still recognize the city. However, these shifts are a much bigger challenge for machines. My research answers the fundamental question: *how can we make machines as adaptive as humans?* In this thesis, we have worked towards addressing this question through advances in the study of robust learning under domain shifts via **domain adaptation**.

The majority of recent machine learning progress has been made without considering these domain shifts [2, 3, 4, 5, 6]. Modern machine learning systems, fueled by deep neural networks, typically require large labeled datasets to achieve good performance, where "good performance" is usually evaluated on a held-out test dataset. This dataset is often drawn from the same distribution as the input training dataset. However, a more representative test distribution would contain new variations that are not the training set. For instance, a Waymo autonomous driving system trained with road test data collected on a sunny California day, might be prone

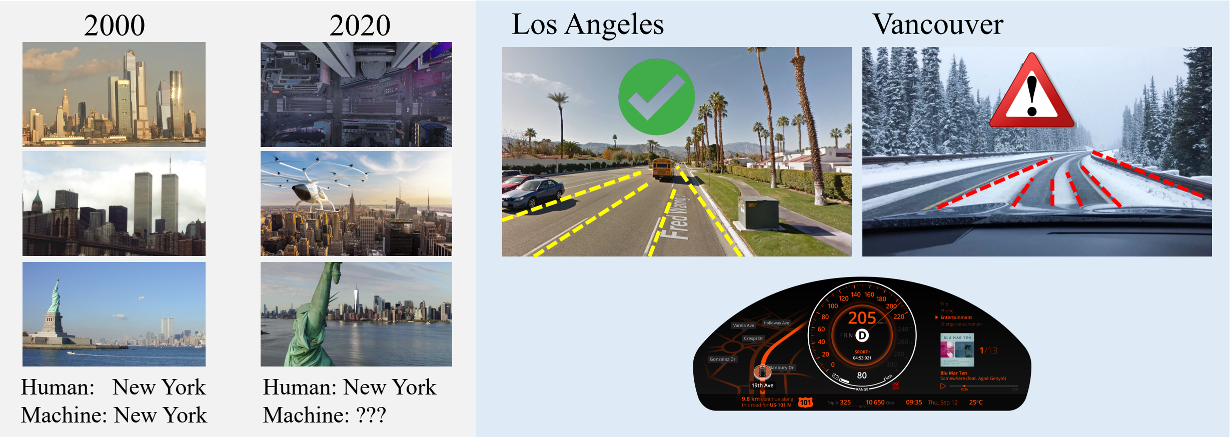

Figure 1.1: Examples of situations where machines cannot adaptive as well as humans. *Left*: Images taken by drones in 2020 with varying perspectives cannot be recognized by models trained with professional images from 2000. *Right*: Autonomous vehicle models trained with California cityscape data fail to detect lane lines in a snowy scene in Canada.

to worse performance on a snowy night in Canada. Domain adaptation can enable the system to function in Canada without abandoning the expensive data and annotations from California, as illustrated in Figure 1.1.

To enable real systems with demonstrated robustness, the study of domain adaptation needs to move from ideals to realities. In current domain adaptation research, there are a few unrealistic assumptions: i) The assumption that domains are perfectly sliced and that domain labels are available. ii) The assumption that the samples of target domains accessible. iii) The trained model can generate invariant representations that work well on both the source and target domains. In this thesis, we propose to mention that true domain labels are hard to obtain, and in reality the target domain is often time-sensitive. Moreover, the best representation for the target domain is not necessarily the best for the source domain. These challenges brought up by these unrealistic assumptions cannot be tackled by the simple devise upon existing methods, and call for entirely new designs. In my dissertation research we propose three approaches to the challenges.

First, we address the issue that true domain labels are hard to obtain. Currently, domain adaptation benchmarks are collected based on human prior knowledge of how domains are partitioned. For example *Cartoon*, *Art* and *Photo* [7]. In most domain adaptation datasets, a specific domain is built by crawling the internet for images with the domain name as a keyword [7, 8]. We argue that this does not fully match with the concept of domain in reality. In practice, datasets differ from each other by how they are generated rather than retrieved. A domain can naturally contain multiple sub-domains. For example, a specific kind of camera that captures four seasons can generate the domain *Photo* with four sub-domains, each representing a completely different season. Moreover, it is hard to decide gap between domains solely based on human prior knowledge. For example, does a lifelike painting belong to the *Art* domain or *Photo*? Therefore, the ground truths of domain membership are hard to obtain in reality. Instead of dwelling on these labels, we address this issue by automatically discovering the domains. For each input with mixed domains, we partition the input by comparing their feature-level similarities. Then we use the partitioned input for further adaptation.

Second, we address the issue that the best representation for target domain may not be the best cross-domain representation. Many domain adaptation works are based on the goal of generating invariant representation across domains via adversarial training [9, 10, 11, 12, 13, 14, 15, 16, 17]. However, we argue that in some cases where there is a small amount of supervision from the target domain, the best model on the target domain may not be obtained via cross-domain representation learning. Specifically, in semi-supervised domain adaptation, we propose to explicitly decompose the two sources of supervision and learn two distinct classifiers whose goals are however shared: to work well on the unlabeled target data. To this end, we pair the labeled source data and the unlabeled target data to learn one classifier, which is essentially a unsupervised domain adaptation (UDA) task. For the other classifier, we pair the labeled and unlabeled target data, which is essentially a semi-supervised learning (SSL) task. That is, we explicitly decompose SSDA into two well-studied tasks.

Third, we address the issue that the model usually has limited access to the target domain in reality. In many applications, user data is private and collecting the user's data for training purposes is strictly forbidden [18, 19, 20, 21, 22, 23, 24]. This issue requires the model to adapt well to the target domain in an online streaming fashion. Unlike the conventional domain adaptation setting in which each data point can be used to adjust the model parameters many times, in online domain adaptation the model has *exactly one* chance to adapt and predict before it is deleted. Therefore at each online query, it is critical for the model to balance between performing well on this particular query and learning a general representation on the entire target domain. To bridge this gap, for each target query we assemble it with multiple different queries drawn from the source distribution to alleviate the single-point bias.

Finally, we conclude the dissertation with a summary of presented works and several future research directions.

Chapter 2: Metrics in Domain Adaptation

In this chapter, we discuss the metrics used in domain adaptation including: i) The metrics that characterize the domain divergence, ii) the evaluation metrics that measure how successful the adaptation is. We first review two distribution divergence metrics and their formulations under domain adaptation setting. Then we review the commonly used evaluation metrics, including evaluation with and without labeled test dataset.

2.1 Domain Divergence Metric

In order to quantatize the gap across domains, a measure of domain divergence is needed. Currently, there are many metrics of differences, such as the Kullback-Leibler (KL) divergence, the total variation distance, the Wasserstein metric, and the Kolmogorov Smirnoff statistic [25, 26], measuring between probability distributions or datasets.

For the methods that are developed based on domain gap metrics, the choice will often influence the behaviour of the adaptive model. In this chapter, we will discuss two widely-used metrics in detail apart from the most popular KL-divergence, the $\mathcal{H}$-divergence and Wasserstein distance metric.

2.1.1 $\mathcal{H}$-Divergence

Given the source distribution $\mathcal{S}$ and target distribution $\mathcal{T}$ over the input and the output space $\mathcal{X} \times \mathcal{Y}$. If the data is generated by a marginal distribution and underlying labeling function pair $(\mathcal{D}, h^*)$, then the upper bound of target risk error w.r.t. $\forall h \in \mathcal{H}$ is

$$R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + d_{\mathcal{H}}(\mathcal{T}(x), \mathcal{S}(x)) + \beta, \tag{2.1}$$

where

$$R_{\mathcal{D}}(h) = \mathbb{E}_{x \backsim \mathcal{D}} |h(x) - h^*(x)|.$$

$d_{\mathcal{H}}$ denotes the $\mathcal{H}$-Divergence for measuring the marginal distribution similarities and β is the optimal joint risk over the two domains. In practice, it is impossible to exactly estimate the $\mathcal{H}$-Divergence [27]. Practically, we can approximate this measure as binary classification task on discriminating the source and the target samples, i.e. approximated by distance $d_{\mathcal{A}} = 2(1 - 2\epsilon)$ (ϵ is the discrimination generalization error). Thus, the $\mathcal{H}$-Divergence metric-based loss between the domain classifier d and the feature extractor g in the context of representation learning is:

$$\min_{g} \max_{d} \mathbb{E}_{x_s \backsim \mathcal{S}(x)} log(d \circ g(x_s)) + \mathbb{E}_{x_s \backsim \mathcal{T}(x)} log(1 - d \circ g(x_t)), \tag{2.2}$$

.

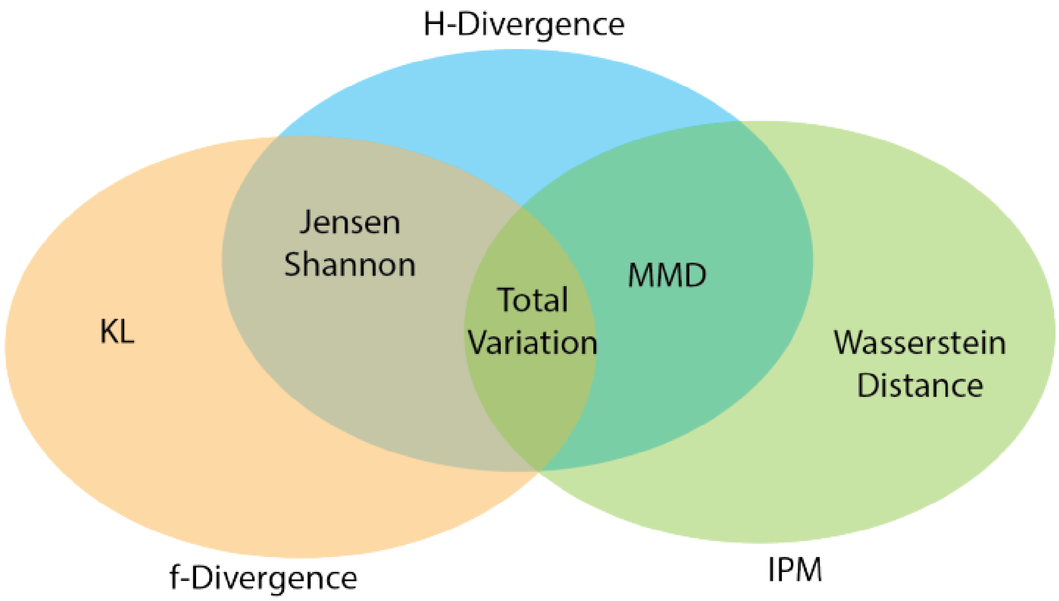

Figure 2.1: A coverage map and the relationship of the commonly-used metrics. In this chapter, we discuss the $\mathcal{H}$-divergence and Wasserstein distance metrics in detail. Picture from $\mathcal{H}$-*divergence: A Decision-theoretic Probability Discrepancy Measure* [28]

2.1.2 Wasserstein Distance

The Wasserstein metric [29] is a distance measure between probability distribution on a given metric space (M, ρ), where $\rho(x, y)$ is a distance function for two instances x and y in the set M. The ρ-th Wasserstein distance between two Borel probability measures [30] $\mathbb{P}$ and $\mathbb{Q}$ is defined as:

$$W_\rho(\mathbb{P}, \mathbb{Q}) = (\inf_{\mu \in \Gamma(\mathbb{P}, \mathbb{Q})} \int \rho(x, y)^p d\mu(x, y))^{1/p}, \tag{2.3}$$

where $\mathbb{P}, \mathbb{Q} \in \{\mathbb{P} : int \rho(x, y)^p d\mathcal{P}(x) < \infty, \forall y \in M\}$ are two probability measures on M with finite p-th moment and $\Gamma(\mathbb{P}, \mathbb{Q})$ is the set of all measures on $M \times M$ with marginals $\mathbb{P}$ and $\mathbb{Q}$.

Wasserstein metric is one of the most widely-used in the problem of optimal transport. In domain adaptation, the metric is often employed to calculate the divergence between the source and target representations in adversarial training. More formally, given an instance $x \in \mathbb{R}^m$ from either the source or target domain, the feature extractor learns a function $f_g : \mathbb{R}_m \rightarrow \mathbb{R}^d$ that maps x to a representation vector with d dimensions.

Given a feature representation $h = f_g(x)$, the domain classifier learns a mapping function $f_w : \mathbb{R}^d \rightarrow \mathbb{R}$ that maps the feature representation to a real number with parameter θ_w. The Wasserstein distance [29] between two representation distributions $\mathbb{P}_{h^s}$ and $\mathbb{P}_{h^t}$, where $h^s = f_g(x^s)$ and $h^t = f_g(x^t)$ can be computed according to:

$$\begin{aligned} W_1(\mathbb{P}_{h^s}, \mathbb{P}_{h^t}) &= \sup_{\|f_w\|_L \leq 1} \mathbb{E}_{\mathbb{P}_{h^s}}[f_w(h)] - \mathbb{E}_{\mathbb{P}_{h^t}}[f_w(h)] \\ &= \sup_{\|f_w\|_L \leq 1} \mathbb{E}_{\mathbb{P}_{x^s}}[f_w(f_g(x))] - \mathbb{E}_{\mathbb{P}_{x^t}}[f_w(f_g(x))]. \end{aligned} \tag{2.4}$$

If the parameters of domain classifiers $\{f_w\}$ are all 1-Lipschitz, then Eq.2.4 can be approximated using:

$$\mathcal{L}_{wd}(x^s, x_t) = \frac{1}{n^s} \sum_{x^s \in X^s} f_w(f_g(x^s)) - \frac{1}{n^t} \sum_{x^t \in X^t} f_w(f_g(x^t)). \tag{2.5}$$

2.2 Evaluation Metrics

The goal of domain adaptation is to transfer information learned on a label-rich source domain, to a target domain without labels. Directly, we can measure whether the goal is achieved by measuring the average classification accuracy on the test dataset of the target domain. The higher the accuracy, the better. However, to measure the neat gain of domain adaptation, we need to consider more. In this section, we review the evaluation metrics for adaptation.

2.2.1 Measuring the Effectiveness of Domain Adaptation Approaches

Given a source dataset D_S and a target dataset D_T, we independently learn three models. A source-only model trained in a supervised learning fashion using all the labels and samples from the source dataset. An oracle model trained in a supervised learning fashion using all the labels and images from the target dataset. The model trained using the proposed domain adaptation approach using all the labels and images from the source dataset, and only the images from the target dataset. Each model has the same architecture, initialization, and is optimized to their best performance. Then we evaluate the test dataset from the target domain by calculating mean average precision, each result denoted as acc_{SO}, acc_{OC} and acc_{ours}.

An effective adaptation from the source to the target should observe $acc_{\text{ours}} > acc_{\text{SO}}$. However, the observation of $acc_{\text{ours}} < acc_{\text{SO}}$ is not meaningless, it indicates that the method might have caused negative transfer phenomenon [8]. The success of the adaptation is measured by the gap between the proposed result and the oracle result $|acc_{\text{CO}} - acc_{\text{ours}}|$, the lower the better. If acc_{ours} approaches acc_{CO}, it indicates that the adapted model trained without any target labels can perform as well as supervised learning with labels, which is an ideal outcome of domain adaptation.

2.2.2 Evaluation Without Test Labels

For unsupervised domain adaptation, we need label-free evaluation for two reasons:

- In an unsupervised adaptation setting, we assume it is hard to obtain labels on the target domain;
- The accuracy of models on the testset might vary, if the testset distribution slightly differs from the entire target dataset.

In [31], a label-free accuracy estimation approach is explored. Specifically, the difference of confidences (DoC) approach yields reliable estimates of a classifier's performance over a variety of shifts and model architectures.

Given an arbitrary target dataset T and a helod-out test dataset B, let F represent a model and $F(x)$ represent the output probabilities of F over instance x, the difference of confidences DoC is computed via AC, which is average confidences based on a featurization $F^{'}$ of the probabilities of the model. Since some of the distribution shifts might change the label space, we consider the AC w.r.t. both T and B, as $K_{B \cap T}$:

$$
\begin{aligned}
AC_B^T &= \frac{1}{|B'|} \sum_{x \in B} \max_{\{K_{B \cap T}\}} (F(x)), \\
DoC_{B,T} &= AC_T^B - AC_B^T. \qquad (2.6)
\end{aligned}
$$

2.3 Conclusion

In this chapter, we discuss the metrics that measure the domain divergence. Most alignment-based domain adaptation methods are developed based on minimizing the defined divergence metric, in order to obtain feature representations that are invariant across domains. The representation is often learned through adversarial learning with a domain critic. We will discuss in detail the adversarial learning in domain adaptation in Chapter 3.

Chapter 3: Domain-Adversarial Representation Learning

Domain-Adversarial Neural Network (DANN) [10] is one of the most popular methods in domain adaptation using deep neural networks. The method aims to learn a representation network that a domain critic cannot distinguish from source domain to the target domain. During training, the model of DANN will optimize over two objectives: i) the label classification loss over the source domain, ii) the domain classification loss on both the source and target domains. In this chapter, we will review and discuss the general domain-adversarial learning in practice.

3.1 Domain-Adversarial Architecture and Implementation

As illustrated in Figure 3.1, a general domain-adversarial network includes a deep feature extractor G_f with parameter θ_f and a label predictor G_y with parameter θ_y, which together form a standard feed-forward architecture for supervised learning. The unsupervised domain-adversarial (target domain without labels) is achieved by adding a domain critic (classifier) G_d which is also connected to the feature extractor. The difference between G_d and G_y is that G_d is connected to G_f via a gradient reversal layer (GRL) that reverse the gradient of G_d by a constant during training.

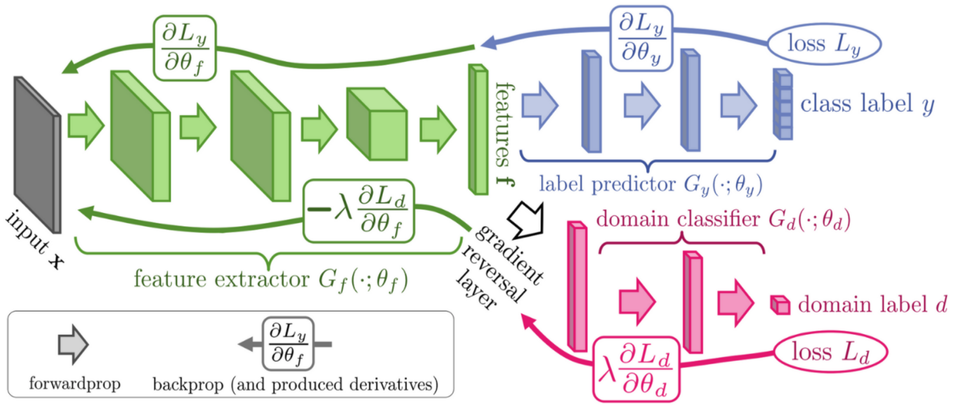
$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
class label y
features $\mathbf{f}$
label predictor $G_y(\cdot;\theta_y)$
$-\lambda\frac{\partial L_d}{\partial \theta_f}$
input $\mathbf{x}$
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
feature extractor $G_f(\cdot;\theta_f)$
domain label d
$\lambda\frac{\partial L_d}{\partial \theta_d}$
loss L_d
forwardprop
$\frac{\partial L_y}{\partial \theta_f}$
backprop (and produced derivatives)

Figure 3.1: The architecture of domain-adversarial neural network. Picture from the work that proposes DANN [10].

3.1.1 Gradient Reversal Layer

The “game-changer” of domain-adversarial is the gradient reverse operation, which enables the adversarial learning by maximizes the domain critic loss, achieved using the GRL module. The GRL module has no parameters associated with it. During the forward propagation, the GRL acts as an identity transformation. However, during the backpropagation, the GRL takes the gradient from the subsequent level and changes its sign, before passing it to the preceding layer. The implementation of GRL is as simple as the following code block.

```
class ReverseLayerF(Function):
    def forward(ctx, x, alpha):
        ctx.alpha = alpha
        return x.view_as(x)
    def backward(ctx, grad_output):
        output = grad_output.neg() * ctx.alpha
        return output, None
```

One important factor to facilitate the successful domain-adversarial learning is the constant λ applied to the GRL. The annealing factor is caculated using:

$$\lambda = \frac{2}{(1 + \exp^{-10 \times p})} - 1, \tag{3.1}$$

where

$$p = \frac{j}{J_{\text{max}}}$$

p indicates percentage of iteration (current j out of total J_{max}) during training. It is worth noticing that DANN can be unstable at the early stage of training, when the network has not fitted the labels of the source domain yet. The ramp-up factor controlled by λ serves as stabilizer, balancing the importance between label and domain learning. An illustration of constant λ changes over $20,000$ iterations training is in Figure 3.2.

3.2 Limitations of Domain-Adversarial

One major limitation of domain-adversarial networks is that the matched data distribution do not directly imply that the class-conditional distributions are well-matched [32, 33]. Therefore, domain-adversarial-based methods can sometimes be hard to train, or may result in generating ambiguous features near the task decision boundary. An illustration of the situation where conventional domain classifier-based methods do not work well is in Figure 3.3. A possible solution to alleviate the problem is to directly improve the domain discriminator. Conditional

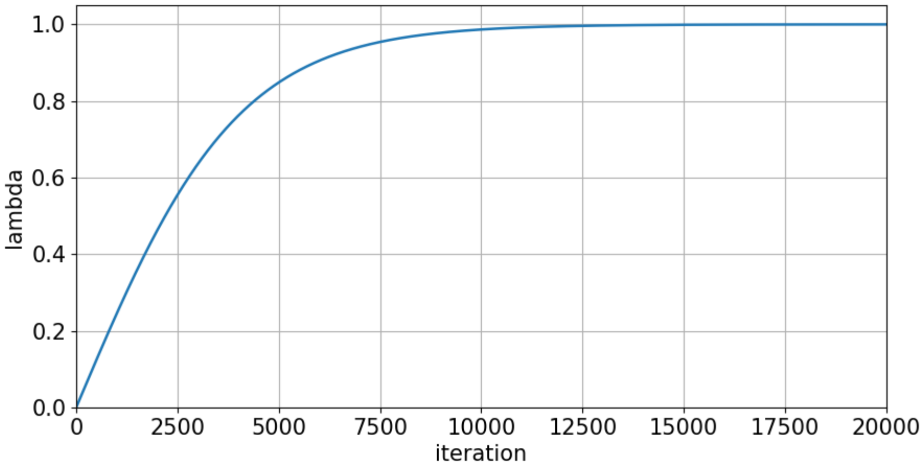

Figure 3.2: Annealing factor λ applied to GRL changes over training $(20,000$ iterations for example), plotted using Eq. 3.1.

Adversarial Domain Adaptation [13] (CDAN) explores a improved solution by pointing out two directions: first, when the joint distribution of feature and class are non-identical across domains, adapting only the feature representation may be insufficient. Second, when the feature distribution is multi-modal, adapting only the feature representation may be challenging for domain-adversarial methods. Therefore, CDAN proposes a conditional domain discriminator conditioned on the cross-covariance of domain-specific feature representations and classifier predictions, which further condition the domain discriminator on the uncertainty of classifier predictions, prioritizing the discriminator on easy-to-transfer samples. Another solution proposed is called DIRT-T [34]. The DIRT-T approach incorporates the natural gradient and guiding the network to avoid crossing high-density data regions with its decision boundary.

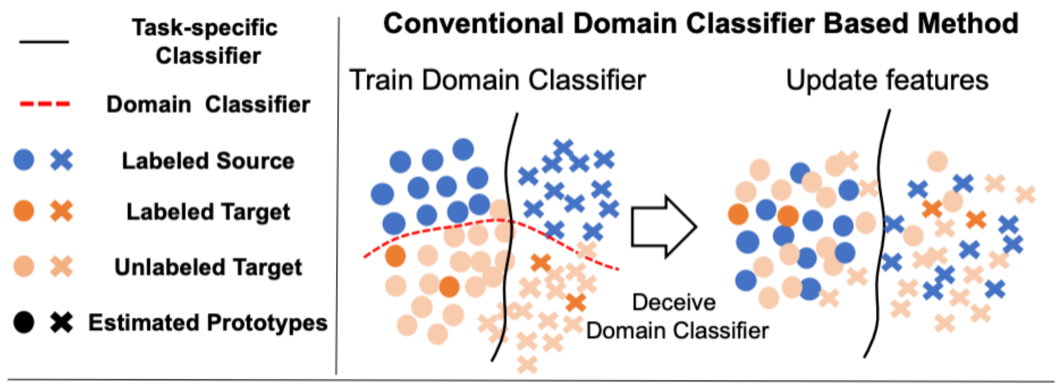

Figure 3.3: Illustration of ambiguous features near the task decision boundary generated by domain-adversarial-based method. Picture partially from [32].

3.3 Other Min-Max Games

Apart from domain-adversarial, there are numerous methods based on the min-max games. Maximum Classifier Discrepancy (MCD) [35] attempts to align distributions of source and target by utilizing the task-specific decision boundaries. In their network architecture, two classifiers (category classifier, not domain classifier) are used to perform a min-mix iterative game between the discrepancy of the two classifiers' outputs.

In a semi-supervised domain adaptation setting, Min-Max Entropy Approach (MME) adversarially optimizes an adaptive few-shot model. The network is composed of a feature encoding network, followed by a classification layer that computes the features' similarity to estimated prototypes, achieving adaptation by maximizing the conditional entropy of unlabeled target data and minimizing it with respect to the feature encode alternatively.

3.4 Conclusion

In this chapter, we discuss an important branch of domain adaptation methods – domain-adversarial representation learning. We first review the domain-adversarial neural network architecture by introducing each component, then we focus on the gradient reversal layer and its implementation. We finally elaborate the limitations of this idea, and review a few strategies that effectively alleviate the limiation.

Chapter 4: Curriculum Manager for Multi-Source Domain Adaptation

The performance of Multi-Source Unsupervised Domain Adaptation depends significantly on the effectiveness of transfer from labeled source domain samples. In this chapter, we proposed an adversarial agent that learns a dynamic curriculum for source samples, called Curriculum Manager for Source Selection (CMSS). The Curriculum Manager, an independent network module, constantly updates the curriculum during training, and iteratively learns which domains or samples are best suited for aligning to the target. The intuition behind this is to force the Curriculum Manager to constantly re-measure the transferability of latent domains over time to adversarially raise the error rate of the domain discriminator. CMSS does not require any knowledge of the domain labels, yet it outperforms other methods on four well-known benchmarks by significant margins. We also provide interpretable results that shed light on the proposed method.

4.1 Introduction

Training deep neural networks requires datasets with rich annotations that are often time-consuming to obtain. Previous proposals to mitigate this issue have ranged from unsupervised [36, 37, 38, 39, 40, 41], self-supervised [42, 43, 44, 45], to low shot learning [35, 46, 47, 48]. Unsupervised Domain Adaptation (UDA), when first introduced in [49], sheds precious insights on how adversarial training can be utilized to get around the problem of expensive manual annotations. UDA aims to preserve the performance on an unlabeled dataset (target) using a model trained on a label-rich dataset (source) by making optimal use of the learned representations from the source.

Intuitively, one would expect that having more labeled samples in the source domain will be beneficial. However, having more labeled samples does not equal better transfer, since the source will inadvertently encompass a larger variety of domains. While the goal is to learn a common representation for both source and target in such a Multi-Source Unsupervised Domain Adaptation (MS-UDA) setting, enforcing each source domain distribution to exactly match the target may increase the training difficulty, and generate ambiguous representations near the decision boundary potentially resulting in negative transfer. Moreover, for practical purposes, we would expect the data source to be largely unconstrained, whereby neither the number of domains or domain labels are known. A good example here would be datasets collected from the Internet where images come from unknown but potentially a massive set of users.

To address the MS-UDA problem, we propose an adversarial agent that learns a dynamic curriculum [50] for multiple source domains, named Curriculum Manager for Source Selection (CMSS). More specifically, a constantly updated curriculum during training learns which domains or samples are best suited for aligning to the target distribution. The CMSS is an independent module from the feature network and is trained by maximizing the error of discriminator in order to weigh the gradient reversal back to the feature network. In our proposed adversarial interplay with the discriminator, the Curriculum Manager is forced to constantly re-measure the transferability of latent domains across time to achieve a higher error of the discriminator. Such a procedure of weighing the source data is modulated over the entire training. In effect, the latent domains with different transferability to the target distribution will gradually converge to different

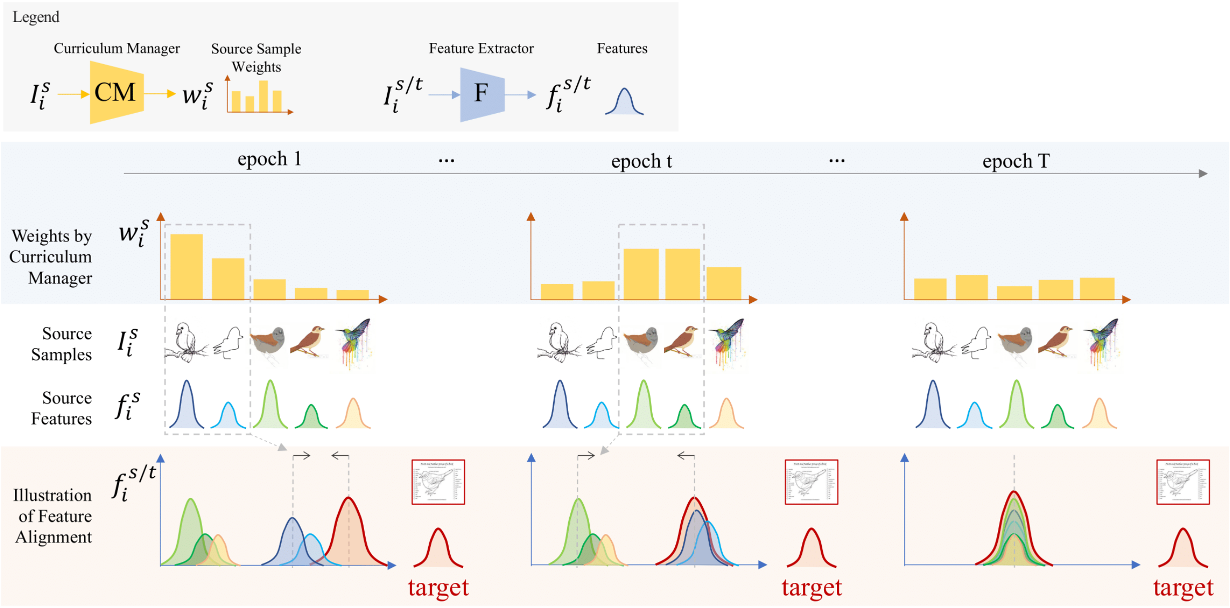

Figure 4.1: Illustration of CMSS during training. All training samples are passed through the feature network F. CMSS prefers samples with better transferability to match the target, and re-measure the transferability at each iteration to keep up with the discriminator. At the end of training after the majority of samples are aligned, the CMSS weights tend to be similar among source samples.

levels of importance without any need for additional domain partitioning prior or clustering.

We attribute the following contributions to this work:

- We propose a novel adversarial method during training towards the MS-UDA problem. Our method does not assume any knowledge of the domain labels or the number of domains.

- Our method achieves state-of-the-art in extensive experiments conducted on four well-known benchmarks, including the large-scale DomainNet ($\sim$ 0.6 million images).

- We obtain interpretable results that show how CMSS is in effect a form of curriculum learning that has great effect on MS-UDA when compared to the prior art. This positively differentiates our approach from previous state-of-the-art.

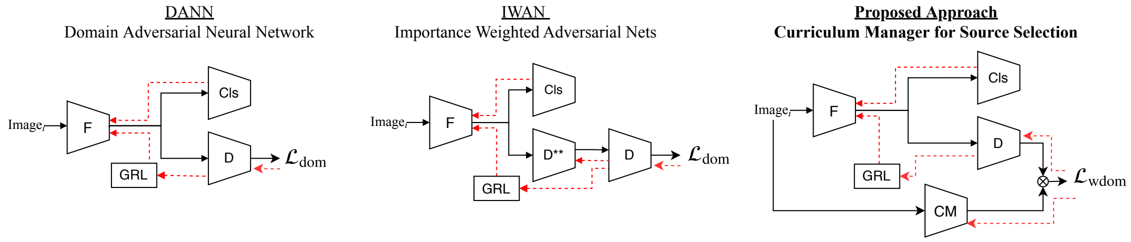
DANN
Domain Adversarial Neural Network
Image$_t$
F
Cls
D
GRL
$\mathcal{L}_{\text{dom}}$
IWAN
Importance Weighted Adversarial Nets
Image$_t$
F
Cls
D**
D
GRL
$\mathcal{L}_{\text{dom}}$
Proposed Approach
Curriculum Manager for Source Selection
Image$_t$
F
Cls
D
GRL
CM
$\mathcal{L}_{\text{wdom}}$

Figure 4.2: Architecture comparison of *left*: DANN [49], *middle*: IWAN [58], and *right*: proposed method. Red dotted lines indicate backward passes. (F: feature extractor, Cls: classifier, D: domain discriminator, GRL: gradient reversal layer, CM: Curriculum Manager, $\mathcal{L}_{\text{dom}}$: Eq.4.1 domain loss, $\mathcal{L}_{\text{wdom}}$: Eq.4.3 weighted domain loss)

4.2 Related Work

UDA is an actively studied area of research in machine learning and computer vision. Since the seminal contribution of Ben-David *et al.* [51, 52], several techniques have been proposed for learning representations invariant to domain shift [53, 54, 55, 56, 57]. In this section, we review some recent methods that are most related to our work.

Multi-Source Unsupervised Domain Adaptation (MS-UDA) assumes that the source training examples are inherently multi-modal. The source domains contain labeled samples while the target domain contains unlabeled samples [7, 49, 59, 60, 61]. In [59], adaptation was performed by aligning the moments of feature distributions between each source-target pair. Deep Cocktail Network (DCTN) [62] considered the more realistic case of existence of category shift in addition to the domain shift, and proposes a k-way domain adversarial classifier and category classifier to generate a combined representation for the target. Because domain labels are hard to obtain in the real world datasets, latent domain discovery [60] – a technique for alleviating the need for explicit domain label annotation has many practical applications. Xiong *et al.* [63] proposed to use square-loss mutual information based clustering with category distribution prior to infer the domain assignment for images. Mancini *et al.* [60] used a domain prediction branch to guide domain discovery using multiple batch-norm layers.

Domain-Adversarial Training has been widely used [14, 64, 65] since Domain-Adversarial Neural Network (DANN) [49] was proposed. The core idea is to train a discriminator network to discriminate source features from target, and train the feature network to fool the discriminator. Zhao *et al.* [61] first proposed to generalize DANN to the multi-source setting, and provides theoretical insights on the multi-domain adversarial bounds. Maximum Classifier Discrepancy (MCD) [35] is another powerful [59, 66, 67, 68] technique for performing adaptation in an adversarial manner using two classifiers. The method first updates the classifiers to maximize the discrepancy between the classifiers' prediction on target samples, followed by minimizing the discrepancy while updating the feature generator.

Domain Selection and Weighting: Some previous methods that employed sample selection and sample weighing techniques for domain adaptation include [69, 70, 71]. Duan *et al.* [70] proposed using a domain selection machine by leveraging a large number of loosely labeled web images from different sources. The authors of [70] adopted a set of base classifiers to predict labels for the target domain as well as a domain-dependent regularizer based on smoothness assumption. Bhatt *et al.* [72] proposed to adapt iteratively by selecting the best sources that learn shared representations faster. Chen *et al.* [64] used a hand-crafted re-weighting vector so that the source domain label distribution is similar to the unknown target label distribution. Mancini *et al.* [73] modeled the domain dependency using a graph and utilizes auxiliary metadata for predictive domain adaptation. Zhang *et al.* [58] employed an extra domain classifier that gives the probability of a sample coming from the source domain. The higher the confidence is from such an extra classifier, the more likely it can be discriminated from the target domain, in which case the importance of the said sample is reduced accordingly.

Curriculum for Domain Adaptation aims at an adaptive strategy over time in order to improve the effectiveness of domain transfer. The curriculum can be hand-crafted or learned. Shu *et. al* [74] designed the curriculum by combining the classification loss and discriminator's loss as a weighting strategy to eliminate the corrupted samples in the source domain. Another work with similar motivation is [40], in which Chen *et. al* proposed to use per-category prototype to measure the prediction confidence of target samples. A manually designed threshold τ is utilized to make a binary decision in selecting partial target samples for further alignment. Kurmi *et. al* [75] used a curriculum-based dropout discriminator to simulate the gradual increase of sample variance.

4.3 Preliminaries

4.3.1 Task Formulation:

In multi-source unsupervised domain adaptation (MS-UDA), we are given an input dataset $\mathcal{D}_{\text{src}} = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{N_s}$ that contains samples from multiple domains. In this chapter, we focus on classification problems, with the set of labels $y_i^s \in \{1, 2, \ldots, n_c\}$, where n_c is the number of classes. Each sample $\mathbf{x}_i^s$ has an associated domain label, $d_i^s \in \{1, 2, \ldots, S\}$, where S is the number of source domains. In this work, we assume source domain label information is not known *a priori*, i.e., number of source domains or source domain label per sample is not known. In addition, given an unlabeled target dataset $\mathcal{D}_{\text{tgt}} = \{\mathbf{x}_i^t\}_{i=1}^{N_t}$, the goal of MS-UDA is to train models using multiple source domains ($\mathcal{D}_{\text{src}}$) and the target domain ($\mathcal{D}_{\text{tgt}}$), and improve performance on the target test set.

4.3.2 Domain-Adversarial training:

First, we discuss the domain-adversarial training formulation from [49] that is the basis from which we extend to MS-UDA. The core idea of domain-adversarial training is to minimize the distributional distance between source and target feature distributions posed as an adversarial game. The model has a feature extractor, a classifier, and a domain discriminator. The classifier takes in feature from the feature extractor and classifies it in n_c classes. The discriminator is optimized to discriminate source features from target. The feature network, on the other hand, is trained to fool the discriminator while at the same time achieve good classification accuracy.

More formally, let $F_\theta : \mathbb{R}^{3\times w\times h} \to \mathbb{R}^d$ denote the feature extraction network, $C_\phi : \mathbb{R}^d \to \mathbb{R}^{n_c}$ denote the classifier, and $D_\psi : \mathbb{R}^d \to \mathbb{R}^1$ denote the domain discriminator. Here, θ, ϕ and ψ are the parameters associated with the feature extractor, classifier, and domain discriminator respectively. The model is trained using the following objective function:

$$\max_{\psi} \min_{\theta,\phi} \; \mathcal{L}_{\text{cls}} - \lambda \mathcal{L}_{\text{dom}} \tag{4.1}$$

$$\begin{aligned}
\text{where} \quad \mathcal{L}_{\text{cls}} &= -\frac{1}{N_s}\sum_{i=1}^{N_s} \tilde{\mathbf{y}}_i \log(C(F(\mathbf{x}_i^s))) \\
\mathcal{L}_{\text{dom}} &= -\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{src}}} \log(D(F(\mathbf{x}))) - \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{tgt}}} \log(1 - D(F(\mathbf{x}))) \\
&= -\frac{1}{N_s}\sum_{i=1}^{N_s} \log(D(F(\mathbf{x}_i^s))) - \frac{1}{N_t}\sum_{i=1}^{N_t} \log\big(1 - D(F(\mathbf{x}_i^t))\big)
\end{aligned}$$

$\mathcal{L}_{\text{cls}}$ is is the cross-entropy loss in source domain (with $\tilde{\mathbf{y}}_i$ being the one-hot encoding of the label y_i), and $\mathcal{L}_{\text{dom}}$ is the discriminator loss that discriminates source samples from the target. Note that both these loss functions use samples from all source domains.

In principle, if domain labels are available, there are two possible choices for the domain discriminator: (1) k domain discriminators can be trained, each one discriminating one of the source domains from the target [49], or (2) a domain discriminator can be trained as a $(k+1)$-way classifier to classify input samples as either one of the source domains or target [61]. However, in our setup, domain labels are unknown and, therefore, these formulations can not be used.

4.4 CMSS: Curriculum Manager for Source Selection

For the source domain that is inherently multi-modal, our goal is to learn a dynamic curriculum for selecting the best-suited samples for aligning to the target feature distribution. At the beginning of training, the Curriculum Manager is expected to prefer samples with higher *transferability* for aligning with the target, *i.e.*, source samples which have similar feature distributions to the target sample. Once the feature distributions of these samples are aligned, our Curriculum Manager is expected to prioritize the next round of source samples for alignment. As the training progresses, the Curriculum Manager can learn to focus on different aspects of the feature distribution as a proxy for better transferability. Since our approach learns a curriculum to prefer samples from different source domains, we refer to it is Curriculum Manager for Source Selection (CMSS).

Our approach builds on the domain-adversarial training framework (described in §4.3). In this framework, our hypothesis is that source samples that are hard for the domain discriminator to separate from the target samples are likely the ones that have similar feature distributions. Our CMSS leverages this and uses the discriminator loss to find source samples that should be aligned first. The preference for source samples is represented as per-sample weights predicted by CMSS. Since our approach is based on domain-adversarial training, weighing $\mathcal{L}_{\text{dom}}$ using these weights will lead to the discriminator encouraging the feature network to bring the distributions of higher weighted source samples closer to the target samples. This signal between the discriminator and feature extractor is achieved using the gradient reversal layer (see [49] for details).

Therefore, our proposed CMSS is trained to predict weights for source samples at each iteration, which maximizes the error of the domain discriminator. Due to this adversarial interplay with the discriminator, the CMSS is forced to re-estimate the preference of source samples across training to keep up with the improving domain discriminator. The feature extractor, F, is optimized to learn features that are both good for classification and confuse the discriminator. To avoid any influence from the classification task in the curriculum design, our CMSS also has an independent feature extractor module that learns to predict weights per-sample given the source images and domain discriminator loss.

4.4.1 Training CMSS:

The CMSS weight for every sample in the source domain, $\mathbf{x}_i^s$, is given by w_i^s. We represent this weighted distribution as $\tilde{\mathcal{D}}_{\text{src}}$. The CMSS network is represented by $G_\rho : \mathbb{R}^{c \times w \times h} \rightarrow \mathbb{R}^1$ with parameters ρ. Given a batch of samples, $\mathbf{x}_1^s, \mathbf{x}_2^s, \ldots \mathbf{x}_b^s$, we first pass these samples to G_ρ to obtain an array of scores that are normalized using softmax function to obtain the resulting weight vector. During training, the CMSS optimization objective can be written as

$$\min_{\rho} \left[\frac{1}{N_s} \sum_{i=1}^{N_s} G_\rho(\mathbf{x}_i^s) \log(D(F(\mathbf{x}_i^s))) \right] \tag{4.2}$$

With the source sample weights generated by CMSS, the loss function for domain discriminator can be written as

$$\mathcal{L}_{\text{wdom}} = -\frac{1}{N_s}\sum_{i=1}^{N_s} G_\rho(\mathbf{x}_i^s)\log(D(F(\mathbf{x}_i^s))) - \frac{1}{N_t}\sum_{i=1}^{N_t}\log\big(1 - D(F(\mathbf{x}_i^t))\big)$$

$$\text{s.t.} \sum_i G_\rho(\mathbf{x}_i^s) = N_s \tag{4.3}$$

The overall optimization objective can be written as

$$\max_{\psi} \min_{\theta,\phi,\rho} \mathcal{L}_{\text{cls}} - \lambda\mathcal{L}_{\text{wdom}} \tag{4.4}$$

where $\mathcal{L}_{\text{cls}}$ is the Cross-Entropy loss for source classification and $\mathcal{L}_{\text{wdom}}$ is the weighted domain discriminator loss from Eq. (4.3), with weights obtained by optimizing Eq. (4.2). λ is the hyperparameter in the gradient reversal layer. We follow [49] and set λ based on the following annealing schedule: $\lambda_p = \frac{2}{1+\exp(-\gamma \cdot p)} - 1$, where p is the current number of iterations divided by the total. γ is set to 10 in all experiments as in [49]. Details of training are provided in Algorithm 1.

4.4.2 CMSS: Theoretical Insights

We first state the classic generalization bound for domain adaptation [76, 77]. Let $\mathcal{H}$ be a hypothesis space of VC-dimension d. For a given hypothsis class $\mathcal{H}$, define the symmetric difference operator as $\mathcal{H}\Delta\mathcal{H} = \{h(\mathbf{x}) \oplus h'(\mathbf{x}) | h, h' \in \mathcal{H}\}$. Let $\mathcal{D}_{\text{src}}$, $\mathcal{D}_{\text{tgt}}$ denote the source and target distributions respectively, and $\hat{\mathcal{D}}_{\text{src}}$, $\hat{\mathcal{D}}_{\text{tgt}}$ denote the empirical distribution induced by sample of size m drawn from $\mathcal{D}_{\text{src}}$, $\mathcal{D}_{\text{tgt}}$ respectively. Let ϵ_s (ϵ_t) denote the true risk on source (target) domain, and $\hat{\epsilon}_s$ ($\hat{\epsilon}_t$) denote the empirical risk on source (target) domain. Then, following

Algorithm 1: Training CMSS

```
Input  : $N_{\text{iter}}$, $N_b^s$, $N_b^t$, $\rho$, $\psi$, $\theta$, $\phi$, $\gamma$, $\eta$
$N_{\text{iter}}$ is Total number of training iterations,
$N_b^s$ and $N_b^t$: Batch size of source and target samples,
$\theta$ is the parameters of the feature extractor,
$\phi$ is the parameter of the classifier,
$\psi$ is the parameter of the domain discriminator,
$\gamma$ is a hyper parameter,
$\eta$ is the learning rate.
for $t \leftarrow$ (1 to $N_{iter}$) do
    Obtain $\lambda = 2/(1 + \exp(-\gamma \cdot (t/N_{iter}))) - 1$
    Obtain randomly sample a source batch $\{(\mathbf{x}_i^s, y_i)\}_{i=1}^{N_b^s} \sim D_{\text{src}}$
    Obtain randomly sample a target batch $\{\mathbf{x}_i^t\}_{i=1}^{N_b^t} \sim \mathcal{D}_{\text{tgt}}$
    Update $\rho \leftarrow \rho - \eta\nabla(\min_\rho -\lambda\mathcal{L}_{\text{wdom}})$
    Update $\psi \leftarrow \psi - \eta\nabla \min_\psi \lambda\mathcal{L}_{\text{dom}}$
    Update $\theta \leftarrow \phi$ by $\theta - \eta\nabla(\min_{\theta,\phi} \mathcal{L}_{\text{cls}} - \lambda\mathcal{L}_{\text{wdom}})$
end
```

Theorem 1 of [77], with probability of at least $1 - \delta, \forall h \in \mathcal{H}$,

$$\epsilon_t(h) \leq \hat{\epsilon}_s(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\hat{\mathcal{D}}_{src}, \hat{\mathcal{D}}_{tgt}) + C \tag{4.5}$$

where C is a constant

$$C = \lambda + O\left(\sqrt{\frac{d\log(m/d) + \log(1/\delta)}{m}}\right)$$

Here, λ is the optimal combined risk (source + target risk) that can be achieved by hypothesis in $\mathcal{H}$. Let $\{\mathbf{x}_i^s\}_{i=1}^m$, $\{\mathbf{x}_i^t\}_{i=1}^m$ be the samples in the empirical distributions $\hat{\mathcal{D}}_{\text{src}}$ and $\hat{\mathcal{D}}_{\text{tgt}}$ respectively. Then, $P(\mathbf{x}_i^s) = 1/m$ and $P(\mathbf{x}_i^t) = 1/m$. The empirical source risk can be written as $\hat{\epsilon}_s(h) = 1/m \sum_i \hat{\epsilon}_{\mathbf{x}_i^s}(h)$

Now consider a CMSS re-weighted source distribution $\hat{\mathcal{D}}_{\text{wsrc}}$, with $P(\mathbf{x}_i^s) = w_i$. For $\hat{\mathcal{D}}_{\text{wsrc}}$

to be a valid probability mass function, $\sum_i w_i^s = 1$ and $w_i^s \geq 0$. Note that $\hat{\mathcal{D}}_{\text{src}}$ and $\hat{\mathcal{D}}_{\text{wsrc}}$ share the same samples, and only differ in weights. The generalization bound for this re-weighted distribution can be written as

$$\epsilon_t(h) \leq \sum_i w_i \hat{\epsilon}_{\mathbf{x}_i^s}(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\hat{\mathcal{D}}_{\text{wsrc}}, \hat{\mathcal{D}}_{\text{tgt}}) + C$$

Since the bound holds for all weight arrays $\mathbf{w} = [w_1^s, w_2^s \dots w_m^s]$ in a simplex, we can minimize the objective over $\mathbf{w}$ to get a tighter bound.

$$\epsilon_t(h) \leq \min_{\mathbf{w} \in \Delta^m} \sum_i w_i \hat{\epsilon}_{\mathbf{x}_i^s}(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\hat{\mathcal{D}}_{\text{wsrc}}, \hat{\mathcal{D}}_{\text{tgt}}) + C \tag{4.6}$$

The first term is the weighted risk, and the second term $d_{\mathcal{H}\Delta\mathcal{H}}(\hat{\mathcal{D}}_{\text{wsrc}}, \hat{\mathcal{D}}_{\text{tgt}})$ is the weighted symmetric divergence which can be realized using our weighted adversarial loss. Note that when $\mathbf{w} = [1/m, 1/m, \dots 1/m]$, we get the original bound (4.5). Hence, the original bound is in the feasible set of this optimization.

4.4.3 Relaxations.

In practice, deep neural networks are used to optimize the bounds presented above. Since the bound (4.6) is minimized over the weight vector $\mathbf{w}$, one trivial solution is to assign non-zero weights to only a few source samples. In this case, a neural network can overfit to these source samples, which could result in low training risk and low domain divergence. To avoid this trivial case, we present two relaxations:

- We use the unweighted loss for the source risk (first term in the bound (4.6)).

- For the divergence term, instead of minimizing $\mathbf{w}$ over all the samples, we optimize only over mini-batches. Hence, for every mini-batch, there is at least one w_i which is non-zero. Additionally, we make weights a function of input, *i.e.*, $w_i = G_\rho(\mathbf{x}_i^s)$, which is realized using a neural network. This will smooth the predictions of w_i, and make the weight network produce a soft-selection over source samples based on correlation with the target.

Note that the G_ρ network discussed in the previous section satisfies these criteria.

4.5 Experimental Results

In this section, we perform an extensive evaluation of the proposed method on the following tasks: digit classification(*MNIST, MNIST-M, SVHN, Synthetic Digits, USPS*), image recognition on the large-scale DomainNet dataset (*clipart, infograph, paiting, quickdraw, real, sketch*), PACS[7] (*art, cartoon, photo* and *sketch*) and Office-Caltech10 (*Amazon, Caltech, Dslr, Webcam*). We compare our method with the following contemporary approaches: Domain Adversarial Neural Network (**DANN**) [49], Multi-Domain Adversarial Neural Network (**MDAN**)[61] and two state-of-the-art discrepancy-based approaches: Maximum Classifier Discrepancy (**MCD**) [35] and Moment Matching for Multi-Source (M^3**SDA**) [59]. We follow the protocol used in other multi-source domain adaptation works [59, 60], where each domain is selected as the target domain while the rest of domains are used as source domains. For **Source Only** and **DANN** experiments, all source domains are shuffled and treated as one domain. To guarantee fairness of comparison, we used the same model architectures, batch size and data pre-processing routines for all compared approaches. All our experiments are implemented in PyTorch.

Table 4.1: **Results on Digits classification**. The proposed CMSS achieves **90.8%** accuracy. Comparisons with MCD and M^3SDA are reprinted from [59]. All experiments are based on a 3-$conv$-layer backbone trained from scratch. (mt, mm, sv, sy, up: *MNIST, MNIST-M, SVHN, Synthetic Digits, UPSP*)

Models	$mm, sv, sy, up \rightarrow mt$	$mt, sv, sy, up \rightarrow mm$	$mt, mm, sy, up \rightarrow sv$	$mt, mm, sv, up \rightarrow sy$	$mt, mm, sv, sy \rightarrow up$	Avg
Source Only	92.3 ± 0.91	63.7 ± 0.83	71.5 ± 0.75	83.4 ± 0.79	90.7 ± 0.54	80.3 ± 0.76
DANN [49]	97.9 ± 0.83	70.8 ± 0.94	68.5 ± 0.85	87.3 ± 0.68	93.4 ± 0.79	83.6 ± 0.82
MDAN [61]	97.2 ± 0.98	**75.7** ± 0.83	82.2 ± 0.82	85.2 ± 0.58	93.3 ± 0.48	86.7 ± 0.74
MCD [35]	96.2 ± 0.81	72.5 ± 0.67	78.8 ± 0.78	87.4 ± 0.65	95.3 ± 0.74	86.1 ± 0.64
M^3SDA [59]	98.4 ± 0.68	72.8 ± 1.13	81.3 ± 0.86	89.5 ± 0.56	96.1 ± 0.81	87.6 ± 0.75
CMSS (ours)	**99.0** ± 0.08	75.3 ± 0.57	**88.4** ± 0.54	**93.7** ± 0.21	**97.7** ± 0.13	**90.8** ± 0.31

4.5.1 Experiments on Digit Recognition

Following DCTN [62] and M^3SDA [59], we sample 25000 images from training subset and 9000 from testing subset of *MNIST, MNIST-M, SVHN* and *Synthetic Digits*. The entire *USPS* is used since it contains only 9298 images in total.

In all the experiments, the feature extractor is composed of three $conv$ layers and two fc layers. The entire network is trained from scratch with batch size equals 16. For each experiment, we run the same setting five times and report the mean and standard deviation. The results are shown in Table 4.1. The proposed method achieves an **90.8%** average accuracy, outperforming other baselines by a large margin (~ 3% improvement on the previous state-of-the-art approach).

4.5.2 Experiments on DomainNet

Next, we evaluate our method on **DomainNet** [59] – a large-scale benchmark dataset used for multi-domain adaptation. The DomainNet dataset contains samples from 6 domains: *Clipart*, *Infograph*, *Painting*, *Quickdraw*, *Real* and *Sketch*. Each domain has **345** categories, and the dataset has ~ **0.6 million** images in total, which is the largest existing domain adaptation dataset.

Table 4.2: **Results on the DomainNet dataset**. CMSS achieves 46.5% average accuracy. When the target domain is *quickdraw* q, CMSS is the only one that outperforms Source Only which indicates *negative transfer* has been alleviated. *Source Only* * is re-printed from [59], *Source Only* is our implemented results. All experiments are based on ResNet-101 pre-trained on ImageNet. (c: *clipart*, i: *infograph*, p: *painting*, q: *quickdraw*, r: *real*, s: *sketch*)

Models	i,p,q	c,p,q	c,i,q	c,i,p	c,i,p	c,i,p	Avg
	$r,s\rightarrow c$	$r,s\rightarrow i$	$r,s\rightarrow p$	$r,s\rightarrow q$	$q,s\rightarrow r$	$q,r\rightarrow s$	
Source Only*	47.6±0.52	13.0±0.41	38.1±0.45	13.3±0.39	51.9±0.85	33.7±0.54	32.9±0.54
Source Only	52.1±0.51	23.4±0.28	47.7±0.96	13.0±0.72	60.7±0.32	46.5±0.56	40.6±0.56
DANN [49]	60.6±0.42	25.8±0.34	50.4±0.51	7.7±0.68	62.0±0.66	51.7±0.19	43.0±0.46
MDAN [61]	60.3±0.41	25.0±0.43	50.3±0.36	8.2±1.92	61.5±0.46	51.3±0.58	42.8±0.69
MCD [35]	54.3±0.64	22.1±0.70	45.7±0.63	7.6±0.49	58.4±0.65	43.5±0.57	38.5±0.61
M^3SDA [59]	58.6±0.53	26.0±0.89	52.3±0.55	6.3±0.58	62.7±0.51	49.5±0.76	42.6±0.64
CMSS (ours)	**64.2**±0.18	**28.0**±0.20	**53.6**±0.39	**16.0**±0.12	**63.4**±0.21	**53.8**±0.35	**46.5**±0.24

We use ResNet-101 pretrained on ImageNet as the feature extractor for in all our experiments. For CMSS, we use a ResNet-18 pretrained on ImageNet. The batch size is fixed to 128. We conduct experiments over 5 random runs, and report mean and standard deviation over the 5 runs.

The results are shown in Table 4.2. CMSS achieves **46.5%** average accuracy, outperforming other baselines by a large margin. We also note that our approach achieves the best performance in each experimental setting. It is also worth mentioning that in the experiment when the target domain is *Quickdraw (q)*, our approach is the only one that outperforms Source Only baseline, while all other compared approaches result in negative transfer (lower performance than the source-only model). This is since *quickdraw* has a significant domain shift compared to all other domains. This shows that our approach can effectively alleviate negative transfer even in such challenging set-up.

Table 4.3: Results on PACS

Models	$c, p, s \to a$	$a, p, s \to c$	$a, c, s \to p$	$a, c, p \to s$	Avg
Source Only	74.9±0.88	72.1±0.75	94.5±0.58	64.7±1.53	76.6±0.93
DANN [49]	81.9±1.13	77.5±1.26	91.8±1.21	74.6±1.03	81.5±1.16
MDAN [61]	79.1±0.36	76.0±0.73	91.4±0.85	72.0±0.80	79.6±0.69
WBN [60]	**89.9**±0.28	89.7±0.56	**97.4**±0.84	58.0±1.51	83.8±0.80
MCD [35]	88.7±1.01	88.9±1.53	96.4±0.42	73.9±3.94	87.0±1.73
M^3SDA [59]	89.3±0.42	89.9±1.00	97.3±0.31	76.7±2.86	88.3±1.15
CMSS (ours)	88.6±0.36	**90.4**±0.80	96.9±0.27	**82.0**±0.59	**89.5**±0.50

4.5.3 Experiments on PACS

PACS [7] is another popular benchmark for multi-source domain adaptation. It contains 4 domains: *art, cartoon, photo* and *sketch*. Images of 7 categories are collected for each domain. There are 9991 images in total. For all experiments, we used ResNet-18 pretrained on ImageNet as the feature extractor following [60]. For the Curriculum Manager, we use the same architecture as the feature extractor. Batch size of 32 is used. We conduct experiments over 5 random runs, and report mean and standard deviation over the runs. The results are shown in Table 4.3 (a: *art*, c: *cartoon*, p: *painting*, s: *sketch*.). CMSS achieves the state-of-the-art average accuracy of **89.5%**. On the most challenging *sketch* (s) domain, we obtain **82.0%**, outperforming other baselines by a large margin.

4.5.4 Experiments on Office-Caltech10

The office-Caltech10 [78] dataset has 10 object categories from 4 different domains: *Amazon, Caltech, DSLR*, and *Webcam*. For all the experiments, we use the same architecture (ResNet-101 pretrained on ImageNet) used in [59]. The experimental results are shown in Table 4.4 (A:

Table 4.4: Results on Office-Caltech 10

Models	$A, C, D \to W$	$A, C, W \to D$	$A, D, W \to C$	$C, D, W \to A$	Avg
Source Only	99.0	98.3	87.8	86.1	92.8
DANN [49]	99.3	98.2	89.7	94.8	95.5
MDAN [61]	98.9	98.6	91.8	95.4	96.1
MCD [35]	99.5	99.1	91.5	92.1	95.6
M^3SDA [59]	99.5	99.2	92.2	94.5	96.4
CMSS (ours)	**99.6**	**99.3**	**93.7**	**96.0**	**97.2**

Amazon, C: *Caltech*, D: *Dslr*, W: *Webcam*). CMSS achieves state-of-the-art average accuracy of **97.2**%.

4.5.5 Comparison with other re-weighting methods

In this experiment, we compare CMSS with other weighing schemes proposed in the literature. We use IWAN [58] for this purpose. IWAN, originally proposed for partial domain adaption, reweights the samples in adversarial training using outputs of discriminator as sample weights (Refer to Figure 4.2). CMSS, however, computes sample weights using a separate network G_ρ updated using an adversarial game. We adapt IWAN for multi-source setup and compare it against our approach. The results are shown in Table 4.5 (abbreviations of domains same as Table 4.2). IWAN obtained 43.1% average accuracy which is close to performance obtained using DANN with combined source domains. For further analysis, we plot how sample weights estimated by both approaches (plotted as mean ± variance) change as training progresses in Figure 4.3. We observe that CMSS selects weights with larger variance which demonstrates its sample selection ability, while IWAN has weights all close to 1 (in which case, it becomes similar to DANN). This illustrates the superiority of our sample selection method. More discussions on

Table 4.5: Comparing re-weighting methods

Models	i,p,q $r,s \to c$	c,p,q $r,s \to i$	c,i,q $r,s \to p$	c,i,p $r,s \to q$	c,i,p $q,s \to r$	c,i,p $q,r \to s$	Avg
DANN [49]	60.6	25.8	50.4	7.7	62.0	51.7	43.0
IWAN [58]	59.1	25.2	49.7	12.9	60.4	51.4	43.1
CMSS (ours)	**64.2**	**28.0**	**53.6**	**16.0**	**63.4**	**53.8**	**46.5**

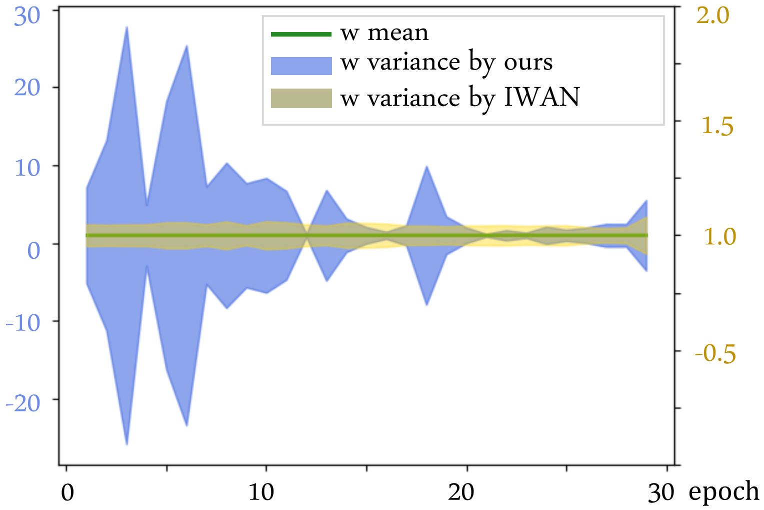

Figure 4.3: Mean and variance of generated weights over time, comparing CMSS to IWAN [58]

sample selection can be found in Section 4.6.3. CMSS also achieves a faster and more stable convergence in test accuracy compared to DANN [49] where we assume a single source domain, which further supports the effectiveness of the learnt curriculum.

4.6 Interpretations

In this section, we are interested in understanding and visualizing the source selection ability of our approach. We conduct two sets of experiments: (i) visualizations of the source selection curriculum over time, and (ii) comparison of our selection mechanism with other sample re-weighting methods.

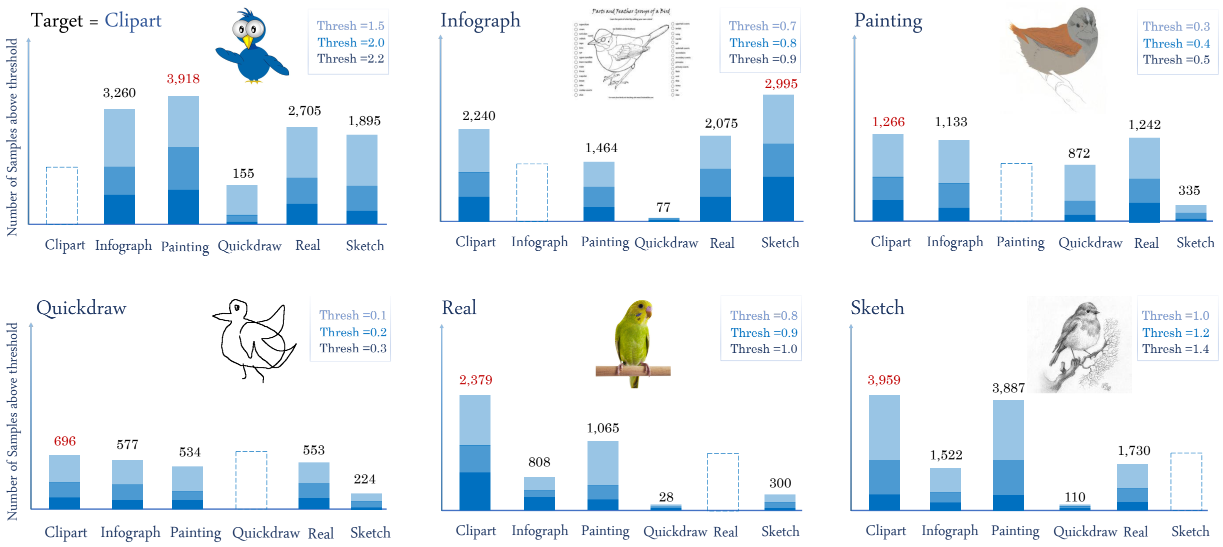

Figure 4.4: **Interpretation results of the sample selection** on DomainNet dataset using the proposed method. In each plot, one domain is selected as the target. In each setting, predictions of CMSS are computed for each sample of the source domains. The bars indicate how many of these samples have weight prediction larger than a manually chosen threshold, with each bar denoting a single source domain. Maximum number of samples are highlighted in red. *Best viewed in color*

4.6.1 Visualizations of Domain Preference

We first investigate if CMSS indeed exhibits domain preference over the course of training as claimed. For this experiment, we randomly select $m = 34000$ training samples from each source domain in DomainNet and obtain the raw weights (before softmax) generated by CMSS. Then, we calculate the number of samples in each domain passing a manually selected threshold τ. We use the number of samples passing this threshold in each domain to indicate the domain preference level. The larger the fraction, more weights are given to samples from the domains, hence, higher the domain preference. Figure 4.4 shows the visualization of domain preference for each target domain. We picked 3 different τ in each experiment for more precise observation. We observe that CMSS does display domain preference (*Clipart* - *Painting*, *Infograph* - *Sketch*,

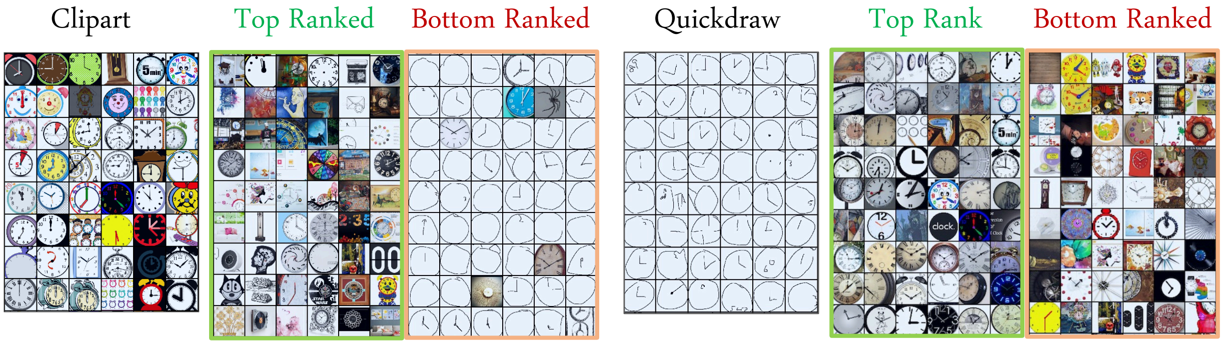

Figure 4.5: Ranked source samples according to learnt weights (class “Clock” of DomainNet dataset). *LHS*: Examples of unlabeled target domain *Clipart* and the Top/Bottom Ranked $\sim$ 50 samples of the source domain composed of *Infograph, Painting, Quickdraw, Real* and *Sketch*. *RHS*: Examples of unlabeled target domain *Quickdraw* and the Ranked samples of source domain composed of *Clipart, Infograph, Painting, Real* and *Sketch*. Weights are obtained at inference time using CMSS trained after 5 epochs.

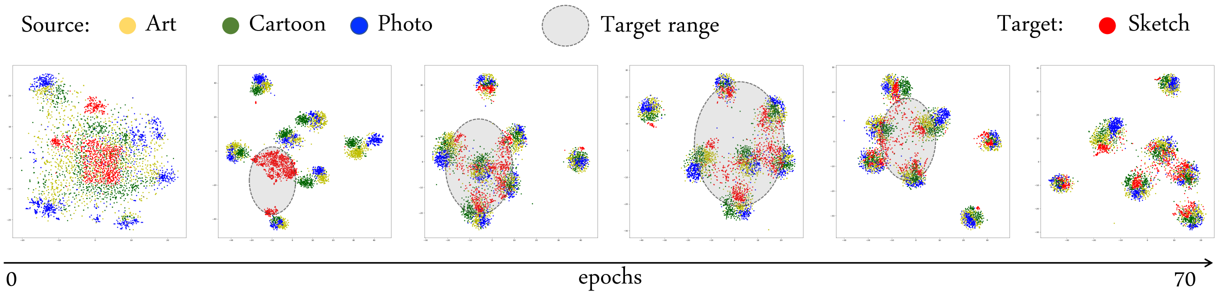

Figure 4.6: t-SNE visualization of features at six different epochs during training. The shaded region is the migrated range of target features. Dateset used is PACS with *sketch* as the target domain.

Real - *Clipart*) that is in fact correlated with the visual similarity of the domains. An exception is *Quickdraw*, where no domain preference is observed. We argue that this is because *Quickdraw* has significant domain shift compared to all other domains, hence no specific domain is preferred. However, CMSS still produces better performance on *Quickdraw*. While there is no domain preference for *Quickdraw*, there is within-domain sample preference as illustrated in Figure 4.5. That is, our approach chooses samples within a domain that are structurally more similar to the target domain of interest. Hence, just visualizing aggregate domain preference does not depict the complete picture. We will present sample-wise visualization in the next section.

4.6.2 Beyond Domain Preference

In addition to domain preference, we are interested in taking a closer look at sample-wise source selection. To do this, we first obtain the weights generated by CMSS for all source samples and rank the source images according to their weights. An example is shown in Figure 4.5. For better understanding, we visualize samples belonging to a fixed category ("Clock" in Figure 4.5).

In Figure 4.5, we find that notion of similarity discovered by CMSS is different for different domains. When the target domain is *Clipart* (left panel of Figure 4.5), source samples with colors and cartoonish shapes are ranked at the top, while samples with white background and simplistic shapes are ranked at the bottom. When the target is *Quickdraw* (right panel of Figure 4.5), one would think that CMSS will simply be selecting images with similar white background. Instead, it prefers samples which are structurally similar to the regular rounded clock shape (as most samples in *Quickdraw* are similar to these). It thus appears that structural similarity is favored in *Quickdraw*, whereas color information is preferred in *Clipart*. This provides support that CMSS selects samples according to ease of alignment to the target distribution, which is automatically discovered per domain. We argue that this property of CMSS has an advantage over approaches such as MDAN [61] which simply weighs manually partitioned domains.

4.6.3 Selection Over Time

In this section, we discuss how source selection varies as training progresses. In Figure 4.3, we plot mean and variance of weights (output of Curriculum Manager) over training iterations. We observe that the variance is high initially, which indicates many samples have weights away from the mean value of 1. Samples with higher weights are preferred, while those with low weights contribute less to the alignment. In the later stages, the variance is very low which indicates most of the weights are close to 1. Hence, our approach gradually adapts to increasingly many source samples over time, naturally learning a curriculum for adaptation. In Figure 4.6, we plot a t-SNE visualization of features at different epochs. We observe that the target domain *sketch* (red) first adapts to *Art* (yellow), and then gradually aligns with *Cartoon* (green) and *Photo* (blue).

4.7 Conclusion

In this chapter, we proposed Curriculum Manager for Source Selection (CMSS) that learns a curriculum for Multi-Source Unsupervised Domain Adaptation. A curriculum is learnt that iteratively favors source samples that align better with the target distribution over the entire training. The curriculum learning is achieved by an adversarial interplay with the discriminator, and achieves state-of-the-art on four benchmark datasets. We also shed light on the inner workings of CMSS, and we hope that will pave the way for further advances to be made in this research area.

Chapter 5: Deep Co-Training with Task Decomposition for Semi-supervised Domain Adaptation

Semi-supervised domain adaptation (SSDA) aims to adapt models trained from a labeled source domain to a different but related target domain, from which unlabeled data and a small set of labeled data are provided. Current methods that treat source and target supervision without distinction overlook their inherent discrepancy, resulting in a source-dominated model that has not effectively use the target supervision. In this paper, we argue that the labeled target data needs to be distinguished for effective SSDA, and propose to explicitly decompose the SSDA task into two sub-tasks: a semi-supervised learning (SSL) task in the target domain and an unsupervised domain adaptation (UDA) task across domains. By doing so, the two sub-tasks can better leverage the corresponding supervision and thus yield very different classifiers. To integrate the strengths of the two classifiers, we apply the well established co-training framework, in which the two classifiers exchange their high confident predictions to iteratively "teach each other" so that both classifiers can excel in the target domain. We call our approach **De**ep **Co**-training with **Ta**sk decomposition (**DeCoTa**). DeCoTa requires no adversarial training and is easy to implement. Moreover, DeCoTa is well founded on the theoretical condition of when co-training would succeed. As a result, DeCoTa achieves state-of-the-art results on several SSDA datasets, outperforming the prior art by a notable 4% margin on DomainNet.

5.1 Introduction

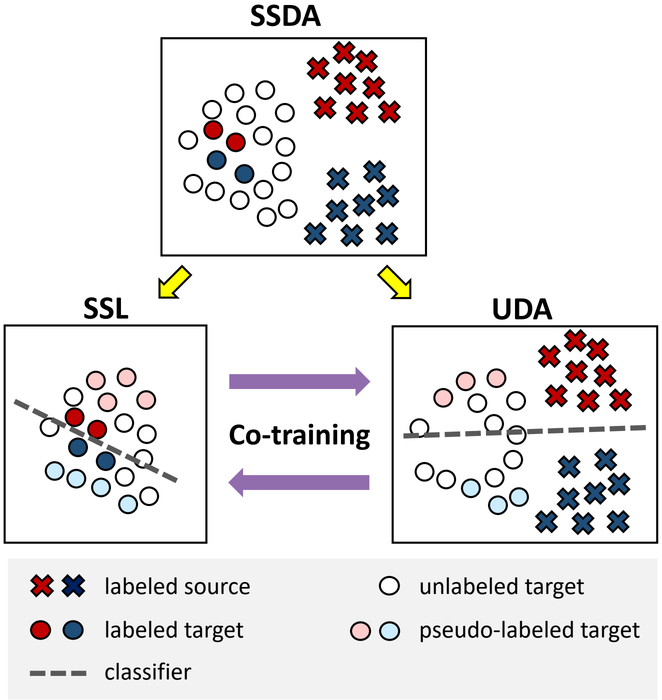

Figure 5.1: **Deep Co**-training with **Ta**sk decomposition (**DeCoTa**). We decompose semi-supervised domain adaptation (SSDA) into two sub-tasks: semi-supervised learning (SSL) in the target domain, and unsupervised DA (UDA) across domains. The two sub-tasks offer different pseudo-label confidences to the unlabeled data (light blue & light red circles), which we leverage via co-training: exchanging their high confident predictions to teach each other.

Domain adaptation (DA) aims to adapt machine learned models from a source domain to a related but different target domain [10, 35, 52, 79]. DA is particularly important in settings where labeled target data is hard to obtain, but labeled source data is plentiful [8, 80, 81], *e.g.*, , adaptation from synthetic to real images [81, 82, 83, 84, 85] and adaptation to a new or rare environment [56, 86, 87, 88]. Most of the existing works focus on the unsupervised domain adaptation (UDA) setting, in which the target domain is completely unlabeled. Several recent works, however, show that adding merely a tiny amount of target labeled data (*e.g.*, , just one labeled image per class) can notably boost the performance [32, 89, 90, 91, 92, 93, 94, 95], suggesting that this setting may be more promising for domain adaptation to succeed. In this paper, we thus focus on the latter setting, which is referred to as semi-supervised domain adaptation (SSDA).

Despite the seemingly nuanced difference between the two settings, methods that are effective for SSDA and UDA can vary substantially. For instance, [32] showed that directly combining the labeled source and labeled target data and then applying popular UDA algorithms like domain adversarial learning [10] or entropy minimization [96] can hardly improve the performance. In other words, the labeled target data have not been effectively used. Existing methods [32, 89, 90] therefore propose additional objectives to strengthen the influence of labeled target data in SSDA.

Intrigued by these findings, we investigate the characteristics of SSDA further and emphasize two fundamental challenges. First, the amount of labeled source data is much larger than that of labeled target data. Second, the two data are inherently different in their distributions. **A single classifier** learned together with both sources of supervision is thus easily dominated by the labeled source data and is unable to take advantage of the additional labeled target data.

To resolve this issue, we propose to explicitly decompose the two sources of supervision and learn **two distinct classifiers** whose goals are however shared: to classify well on the unlabeled target data. To this end, we pair the labeled source data and the unlabeled target data to learn one classifier, which is essentially a UDA task. For the other classifier, we pair the labeled and unlabeled target data, which is essentially a semi-supervised learning (SSL) task. That is, **we explicitly decompose SSDA into two well-studied tasks.**

For each sub-task, one may apply any existing algorithms independently. In this paper, we however investigate the idea of learning the two classifiers jointly for two compelling reasons. First, the two tasks share the same goal and same unlabeled data, meaning that they are *correlated*.

Second, learning with distinct labeled data implies that the two classifiers will converge differently in what types of mistakes they make and on which samples they are confident and correct, meaning that they are *complementary* to each other.

We therefore propose to learn the two classifiers jointly via co-training [97, 98, 99][1], which is arguably one of the most established algorithm for learning with multi views: in our case, two correlating and complementary tasks. The approach is straightforward: train a separate classifier on each task using its labeled data, and use them to create pseudo-labels for the unlabeled data. As the two classifiers are trained with distinct supervision, they will yield different predictions. In particular, there will be samples that only one classifier is confident about (and more likely to be correct). By labeling these samples with the confident classifier's predictions and adding them to the training set of the other classifier to re-train on, the two classifiers are essentially "teaching each other" to improve. To this end, we employ a simple *pseudo-labeling-based algorithm with deep learning*, similar to [101], to train each classifier. Pseudo-labeling-based algorithms have been shown powerful for both the UDA and SSL tasks [102, 103]. In other words, we can apply the same algorithm for both sub-tasks, greatly simplifying our overall framework which we name **DECOTA**: **De**ep **Co**-training with **Ta**sk Decomposition.

We evaluate DECOTA on two benchmark datasets for SSDA: DomainNet [8] and Office-home [104]. While very simple to implement and without any adversarial training [32, 90], DECOTA significantly outperforms the state-of-the-art results [89, 90] on DomainNet by over 4% and is on a par with them on Office-home. *We attribute this to the empirical evidence that our task decomposition fits the theoretical condition of relaxed ϵ-expandability [98, 99], which*

[1] We note that, co-training [97] and co-teaching [100] share similar concepts but are fundamentally different. See 5.2 for a discussion.

is sufficient for co-training to succeed. Another strength of DeCoTa is that it requires no extra learning process like feature decomposition to create views from data [99, 105, 106]. To the best of our knowledge, our paper is the first to enable deep learning with co-training on SSDA.

The contributions of this work are as follow. (1) We explicitly decompose the two very different sources of supervision, labeled source and labeled target data, in SSDA. (2) We present DeCoTa, a simple deep learning based co-training approach for SSDA to jointly learn two classifiers, one for each supervision. (3) we provide intermediate results and insights that illustrate why DeCoTa works. Specifically, we show that DeCoTa satisfies the ϵ-expandability requirement [98] of co-training. (4) Lastly, we support this work with strong empirical results that outperform state-of-the-art.

5.2 Related Work

Unsupervised domain adaptation (UDA). UDA has been studied extensively. Many methods [34, 107, 108] matched the feature distributions between domains by minimizing their divergence. One mainstream approach is by domain adversarial learning [10, 14, 81, 87, 109, 110, 111, 112]. More recent works [34, 35, 36, 113] learn features based on the cluster assumption [96]: classifier boundaries should not cross high density target data regions. For example, [35, 113] attempted to push target features away from the boundary, using minmax training. Some other approaches employ self-training with pseudo-labeling [114, 115, 116, 117] to progressively label unlabeled data and use them to fine-tune the model [103, 106, 118, 119, 120, 121, 122, 123]. A few recent methods use MixUp [124], but mainly to augment adversarial learning based UDA approaches (*e.g.*, , [10]) by stabilizing the domain discriminator [111, 125] or smoothing the predictions [126, 127]. In contrast, we apply MixUp to create better pseudo-labeled data for co-training, without adversarial learning.

Semi-supervised domain learning (SSDA). SSDA attracts less attention in DA, despite its promising scenario in balancing accuracy and labeling effort. With few labeled target data, SSDA can quickly reshape the class boundaries to boost the accuracy [32, 90]. Many SSDA works are proposed prior to deep learning [92, 95, 128, 129], matching features while maintaining accuracy on labeled target data. [91, 130] employed knowledge distillation [131] to regularize the training on labeled target data. More recent works use deep learning, and find that the popular UDA principle of aligning feature distributions could fail to learn discriminative class boundaries in SSDA [32]. [32] thus proposed to gradually move the class prototypes (used to derive class boundaries) to the target domain in a minimax fashion; [90] introduced opposite structure learning to cluster target data and scatter source data to smooth the process of learning class boundaries. Both works [32, 90] and [89] concatenate the target labeled data with the source data to expand the labeled data. [93] incorporates meta-learning to search for better initial condition in domain adaptation. SSDA is also related to [132, 133], in which active learning is incorporated to label data for improving domain adaptation.

Co-training. Co-training, a powerful semi-supervised learning (SSL) method proposed in [97], looks at the available data with two views from which two models are trained interactively. By adding the confident predictions of one model to the training set of the other, co-training enables the models to "teach each other". There were several assumptions to ensure co-training's effectiveness [97], which were later relaxed by [98] with the notion of ϵ-expandability. [99] broadened the scope of co-training to a single-view setting by learning to decompose a fixed feature representation into two artificially created views; [106] subsequently extended this framework to use co-training for (semi-supervised) domain adaptation[2]. A recent work [105] extended co-training to deep learning models, by encouraging two models to learn different features and behave differently on single-view data. One novelty of DECOTA is that it works with single-view data (both the UDA and SSL tasks are looking at images) but requires no extra learning process like feature decomposition to artificially create views from such data [99, 105, 106].

Co-training vs. co-teaching. Co-teaching [100] was proposed for learning with noisy data, which shares a similar procedure to co-training by learning two models to filter out noisy data for each other. There are several key differences between them and **DECOTA** *is based on co-training*. As in [100], co-teaching is designed for supervised learning with noisy labels, while co-training is for learning with unlabeled data by leveraging two views. DECOTA decomposes SSDA into two tasks (two views) to leverage their difference to improve the performance — the core concept of co-training [106]. In contrast, co-teaching does not need two views. Further, co-teaching relies on the memorization of neural nets to select small loss samples to teach the other classifiers, while DECOTA selects high confident ones from unlabeled data.

5.3 Deep Co-training with Task Decomposition

5.3.1 Approach Overview

Co-training strategies have traditionally been applied to data with two views, e.g., audio and video, or webpages with HTML source and link-graph, after which a classifier is trained in each view and they teach each other on the unlabeled data. This is the original formulation from Blum and Mitchell [97], which is later extended to single-view data by [99] for linear models

[2] Similar to [32, 90], [106] simply concatenated the target labeled data with the source data to expand the labeled data.

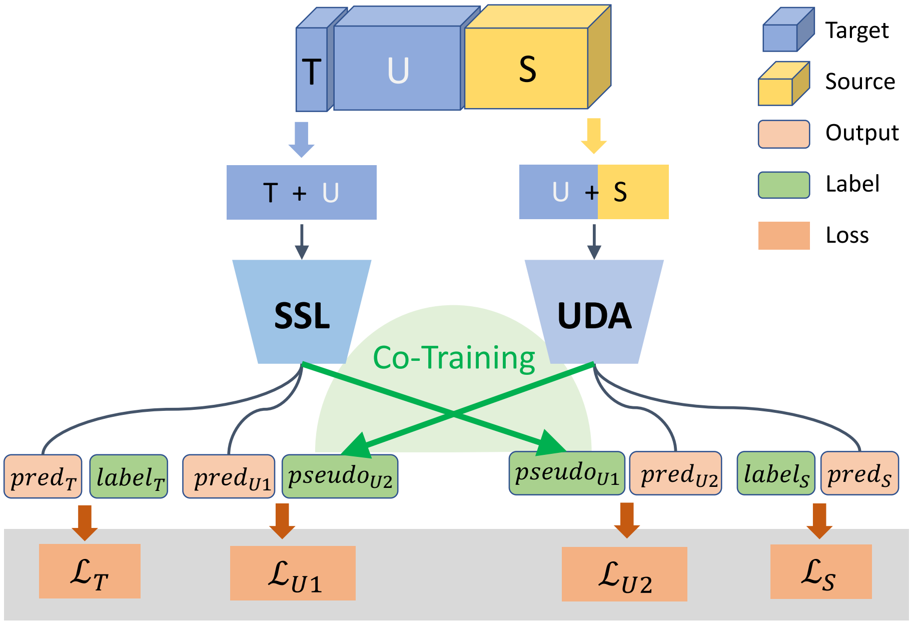

Figure 5.2: The overall framework of **DeCoTa**. It decomposes the SSDA task into SSL and UDA tasks that exchange pseudo-labels for unlabeled target U.

and by [105] for deep neural networks. Both methods require additional objective functions or tasks (*e.g.*, , via generating adversarial examples [134]) to learn to create *artificial* views such that co-training can be applied.

In this paper, we have however discovered that in semi-supervised domain adaptation (SSDA), one can actually conduct co-training using single-view data (all are images) without such an additional learning subroutine. The key is to leverage the inherent discrepancy of the labeled data (*i.e.*, , supervision) provided in SSDA: the labeled data from the source domain, $D_{\mathcal{S}} = \{(s_i, y_i)\}_{i=1}^{N_S}$, and the labeled data from the target domain, $D_{\mathcal{T}} = \{(t_i, y_i)\}_{i=1}^{N_T}$, which is usually much smaller than $D_{\mathcal{S}}$. By combining each of them with the unlabeled samples from the target domain, $D_{\mathcal{U}} = \{u_i\}_{i=1}^{N_U}$, we can construct two sub-tasks in SSDA Fig. 5.2 and Fig. 5.3):

- an **unsupervised domain adaptation (UDA)** task that trains a model $\boldsymbol{w}_g$ using $D_{\mathcal{S}}$ and $D_{\mathcal{U}}$,
- a **semi-supervised learning (SSL)** task that trains another model $\boldsymbol{w}_f$ using $D_{\mathcal{T}}$ and $D_{\mathcal{U}}$.

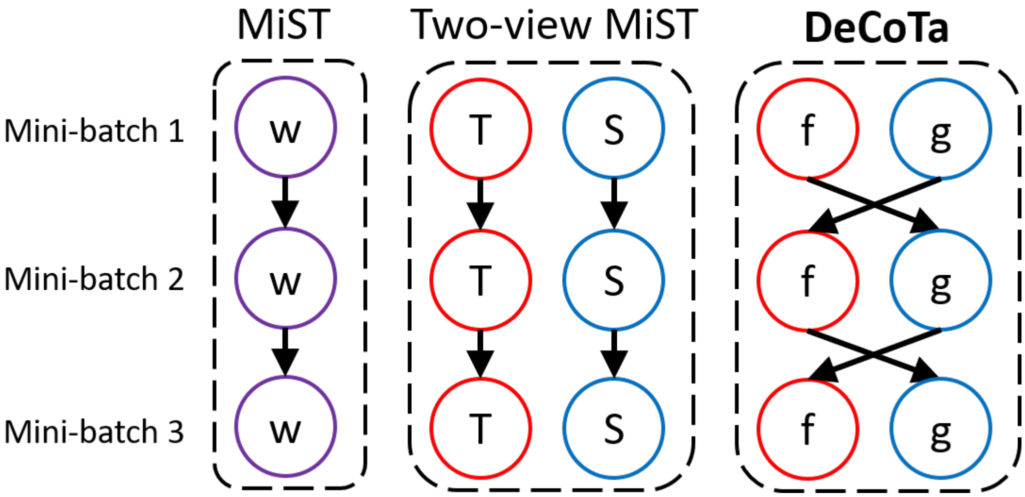

Figure 5.3: Comparison among **MiST**, two-view **MiST** (*i.e.*, , two-task **MiST**), and **DeCoTa**. The color on the circles means the labeled data: red for $D_{\mathcal{T}}$, blue for $D_{\mathcal{S}}$, and purple for both. The arrows indicate which model provides the pseudo-labels for which model to learn from.

We learn both models by mini-batch stochastic gradient descent (SGD). At every iteration, we sample three data sets, $S = \{(s_b, y_b)\}_{b=1}^B$ from $D_{\mathcal{S}}$, $T = \{(t_b, y_b)\}_{b=1}^B$ from $D_{\mathcal{T}}$, and $U = \{u_b\}_{b=1}^B$ from $D_{\mathcal{U}}$, where B is the mini-batch size. We can then predict on U using the the two models $\boldsymbol{w}_g$ and $\boldsymbol{w}_f$, creating the pseudo-label sets $U^{(f)}$ and $U^{(g)}$ that will be used to update $\boldsymbol{w}_f$ and $\boldsymbol{w}_g$,

$$\begin{aligned}
U^{(f)} =& \{(u_b, \hat{y}_b = \arg\max_c p(c|u_b; \boldsymbol{w}_g)); \\
& \text{if } \max_c p(c|u_b; \boldsymbol{w}_g) > \tau\}, \\
U^{(g)} =& \{(u_b, \hat{y}_b = \arg\max_c p(c|u_b; \boldsymbol{w}_f)); \\
& \text{if } \max_c p(c|u_b; \boldsymbol{w}_f) > \tau\},
\end{aligned} \tag{5.1}$$

where u_b is an unlabeled sample drawn from U, $p(c|u_b; \cdot)$ is the predicted probability for a class c, and τ is the threshold for pseudo-label selection. In other words, we use one model's (say $\boldsymbol{w}_g$)

Algorithm 2: The **DeCoTa** algorithm

```
Input : $\boldsymbol{w}_f$ and $\boldsymbol{w}_g$, learning rate $\eta$, batch size $B$, iteration $N_{\max}$, beta distribution coefficient $\alpha$,
        confidence threshold $\tau$, data $D_{\mathcal{S}}$, $D_{\mathcal{T}}$, $D_{\mathcal{U}}$;
for $n \leftarrow 1$ to $N_{\max}$ do
    Sample $S = \{(s_b, y_b)\}_{b=1}^{B}$ from $D_{\mathcal{S}}$,
    Sample $T = \{(t_b, y_b)\}_{b=1}^{B}$ from $D_{\mathcal{T}}$,
    Sample $U = \{u_b\}_{b=1}^{B}$ from $D_{\mathcal{U}}$;
    Set $U^{(f)} = \emptyset$, $U^{(g)} = \emptyset$;
    for $b \leftarrow 1$ to $B$ do
        if $\max_c p(c|u_b; \boldsymbol{w}_g) > \tau$ then
            Update $U^{(f)} \leftarrow U^{(f)} + \{(u_b, \hat{y}_b)\}$, $\hat{y}_b = \arg\max_c p(c|u_b; \boldsymbol{w}_g)$;
        end
        if $\max_c p(c|u_b; \boldsymbol{w}_f) > \tau$ then
            Update $U^{(g)} \leftarrow U^{(g)} + \{(u_b, \hat{y}_b)\}$, $\hat{y}_b = \arg\max_c p(c|u_b; \boldsymbol{w}_f)$;
        end
    end
    Obtain $\tilde{U}^{(f)} = \{\textsc{MixUp}(U_i^{(f)}, T_i; \alpha)\}_{i=1}^{|U^{(f)}|}$;
    Obtain $\tilde{U}^{(g)} = \{\textsc{MixUp}(U_i^{(g)}, S_i; \alpha)\}_{i=1}^{|U^{(g)}|}$;
    Update $\boldsymbol{w}_f \leftarrow \boldsymbol{w}_f - \eta \left(\nabla\mathcal{L}(\boldsymbol{w}_f, T) + \nabla\mathcal{L}(\boldsymbol{w}_f, \tilde{U}^{(f)})\right)$;
    Update $\boldsymbol{w}_g \leftarrow \boldsymbol{w}_g - \eta \left(\nabla\mathcal{L}(\boldsymbol{w}_g, S) + \nabla\mathcal{L}(\boldsymbol{w}_g, \tilde{U}^{(g)})\right)$;
end
Output: $\boldsymbol{w}_f$ and $\boldsymbol{w}_g$ (for model ensemble).
```

high confident prediction to create pseudo-labels for u_b, which is then included in $U^{(f)}$ that will be used to train the other model $\boldsymbol{w}_f$. By looking at $U^{(f)}$ and $U^{(g)}$ jointly, we are indeed asking one model to simultaneously be a *teacher* and a *student*: it provides confident pseudo-labels for the other model to learn from, and learns from the other model's confident pseudo-labels.

We call this approach **DeCoTa**, which stands for Deep Co-training with Task Decomposition. In the following, we will discuss how to improve the pseudo-label quality (*i.e.*, , its coverage and accuracy) for **DeCoTa**, and provide in-depth analysis why DeCoTa works.

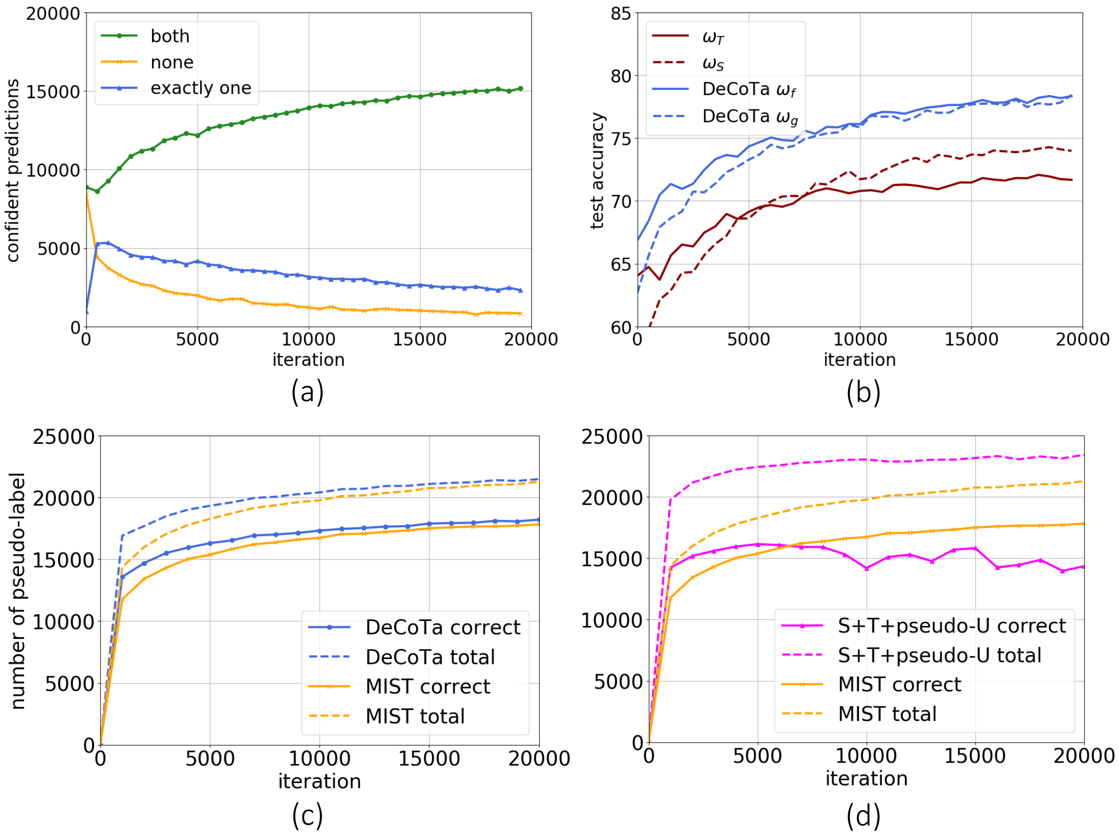

Figure 5.4: **Analysis on the two-task decomposition.** We use DomainNet [8] (Real to Clipart; three-shot). (a) We show the number of test examples that *both*, *exactly one*, and *none* of the models have high confidence on (in total, $18,325$). The two tasks hold unique expertise (*i.e.*, , there is a 14% portion of the data that exactly one view is confident on), satisfying the condition of co-training in Eq. (5.6). (b) We show the power of co-training: the same tasks without co-training perform worse, indicating that the models benefit from each other. The analysis is on DomainNet (R to C; three-shot) and we will clarify it. We further analyze pseudo-labels in (c) and (d). For every 1K iterations (*i.e.*, , 24K unlabeled data with possible repetition), we accumulate the number of data that have confident (> 0.5) and correct predictions by at least one classifier. (c) Comparison of pseudo-label quantity and quality using DeCoTa vs. MiST. (d) MiST vs. self-training (S+T+pseudo-U). It can be observed that DeCoTa has the largest number of correct pseudo-labels.

5.3.1.1 DeCoTa with High-quality Pseudo-labels

The pseudo-labels acquired from each model are understandably noisy. At the beginning of the training, this problem is especially acute, and affects the efficacy of the model as the training progresses. Our experience shows that mitigation is necessary to handle noise in the

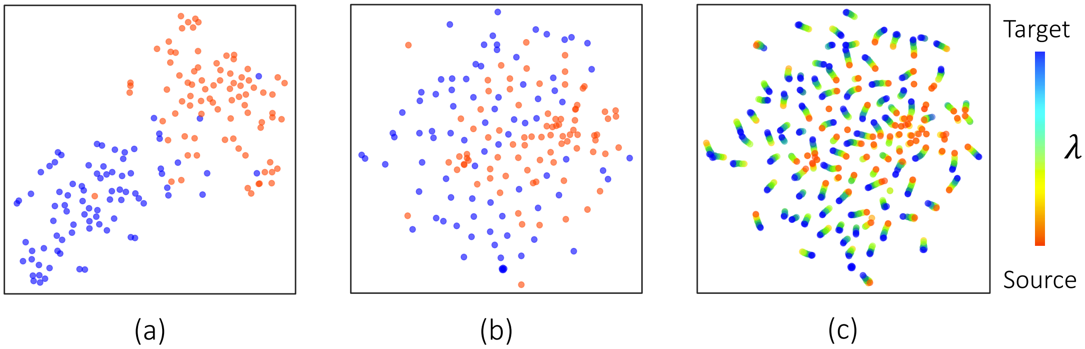

Figure 5.5: t-SNE visualization of S (red dots, sampled from $D_{\mathcal{S}}$) and U (blue dots, sampled from $D_{\mathcal{U}}$): (a) before and (b) after including MixUp in calculating the projection; (c) t-SNE of S, U, and MixUp(S, U). We see a clear data transition along λ.

pseudo-labels to further enhance **DeCoTa**, for which we follow recent works of SSL [101] to apply MixUp [124, 135]. MixUp is an operation to construct virtual examples by convex combinations. Given two labeled examples (x_1, y_1) and (x_2, y_2), we define MixUp $((x_1, y_1), (x_2, y_2); \alpha)$

$$\lambda \sim \text{Beta}(\alpha, \alpha),$$

$$\tilde{x} = (1 - \lambda)x_1 + \lambda x_2, \qquad \tilde{y} = (1 - \lambda)\boldsymbol{e}_{y_1} + \lambda \boldsymbol{e}_{y_2} \tag{5.2}$$

to obtain a virtual example $(\tilde{x}, \tilde{y})$, where $\boldsymbol{e}_y$ is a one-hot vector with the y^{th} element being 1. λ controls the degree of MixUp while Beta refers to the standard beta distribution.

We perform MixUp between labeled and pseudo-labeled data: *i.e.*, , between samples in $U^{(f)}$ and T, and between samples in $U^{(g)}$ and S to obtain two sets of virtual examples $\tilde{U}^{(f)}$ and $\tilde{U}^{(g)}$. We then update $\boldsymbol{w}_f$ and $\boldsymbol{w}_g$ by SGD,

$$\boldsymbol{w}_g \leftarrow \boldsymbol{w}_g - \eta \left(\nabla \mathcal{L}(\boldsymbol{w}_g, S) + \nabla \mathcal{L}(\boldsymbol{w}_g, \tilde{U}^{(g)}) \right), \tag{5.3}$$

$$\boldsymbol{w}_f \leftarrow \boldsymbol{w}_f - \eta \left(\nabla \mathcal{L}(\boldsymbol{w}_f, T) + \nabla \mathcal{L}(\boldsymbol{w}_f, \tilde{U}^{(f)}) \right),$$

where η is the learning rate and $\mathcal{L}$ is the averaged loss over examples. We use the cross-entropy loss.

In our experiments, we have found that MIXUP can

- effectively **denoise** an incorrect pseudo-label by mixing it with a correct one (from S or T). The resulting $\tilde{y}$ at least contains a λ portion of correct labels;
- smoothly **bridge** the domain gap between U and S. This is done by interpolating between $U^{(g)}$ and S. The resulting $\tilde{x}$ can be seen as an intermediate example between domains.

In other words, MIXUP encourages the models to behave linearly between accurately labeled and pseudo-labeled data, which reduces the undesirable oscillations caused by noisy pseudo-labels and stabilizes the predictions across domains. We note that, our usage of MIXUP is fundamentally different from [111, 125, 126, 127] that employed MIXUP as auxiliary losses to augment existing DA algorithms like [10].

We illustrate this in Fig. 5.5. A model pre-trained on D_S is used to generate feature embeddings. We then employ t-SNE [136] to perform two tasks simultaneously, namely clustering the embedded samples as well as projecting them into a 2D space for visualization. In (a), only S sampled from D_S and U sampled from $D_{\mathcal{U}}$ are embedded, while in (b) and (c), additional samples from MIXUP of S and U were added to the fold to influence t-SNE's clustering step. (b) shows only the finally projected S and U samples afterwards while (c) shows the additional projected MIXUP samples as a function of λ. One can easily see that MIXUP effectively closes the gap between the source and target domain.

5.3.2 Constraints for Effective Co-training

In DECOTA, we perform co-training via a decomposition of tasks on single-view data. To explain further why DECOTA works, we provide analysis in this subsection on the difference made by splitting the SSDA problem into two tasks for co-training. That is, we would like to verify that the decomposition leads to two tasks that fit into the assumption of co-training [98]. To begin with, we train two models: one model, $\boldsymbol{w}_S$, is trained with S and $\tilde{U}^{(S)}$ while the other model, $\boldsymbol{w}_T$, is trained with T and $\tilde{U}^{(T)}$. $\tilde{U}^{(S)}$ is obtained from applying $\boldsymbol{w}_S$ to U for pseudo-labels, follow by MIXUP with S. The same definition goes for $\tilde{U}^{(T)}$. Essentially, both the UDA and SSL task prepare their own pseudo-labels *independently* using their respective model in a procedure that is similar to self-training [114, 115, 116, 117].

Table 5.1: Comparing with deep co-training methods [105] for SSDA on DomainNet, 3-shot. (See Section 5.4 for details.)

Method	R to C	R to P	P to C	C to S	S to P	R to S	P to R	Mean
Deep Co-Training [105] w/o MIXUP	73.7	67.6	73.2	63.9	66.7	64.1	79.3	69.7
Deep Co-Training [105] with MIXUP	74.2	69.1	72.3	64.1	67.9	65.1	79.4	70.3
DECOTA	80.4	75.2	78.7	68.6	72.7	71.9	81.5	75.6

After training, we apply $\boldsymbol{w}_T$ to the entire $D_{\mathcal{U}}$ and compute for each $u \in D_{\mathcal{U}}$ the binary confidence indicator

$$h_T(u) = \begin{cases} 1 & \text{if } \max_c p(c|u; \boldsymbol{w}_T) > \tau, \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

Here, high confident examples will get a value 1, otherwise 0. We also apply $\boldsymbol{w}_S$ to $D_{\mathcal{U}}$ to obtain $h_S(u)$. Denote by $\bar{h}_T(u) = 1 - h_T(u)$ the not function of $h_T(u)$, we compute the following three indicators to summarize the entire $D_{\mathcal{U}}$

$$\begin{aligned} h_{\text{both}} &: \sum_{u \in D_{\mathcal{U}}} h_T(u) h_S(u), \\ h_{\text{one}} &: \sum_{u \in D_{\mathcal{U}}} h_T(u) \bar{h}_S(u) + \bar{h}_T(u) h_S(u), \\ h_{\text{none}} &: \sum_{u \in D_{\mathcal{U}}} \bar{h}_T(u) \bar{h}_S(u), \end{aligned} \tag{5.5}$$

corresponding to the number of examples that *both*, *exactly one*, and *none* of the models have high confidence on, respectively. Intuitively, if the two models are exactly the same, h_{one} will be 0, meaning that they are either both confident or not on an example. On the contrary, if the two models are well optimized but hold their specialties, both h_{one} and h_{both} will be of high values and h_{none} will be low.

We ran the study on DomainNet [8], in which we use *Real* as source and *Clipart* as target. We consider a 126-class classification problem, in which $|D_{\mathcal{S}}| = 70,358$, $|D_{\mathcal{U}}| = 18,325$, and $|D_{\mathcal{T}}| = 378$ (*i.e.*, , a three-shot setting where each class in the target domain is given three labeled samples). We initialize $\boldsymbol{w}_S$ and $\boldsymbol{w}_T$ with a ResNet [137] pre-trained on $D_{\mathcal{S}}$, and evaluate Eq. (5.4) and Eq. (5.5) every 500 iterations (with a $\tau = 0.5$ confidence threshold in selecting pseudo-labels.).

Fig. 5.4 (a) shows the results. *The two models do hold their specialties (*i.e., *, yield different high-confident predictions)*. Even at the end of training, there is a 14% portion of data that one model is confident on but not the other (the blue curve). Thus, if we can properly fuse their specialties during training — one model provides the pseudo-labels to the data on which the other model is uncertain — we are likely to jointly learn stronger models at the end.

This is indeed the core idea of our co-training proposal. Theoretically, the two "views" (or, tasks in our case) must satisfy certain conditions, *e.g.*, , ϵ-expandability [98]. [99, 106] relaxed it and only needed the expanding condition to hold on average in the unlabeled set, which can be formulated as follows, using h_{both}, h_{one}, and h_{none}

$$h_{\text{one}} \geq \epsilon \min(h_{\text{both}}, h_{\text{none}}). \tag{5.6}$$

To satisfy Eq. (5.6), there must be sufficient examples that exactly one model is confident on so that the two models can benefit from teaching each other. Referring to Fig. 5.4 (a) again, our two tasks consistently hold a ϵ around 2 after the first 500 iterations (*i.e.*, , after the models start to learn the task-specific idiosyncrasies), suggesting the feasibility of applying co-training to our decomposition. The power of co-training is clearly illustrated in Fig. 5.4 (b). The two models without co-training, $\underline{\boldsymbol{w}_T \text{ and } \boldsymbol{w}_S}$, perform worse than their co-training counterparts, $\underline{\boldsymbol{w}_f \text{ and } \boldsymbol{w}_g}$ (see Section 5.3.1, Eq. (5.1), Eq. (5.3)), even using the same architecture and data.

5.3.3 Comparing to Other Co-training Approaches

With our approach outlined, it is worthwhile to contrast DECOTA with prior co-training work in domain adaptation. In particular, DECOTA is notably different from the approach known as Co-training for DA (CODA) [106]. While CODA also utilizes co-training for SSDA using single-view data, it differs from DECOTA fundamentally as follow:

1. CODA takes a feature-centric view in that the two *artificial* views in its co-training procedure are constructed by decomposing the feature dimensions into two mutually exclusive subsets. DECOTA on the other hand achieves effective co-training with a two-task decomposition.

2. The two views in CODA do not exchange high confident pseudo-labels in a mini-batch fashion like DECOTA. Nor does CODA utilize MIXUP, which we have shown to be valuable for SSDA. Instead, CODA explicitly conducts feature alignment by minimizing the difference between the distributions of the source and target domains.

3. CODA trains a logistic regression classifier. In the era of deep learning, while co-training has been used in multiple vision tasks, DECOTA is *the first work in SSDA* utilizing deep learning, co-training, and mixup in a cohesive and principled fashion, achieving state of the art performance.

Since CODA is not deep learning based, to further justify the efficacy of DECOTA, we took the deep co-training work described in [105] that was designed for semi-supervised image recognition, and customize it for SSDA. [105] constructs multi-views for co-training via two different adversarial perturbations on the same image samples, after which the two networks are trained to make different mistakes on the same adversarial examples. For fair comparison, we compare [105] both with and without MIXUP, using the DomainNet [8] dataset. The results are given in Table 5.1. DECOTA outperforms [105] by a margin.

5.4 Experiments

We consider the one-/three-shot settings, following [32], where each class is given one or three labeled target examples. We train with $D_{\mathcal{S}}$, $D_{\mathcal{T}}$, and unlabeled $D_{\mathcal{U}}$. We then reveal the true label of $D_{\mathcal{U}}$ for evaluation.

Datasets. We use **DomainNet** [8], a large-scale benchmark dataset for domain adaptation that has 345 classes and 6 domains. We follow [32], using a 126-class subset with 4 domains (*i.e.*, , R:

Table 5.2: Accuracy on DomainNet (%) for three-shot setting with 4 domains, using ResNet-34.

Method	R to C	R to P	P to C	C to S	S to P	R to S	P to R	Mean
S+T	60.8	63.6	60.8	55.6	59.5	53.3	74.5	61.2
DANN [10]	62.3	63.0	59.1	55.1	59.7	57.4	67.0	60.5
ENT [32]	67.8	67.4	62.9	50.5	61.2	58.3	79.3	63.9
MME [32]	72.1	69.2	69.7	59.0	64.7	62.2	79.0	68.0
UODA [90]	75.4	71.5	73.2	64.1	69.4	64.2	80.8	71.2
APE [89]	76.6	72.1	76.7	63.1	66.1	67.8	79.4	71.7
ELP [138]	74.9	72.1	74.4	64.3	69.7	64.9	81.0	71.6
DeCoTa	**80.4**	**75.2**	**78.7**	**68.6**	**72.7**	**71.9**	**81.5**	**75.6**

Table 5.3: Accuracy on Office-Home (%) for three-shot setting with 4 domains, using VGG-16.

Method	R to C	R to P	R to A	P to R	P to C	P to A	A to P	A to C	A to R	C to R	C to A	C to P	Mean
S+T	49.6	78.6	63.6	72.7	47.2	55.9	69.4	47.5	73.4	69.7	56.2	70.4	62.9
DANN [10]	56.1	77.9	63.7	73.6	52.4	56.3	69.5	50.0	72.3	68.7	56.4	69.8	63.9
ENT [32]	48.3	81.6	65.5	76.6	46.8	56.9	73.0	44.8	75.3	72.9	59.1	77.0	64.8
MME [32]	56.9	82.9	65.7	76.7	53.6	59.2	75.7	54.9	75.3	72.9	61.1	76.3	67.6
UODA [90]	57.6	83.6	67.5	**77.7**	54.9	**61.0**	77.7	**55.4**	**76.7**	73.8	61.9	**78.4**	68.9
APE [89]	56.0	81.0	65.2	73.7	51.4	59.3	75.0	54.4	73.7	71.4	61.7	75.1	66.5
ELP [138]	57.1	83.2	67.0	76.3	53.9	59.3	75.9	55.1	76.3	73.3	61.9	76.1	68.0
DeCoTa	**59.9**	**83.9**	**67.7**	77.3	**57.7**	60.7	**78.0**	54.9	76.0	**74.3**	**63.2**	**78.4**	**69.3**

Real, C: Clipart, P: Painting, S: Sketch.) and report 7 different adaptation scenarios. We also use **Office-Home** [104], another benchmark that contains 65 classes, with 12 adaptation scenarios constructed from 4 domains (*i.e.*, , R: Real world, C: Clipar t, A: Art, P: Product).

Implementation details. We implement using Pytorch [139]. We follow [32] to use ResNet-34 [137] on DomainNet and VGG-16 [140] on Office-Home. We also provide ResNet-34 results on Office-Home in order to fairly compare with [89] in supplementary. The networks are pre-trained on ImageNet [141, 142]. We follow [32, 143] to replace the last linear layer with a K-way cosine classifier (*e.g.*, , $K = 126$ for DomainNet) and train it at a fixed temperature (0.05 in all our experiments). We initialize $\boldsymbol{w}_f$ with a model first fine-tuned on $D_{\mathcal{S}}$, and initialize $\boldsymbol{w}_g$ with a model first fine-tuned on $D_{\mathcal{S}}$ and then fine-tuned on $D_{\mathcal{T}}$. We do so to encourage the two

Table 5.4: Ablation Study (three shots). (a)-(b): comparison of MiST and DeCoTa and the vanilla ensemble of two independently trained MiST; (c): comparison of Two-view MiST (without co-training) and DeCoTa; (d) comparison of MiST and S+T+pseudo-U without MixUp; (e) each model of DeCoTa on the source domain test data, comparing to supervised training on source (S), average of DomainNet. All accuracy in (%).

(a) Comparing MiST, Vanilla-Ensemble of two MiST (with different initialization), and DeCoTa on DomainNet

Method	R to C	R to P	P to C	C to S	S to P	R to S	P to R	Mean
MiST	78.1	75.2	76.7	68.3	72.6	71.5	79.8	74.6
Vanilla-Ensemble	79.7	75.0	77.2	68.4	72.1	70.8	79.7	74.7
DeCoTa	80.4	75.2	78.7	68.6	72.7	71.9	81.5	75.6

(b) Comparing MiST, Vanilla-Ensemble of two MiST (with different initialization), and DeCoTa on Office-Home

Method	R to C	R to P	R to A	P to R	P to C	P to A	A to P	A to C	A to R	C to R	C to A	C to P	Mean
MiST	54.7	81.2	64.0	69.4	51.7	58.8	69.1	47.6	70.6	65.3	60.8	73.8	63.9
Vanilla-Ensemble	56.1	81.8	63.4	72.9	54.1	55.1	74.2	49.5	72.1	67.4	55.2	75.6	64.7
DeCoTa	59.9	83.9	67.7	77.3	57.7	60.7	78.0	54.9	76.0	74.3	63.2	78.4	69.3

(c) Comparing the decomposed tasks trained independently to using DeCoTa

Method	Task	R to C	R to P	P to C	C to S	S to P	R to S	P to R	Mean
Decomposed tasks (without co-training)	$\boldsymbol{w}_f$	72.1	65.7	71.8	61.0	63.0	59.9	75.9	67.0
	$\boldsymbol{w}_g$	76.3	72.2	70.3	63.7	69.4	66.9	76.1	70.7
	Ensemble	77.3	72.0	75.1	65.7	69.3	66.1	78.7	72.0
DeCoTa	$\boldsymbol{w}_f$	80.1	74.6	78.6	68.4	72.5	71.2	81.1	75.2
	$\boldsymbol{w}_g$	80.0	74.5	78.4	68.3	72.2	71.3	80.6	75.0
	Ensemble	80.4	75.2	78.7	68.6	72.7	71.9	81.5	75.6

(d) Comparing MiST and the S+T+pseudo-U with no MixUp on DomainNet

Method	R to C	R to P	P to C	C to S	S to P	R to S	P to R	Mean
S+T+pseudo-U	70.0	67.2	68.3	57.2	61.1	58.7	71.2	65.6
MiST	78.1	75.2	76.7	68.3	72.6	71.5	79.8	74.6

(e) Accuracy on source domain

$\boldsymbol{w}_f$	$\boldsymbol{w}_g$	**DeCoTa**	**S**
65.3	98.2	93.5	98.8

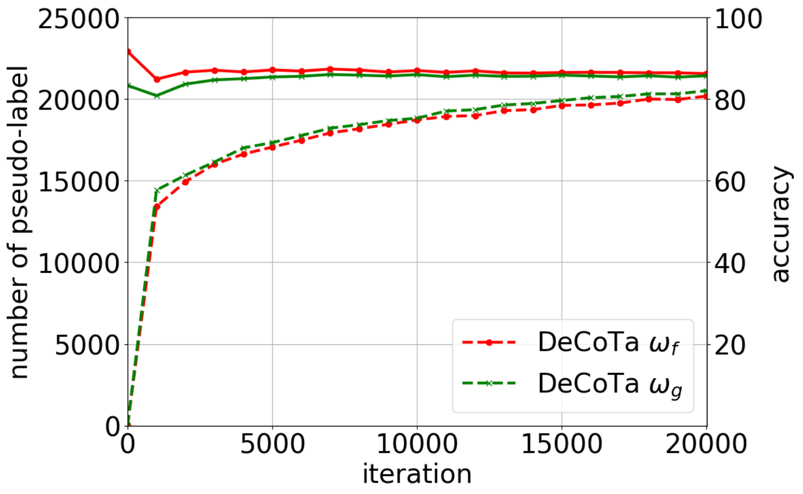

Figure 5.6: Number (dashed, left) and accuracy (solid, right) of pseudo-labels on DomainNet three-shot setting, Real to Clipart.

models to be different at the beginning. At each iteration, we sample three mini-batches $S \subset D_{\mathcal{S}}$, $T \subset D_{\mathcal{T}}$, and $U \subset D_{\mathcal{U}}$ of equal sizes $B = 24$ (cf. Section 5.3.1.1). We set the confidence threshold $\tau = 0.5$, and beta distribution coefficient $\alpha = 1.0$. We use SGD with momentum of 0.9 and an initial learning rate of 0.001, following [32]. We train for 50K/10K iterations on DomainNet/Office-Home. We note that, DeCoTa does not increase the training time since at each iteration, it only updates and learns from the pseudo-labels of the current mini-batch of unlabeled data, not the entire unlabeled data.

Baselines. We compare to four state-of-the-art SSDA approaches, **MME** [32], **UODA** [90], **APE** [89], and **ELP** [138]. We also compare to **S+T**, a model trained with $D_{\mathcal{S}}$ and $D_{\mathcal{T}}$, without using $D_{\mathcal{U}}$. Additionally, we compare to **DANN** [10] (domain adversarial learning) and **ENT** [96] (entropy minimization), both of which are important prior work on UDA. We modify them such that $D_{\mathcal{S}}$ and $D_{\mathcal{T}}$ are used jointly to train the classifier, following [32]. We denote by **S** the model trained only with the source data $D_{\mathcal{S}}$.

Variants of our approach. We consider variants of our approach for extensive ablation studies.

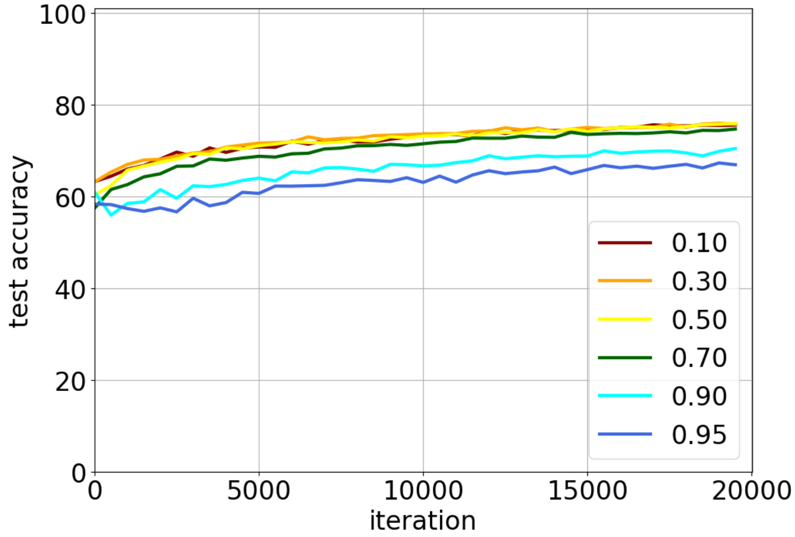

Figure 5.7: **DeCoTa**'s sensitivity to pseudo-label threshold τ on DomainNet three-shot setting, Real to Clipart.

We first introduce a model we called MixUp Self-Training (MiST). MiST is trained as follows

$$\begin{aligned} \boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \nabla \mathcal{L}(\boldsymbol{w}, S) + \nabla \mathcal{L}(\boldsymbol{w}, T) \\ + \nabla \mathcal{L}(\boldsymbol{w}, \tilde{U}_S^{(\boldsymbol{w})}) + \nabla \mathcal{L}(\boldsymbol{w}, \tilde{U}_T^{(\boldsymbol{w})})), \end{aligned} \tag{5.7}$$

where $\tilde{U}_S^{(\boldsymbol{w})}$ and $\tilde{U}_T^{(\boldsymbol{w})}$ are pseudo-labels obtained from $\boldsymbol{w}$, followed by MixUp with S and T, respectively. MiST basically lumps all the pseudo and hard labeled samples together during training, and is intended for comparing with the effect of co-training. **S+T+pseudo-U** is the model trained with self-training, but without MixUp. **Two-view MiST** is the direct ensemble of independently trained models, one for each view, using MiST (cf. Section 5.3.2). **Vanilla-Ensemble** is the ensemble model by combining two MiST trained on $D_{\mathcal{S}}$, $D_{\mathcal{T}}$, and $D_{\mathcal{U}}$ but with different initialization. For all the variants that train only one model, we initialize it with a pre-trained model fine-tuned on $D_{\mathcal{S}}$ and then fine-tuned on $D_{\mathcal{T}}$. Otherwise, we initialize the two models in the same way as DeCoTa. *We note that, for any methods that involve two models, we*

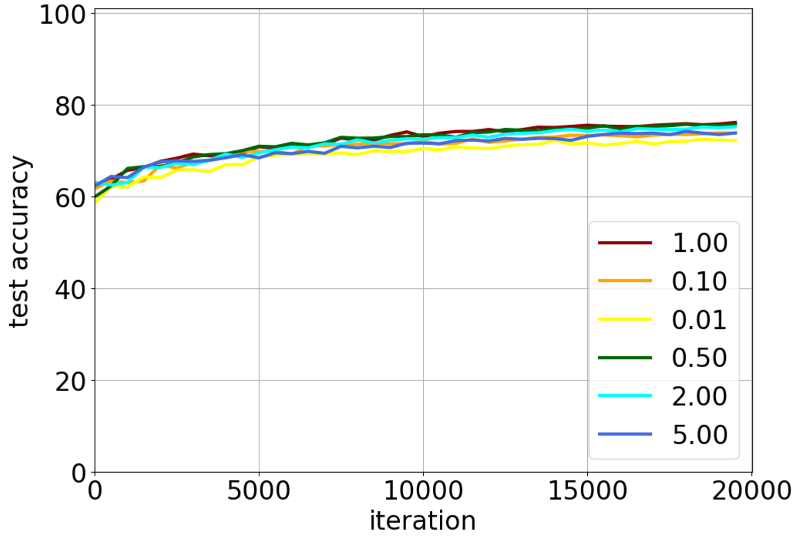

Figure 5.8: **DECOTA**'s sensitivity to the Beta distribution coefficient α on DomainNet three-shot setting, Real to Clipart.

perform ensemble on their output probability.

Main results. We summarize the comparison with baselines in Table 5.2 and Table 5.3. We mainly report the three-shot results and leave the one-shot results in the supplementary material. DECOTA outperforms other methods by a large margin on DomainNet, and outperforms all methods on Office-Home (mean). The smaller gain on Office-Home may be due to its smaller data size and limited scenes. DomainNet is larger and more diverse; the significant improvement on it is a stronger indicator of the effectiveness of our algorithm.

We further provide detailed analysis on DECOTA. We mainly report the DomainNet three-shot results. Other detailed results can be found in the supplementary material.

Task decomposition. We first compare DECOTA to MIST. As shown in Table 5.4 (a)-(b), DECOTA outperforms MIST by 1% on DomainNet and 5% on Office-Home on the three-shot setup. Fig. 5.4 (c) further shows the number of pseudo-labels involved in model training (those with confidence larger than $\tau = 0.5$). We see that DECOTA always generates more pseudo-

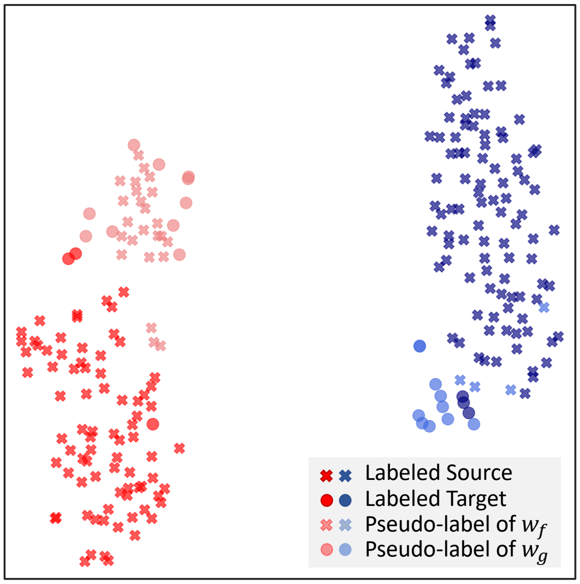

Figure 5.9: t-SNE visualization of pseudo-labels assigned by $\boldsymbol{w}_f$ and $\boldsymbol{w}_g$ in **DECOTA** (see text for details).

label data with a higher accuracy than MIST (also in Fig. 5.4 (b)), justifying our claim that the decomposition helps keep $D_{\mathcal{S}}$'s and $D_{\mathcal{T}}$'s specialties, producing high confident predictions on more unlabeled data as a result.

Co-training. We compare DECOTA to **two-view MIST**. Both methods decompose the data into a SSL and a UDA task. The difference is in how the pseudo-label set was generated (cf. Eq. (5.1)): **Two-view MIST** constructs each set independently (cf. Section 5.3.2). DECOTA outperforms two-view MIST by a margin, not only on ensemble, but also on each view alone, justifying the effectiveness of two models exchanging their specialties to benefit each other. As in Table 5.4 (c), each model of DECOTA outperforms MIST.

MIXUP. We examine the importance of MIXUP. Specifically, we compare MIST and **S+T+pseudo-U**. The second model trains in the same way as MIST, except that it does not apply MIXUP. On DomainNet (3-shot), MIST outperforms **S+T+pseudo-U** by **9%** on average. We attribute this difference to the *denoising* effect by MIXUP: MIXUP is performed after the pseudo-label set is defined, so it does not directly affect the number of pseudo-labels, but the quality. We further calculate the number of correctly assigned pseudo-labels along training, as shown in Fig. 5.4 (d). With MIXUP, the correct pseudo-label pool boosts consistently. In contrast, **S+T+pseudo-U** reinforces itself with wrongly assigned pseudo-labels; the percentage thus remains constantly low. Comparison results are shown in Table 5.4 (d).

Comparison to vanilla model ensemble. Since DECOTA combines $\boldsymbol{w}_f$ and $\boldsymbol{w}_g$ in making predictions, for a fair comparison we train two MIST models (both use $D_{\mathcal{S}} + D_{\mathcal{T}} + D_{\mathcal{U}}$), each with different initialization, and perform model ensemble. As shown in Table 5.4 (a)-(b), DECOTA outperforms this vanilla model ensemble, especially on Office-Home, suggesting that our improvement does not simply come from model ensemble, but from co-training.

On the "two-classifier-convergence" problem [144]. DECOTA is based on co-training and thus does not suffer the problem. This is shown in Table 5.4 (a, b): MIST and Vanilla-Ensemble are based on self-training and DECOTA outperformed them. Even at the end of training when two classifiers have similar accuracy (see Table 5.4 (c)), combining them still boosts the accuracy: *i.e.*, , they make different predictions.

Results on the source domain. While $\boldsymbol{w}_f$ and $\boldsymbol{w}_g$ have similar accuracy on $D_{\mathcal{U}}$, the fact that $\boldsymbol{w}_f$ does not learn from $D_{\mathcal{S}}$ suggest their difference in classifying source domain data. We verify this in Table 5.4 (e), where we apply each model individually on a hold-out set from the source domain (provided by DomainNet). We see that $\boldsymbol{w}_g$ clearly dominates $\boldsymbol{w}_f$. Its accuracy is even on a par with a model trained only on $D_{\mathcal{S}}$, showing one advantage of DECOTA— the model can keep its discriminative ability on the source domain.

Main results on the one-shot setting. We report the comparison with baselines in the one-shot setting on DomainNet in Table 5.5 and Office-Home in Table 5.6. **DECOTA** outperforms the

Table 5.5: Accuracy on DomainNet (%) for the one-shot setting with four domains, using ResNet-34.

Method	R to C	R to P	P to C	C to S	S to P	R to S	P to R	Mean
S+T	58.1	61.8	57.7	51.5	55.4	49.1	73.1	58.1
DANN [10]	61.2	62.3	56.4	54.0	57.9	55.9	65.6	59.0
ENT [32]	60.0	60.2	54.9	48.3	55.8	49.4	74.4	57.6
MME [32]	69.5	68.1	64.4	56.7	62.0	59.2	76.9	65.3
UODA [90]	72.7	70.3	69.8	60.5	66.4	62.7	77.3	68.5
APE [89]	70.4	70.8	72.9	56.7	64.5	63.0	76.6	67.6
ELP [138]	72.8	70.8	72.0	59.6	66.7	63.3	77.8	69.0
DeCoTa	**79.1**	**74.9**	**76.9**	**65.1**	**72.0**	**69.7**	**79.6**	**73.9**

Table 5.6: Accuracy on Office-Home (%) for the one-shot setting with four domains, using VGG-16.

Method	R to C	R to P	R to A	P to R	P to C	P to A	A to P	A to C	A to R	C to R	C to A	C to P	Mean
S+T	39.5	75.3	61.2	71.6	37.0	52.0	63.6	37.5	69.5	64.5	51.4	65.9	57.4
DANN [10]	**52.0**	75.7	62.7	72.7	45.9	51.3	64.3	44.4	68.9	64.2	52.3	65.3	60.0
ENT [32]	23.7	77.5	64.0	74.6	21.3	44.6	66.0	22.4	70.6	62.1	25.1	67.7	51.6
MME [32]	49.1	78.7	65.1	74.4	46.2	56.0	68.6	45.8	72.2	68.0	57.5	71.3	62.7
UODA [90]	49.6	79.8	**66.1**	75.4	45.5	**58.8**	**72.5**	43.3	**73.3**	70.5	**59.3**	72.1	**63.9**
ELP [138]	49.2	79.7	65.5	75.3	46.7	56.3	69.0	**46.1**	72.4	68.2	67.4	71.6	63.1
DeCoTa	47.2	**80.3**	64.6	**75.5**	**47.2**	56.6	71.1	42.5	73.1	**71.0**	57.8	**72.9**	63.3

state-of-the-art methods by 4.9% on DomainNet (ResNet-34), while performs slightly worse than [90] by 0.6% on Office-Home (VGG-16). Nevertheless, **DeCoTa** attains the highest accuracy on 5 adaptation scenarios of Office-Home in the one-shot setting.

Office-Home results on other backbones We report the comparison with baselines on Office-Home using a ResNet-34 backbone in Table 5.7, following [89][3]. **DeCoTa** attains the state-of-the-art result.

Results on Office-31 We report the comparison with available baseline results on Office-31 [146] in Table 5.8, using ResNet-34 backbone. Following [32], two adaptation scenarios are compared

[3] Most existing papers only reported Office-Home results using VGG-16. We followed [89] to further report ResNet-34. Some algorithms reported in Table 5.3 are missing in Table 5.7 since they do not release code.

Table 5.7: Accuracy on Office-Home (%) for the three-shot setting with four domains, using ResNet-34.

Method	R to C	R to P	R to A	P to R	P to C	P to A	A to P	A to C	A to R	C to R	C to A	C to P	Mean
S+T	55.7	80.8	67.8	73.1	53.8	63.5	73.1	54.0	74.2	68.3	57.6	72.3	66.2
DANN [10]	57.3	75.5	65.2	69.2	51.8	56.6	68.3	54.7	73.8	67.1	55.1	67.5	63.5
ENT [32]	62.6	85.7	70.2	79.9	60.5	63.9	79.5	61.3	79.1	76.4	64.7	79.1	71.9
MME [32]	64.6	85.5	71.3	80.1	64.6	65.5	79.0	63.6	79.7	76.6	67.2	79.3	73.1
APE [89]	66.4	86.2	73.4	82.0	65.2	66.1	81.1	63.9	80.2	76.8	66.6	79.9	74.0
DeCoTa	**70.4**	**87.7**	**74.0**	**82.1**	**68.0**	**69.9**	**81.8**	**64.0**	**80.5**	**79.0**	**68.0**	**83.2**	**75.7**

Table 5.8: SSDA results on Office-31, on two scenarios (following [32]).

Method	Webcam (W) to Amazon (A)		DSLR (D) to Amazon (A)	
	1-shot	3-shot	1-shot	3-shot
S+T	69.2	73.2	68.2	73.3
DANN [10]	69.3	75.4	70.4	74.6
ENT [32]	69.1	75.4	72.1	75.1
MME [32]	73.1	76.3	73.6	77.6
Ours	**76.0**	**76.8**	**74.2**	**78.3**

(Webcam to Amazon, DSLR to Amazon). Our approach DeCoTa consistently outperforms the compared methods.

Larger-shot results We provide *10,20,50*-shot SSDA results on DomainNet in Table 5.9. We randomly select and add additional samples per class from the target domain to the target labeled pool. As a semi-supervised setting, we compared with both domain adaptation (DA) and semi-supervised learning (SSL) baselines [145]. The implementation details are the same as those of *1,3*-shot. DeCoTa improves along with more shots and can outperform baselines.

Numbers and accuracy of pseudo-labels We showed the number of total and correct pseudo-labels by the two classifiers of **DeCoTa** along the training iterations in Fig. 5.4(c). The analysis is on DomainNet three-shot setting, from Real to Clipart. Concretely, for every $1K$ iterations (*i.e.*, , 24K unlabeled data), we accumulated the number of unlabeled data that have confident (with

Table 5.9: Results on DomainNet at *10, 20, 50*-shot, using ResNet-34. We tune hyper-parameters for SSL methods similarly to DA methods.

	R to C			R to P			P to C			C to S			S to P			R to S			P to R			Mean		
n-shot →	10	20	50	10	20	50	10	20	50	10	20	50	10	20	50	10	20	50	10	20	50	10	20	50
S+T	69.1	72.4	77.5	67.3	70.2	73.4	68.2	72.5	77.7	62.9	67.3	71.8	64.8	67.9	72.6	61.3	65.5	70.2	78.0	79.3	82.2	67.4	70.7	75.1
DANN [10]	66.2	68.0	71.1	65.1	67.1	69.0	62.4	64.5	68.2	60.0	62.4	66.8	61.3	63.8	67.6	61.4	63.2	66.9	71.6	74.7	78.1	64.0	66.2	69.7
ENT [32]	77.9	80.0	83.0	72.3	74.9	77.7	77.5	79.1	82.3	66.3	70.1	75.0	66.3	71.0	75.7	63.9	68.3	74.6	**81.2**	**82.9**	84.5	72.2	75.2	79.0
MME [32]	77.0	78.5	80.9	71.9	74.0	76.4	75.6	76.9	80.4	65.9	68.6	72.5	68.6	70.9	74.4	66.7	69.7	72.7	80.8	82.2	83.3	72.4	74.4	77.2
Mixup [124]	73.4	79.5	83.1	68.3	72.2	75.4	75.0	79.5	83.1	63.7	69.4	75.0	68.5	72.4	76.2	62.9	69.9	75.0	78.8	82.3	**84.7**	70.1	75.0	78.9
FixMatch [145]	76.6	79.5	82.3	73.0	74.7	76.4	75.8	79.4	83.3	70.1	73.1	76.9	71.3	73.3	77.0	68.7	71.6	74.2	79.7	81.9	84.2	73.6	76.2	79.2
DeCoTa	**81.8**	**82.6**	**85.0**	**75.1**	**76.6**	**78.7**	**81.3**	**81.7**	**84.5**	**73.7**	**75.3**	**78.0**	**73.4**	**75.7**	**77.7**	**73.7**	**75.5**	**77.8**	80.7	80.1	83.9	**77.1**	**78.2**	**80.8**

Table 5.10: Comparison between **DeCoTa** and **MiST**: test accuracy on DomainNet and Office-Home dataset (%).

(a) DomainNet

Setting	Method	R to C	R to P	P to C	C to S	S to P	R to S	P to R	Mean
1-shot	**MiST**	74.8	73.6	74.5	65.0	72.0	67.0	77.6	72.1
	DeCoTa	79.1	74.9	76.9	65.1	72.0	69.7	79.6	73.9
3-shot	**MiST**	78.1	75.2	76.7	68.3	72.6	71.5	79.8	74.6
	DeCoTa	80.4	75.2	78.7	68.6	72.7	71.9	81.5	75.6

(b) Office-Home

Setting	Method	R to C	R to P	R to A	P to R	P to C	P to A	A to P	A to C	A to R	C to R	C to A	C to P	Mean
1-shot	**MiST**	42.7	77.5	62.9	73.1	39.4	54.8	67.1	40.0	66.9	67.9	56.8	69.4	59.9
	DeCoTa	47.2	80.3	64.6	75.5	47.2	56.6	71.1	42.5	73.1	71.0	57.8	72.9	63.3
3-shot	**MiST**	54.7	81.2	64.0	69.4	51.7	58.8	69.1	47.6	70.6	65.3	60.8	73.8	63.9
	DeCoTa	59.9	83.9	67.7	77.3	57.7	60.7	78.0	54.9	76.0	74.3	63.2	78.4	69.3

confidence $> \tau = 0.5$) and correct predictions by at least one classifier. We further plot them independently for each classifier (*i.e.*, , $\boldsymbol{w}_f$ and $\boldsymbol{w}_g$) in Fig. 5.6. The accuracy of pseudo-labels remains stable (*i.e.*, , the number of confident and correct predictions divided by the number of confident predictions) but the number increases along training.

Task decomposition We report the comparison of **DeCoTa** and **MiST** on DomainNet and Office-Home in all the adaptation scenarios. As shown in Table 5.10, **DeCoTa** outperform **MiST** on all the setting by $1 \sim 2\%$ on DomainNet and $3 \sim 5\%$ on Office-Home, which further confirms the effectiveness of task decomposition — explicitly considering the discrepancy between the two sources of supervision — in **DeCoTa**.

Table 5.11: Comparison between **DECOTA** and **one-direction teaching**: accuracy on DomainNet (%) three-shot setting.

Method	R to C	R to P	P to C	C to S	S to P	R to S	P to R	Mean
$\boldsymbol{w}_f$ teaching	73.8	67.2	73.7	63.1	65.9	61.7	78.2	69.1
$\boldsymbol{w}_g$ teaching	77.5	74.5	74.2	64.8	71.6	69.0	79.0	72.9
DECOTA	80.4	75.2	78.7	68.6	72.7	71.9	81.5	75.6

Table 5.12: Comparison on the source domain test data of DomainNet (%). Here we compare the two-task models of **DECOTA** in the three-shot setting to the source-only model (S).

Method	R to C	R to P	P to C	C to S	S to P	R to S	P to R	Mean
$\boldsymbol{w}_f$	55.2	68.2	43.8	59.5	50.8	56.9	61.0	56.3
$\boldsymbol{w}_g$	97.2	97.1	99.3	98.7	98.9	96.8	99.4	98.2
S	98.1	98.2	99.5	98.9	99.2	98.2	99.6	98.8

One-direction training We further consider another variant of **DECOTA** named **one-direction teaching**, in which only one task teaches the other. Instead of co-training, we use either $\boldsymbol{w}_f$ or $\boldsymbol{w}_g$ to generate pseudo-labels for both tasks[4], while keeping the other setups the same as **DECOTA**. This study is designed to measure the complementary specialties of the two tasks. As shown in Table 5.11, the performance drops notably by using one-direction teaching. The results suggest that the two tasks provide unique expertise and complement each other, instead of one dominating the other.

Results on the source domain We report the results on the source domain test set using $\boldsymbol{w}_f$ and $\boldsymbol{w}_g$ of **DECOTA** on DomainNet (three-shot) in Table 5.12. While $\boldsymbol{w}_f$ and $\boldsymbol{w}_g$ have similar accuracy on the target domain test set, the fact that $\boldsymbol{w}_f$ does not learn from D_S suggests their difference in classifying source domain data. Table 5.12 confirms this: we see that $\boldsymbol{w}_g$ clearly dominates $\boldsymbol{w}_f$. Its accuracy is even on a par with a model trained only on D_S, showing one advantage of DECOTA— the model can keep its discriminative ability on the source domain.

[4] That is, **one-direction teaching** constructs both pseudo-label sets, *i.e.*, , $U^{(f)}$ and $U^{(g)}$ in Equation 5.1 of the main text, by the same model (we hence have two versions, $\boldsymbol{w}_f$ teaching or $\boldsymbol{w}_g$ teaching).

Sensitivity to the confidence threshold τ We investigate **DECOTA**'s sensitivity to the confidence threshold τ for assigning pseudo-labels (cf. Eq. (5.1) and Eq. (5.4)). As shown in Fig. 5.7, the variance in accuracy is small when $\tau \leq 0.7$. The accuracy drops notably when $\tau \geq 0.9$. We surmise that it is due to too few pseudo-labeled data are picked under a high threshold.

Analysis on the Beta distribution coefficient α Fig. 5.8 shows **DECOTA**'s sensitivity to the MIXUP hyper-parameter α in Eq. (5.2): α is the coefficient of the Beta distribution, which influences the sampled value of λ, an indicator of the "propotion" in the MIXUP algorithm. We report **DECOTA**'s result on DomainNet three-shot setting, adapting from Real to Clipart. The best performance is achieved by $\alpha = 1.0$, equivalent to a uniform distribution of $\lambda \in [0, 1]$. This result is consistent with our hypothesis that MIXUP connects the source and target domains with interpolated feature spaces in-between.

Training time DECOTA does not increase the training time much for two reasons. First, at each iteration (*i.e.*, , mini-batch), it only updates and learns from the pseudo-labels of *the current mini-batch* of unlabeled data, not the entire unlabeled data. Second, assigning pseudo-labels only requires a forward pass of the mini-batch, just like most domain adaptation algorithms normally do to compute training losses. The only difference is that DECOTA trains two classifiers and needs to perform the forward pass of unlabeled data twice.

t-SNE visualizations on DECOTA tasks We visualize $D_{\mathcal{S}}$, $D_{\mathcal{T}}$, and the $D_{\mathcal{U}}$ pseudo-labels by each task of **DECOTA** in Fig. 5.9. For clarity, we select two classes for illustration. The colors blue and red represent the two classes; the shapes circle and cross represent data from $D_{\mathcal{T}}$ (labeled target data) and $D_{\mathcal{S}}$ (labeled source data), respectively. The colors light blue and light red represent the pseudo-labels of each class on $D_{\mathcal{U}}$, in which the shape circle indicates that the pseudo-labels are provided by $\boldsymbol{w}_f$ (learned with $D_{\mathcal{T}}$) and the shape cross indicates that the pseudo-labels are provided by $\boldsymbol{w}_g$ (learned with D_S). The visualization is based on DomainNet three-shot setting, from Real to Clipart, trained for $10,000$ iterations. We see that $\boldsymbol{w}_f$ tends to assign pseudo-labels to unlabeled data whose features are closer to $D_\mathcal{T}$; $\boldsymbol{w}_g$ tends to assign pseudo-labels to unlabeled data whose features are closer to D_S. Such a behavior is aligned with the seminal work of semi-supervised learning by [147].

5.5 Conclusion

We introduce **DECOTA**, a simple yet effective approach for semi-supervised domain adaptation (SSDA). Our key contribution is the novel insight that the two sources of supervisions (*i.e.*, , the labeled target and labeled source data) are inherent different and should not be combined directly. DECOTA thus explicitly decomposes SSDA into two tasks (*i.e.*, , views), a semi-supervised learning task and an unsupervised domain adaptation task, in which each supervision can be better leveraged. To encourage knowledge sharing and integration between the two tasks, we employ co-training, a well-established technique that allows for distinct views to learn from each other. We provided empirical evidence that the two tasks satisfy the theoretical condition of co-training, which makes DECOTA well founded, simple (without adversarial learning), and superior in performance.

Chapter 6: Online Adaptation for Cross-domain Streaming Data

In the context of online privacy, many methods propose complex privacy and security preserving measures to protect sensitive data. In this chapter we argue that: not storing any sensitive data is the best form of security. Thus we propose an online framework that "burns after reading", i.e. each online sample is immediately deleted after it is processed. Meanwhile, we tackle the inevitable distribution shift between the labeled public data and unlabeled private data as a problem of unsupervised domain adaptation. Specifically, we propose a novel algorithm that aims at the most fundamental challenge of the online adaptation setting–the lack of diverse source-target data pairs. Therefore, we design a **Cro**ss-**Do**main **Bo**otstrapping approach, called **CRODOBO**, to increase the combined diversity across domains. Further, to fully exploit the valuable discrepancies among the diverse combinations, we employ the training strategy of multiple learners with co-supervision. CRODOBO achieves state-of-the-art online performance on four domain adaptation benchmarks.

6.1 Introduction

With the onslaught of the pandemic, the internet has become an even more ubiquitous presence in all of our lives. Living in an enormous web connecting us to each other, we now face a new reality: it is very hard to escape one's past on the Internet since every photo, status update,

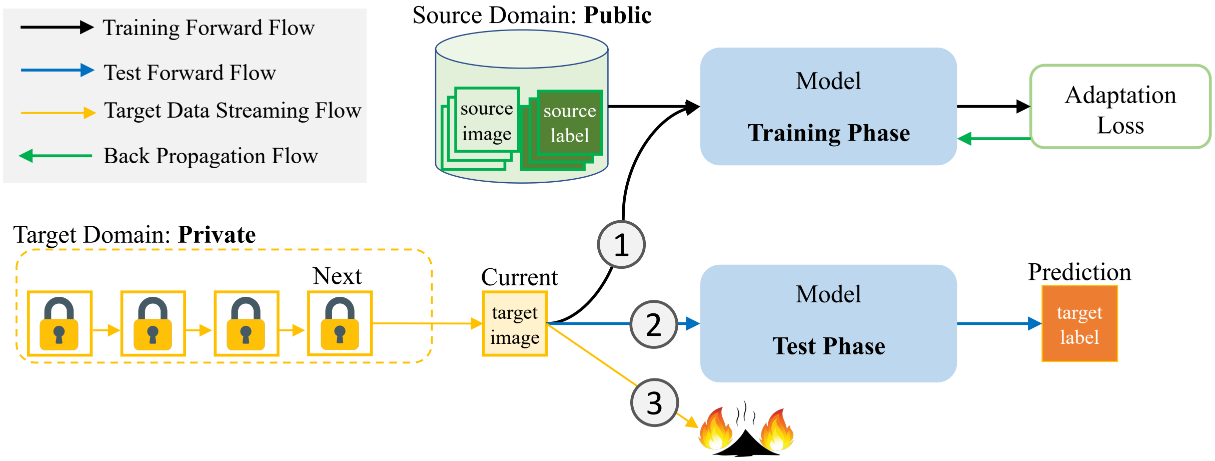

Figure 6.1: The data-flow of the proposed **Burn After Reading** framework at one iteration. The iteration contains a training and a test phase. In training phase, the model takes labeled data from the public source domain, and the current unlabeled target data from the private target domain. The model updates based on the adaptation loss and then moves to test phase. After prediction, the current target data is permanently **deleted** from the target domain. Each target data is (1) trained (2) tested (3) deleted. *Best viewed in color.*

and tweet lives forever in the cloud [19, 148]. Moreover, recommender systems that actively explore the user data [149, 150] for data-driven algorithms have brought controversy that the right to privacy is more important than the convenience. Fortunately, we have *the Right to Be Forgotten* (RTBF), which gives individuals the right to ask organizations to delete their personal data. Recently, many solutions [151, 152] have been proposed that try to preserve privacy in the context of deep learning, mostly focused on the Federated Learning [153, 154]. Federated Learning allows asynchronous update of multiple nodes, in which sensitive data is stored only on a few specific nodes. However, recent studies [155, 156, 157] show that private training data can be leaked through the gradients sharing mechanism deployed in distributed models. In this chapter, we argue that: *not storing any sensitive data is the best form of security*.

The best form of security requires us to delete the user data after use, which necessitates an online framework. However, existing online learning frameworks [158, 159] cannot meet this need without addressing the distribution shift from public data, *i.e.* source domain, to the private user data, *i.e.* target domain. Therefore, in this chapter we propose an online domain adaptation framework in which the target domain streaming data is deleted immediately after adapted. We name the framework “Burn After Reading”, as illustrated in Figure 6.1. The task that is seemingly an extended setting of unsupervised domain adaptation (UDA), however, cannot simply be solved by the online implementation of the offline UDA methods. We explain the reason with a comprehensive analysis of the existing domain adaptation methods. To begin with, existing offline UDA methods rely heavily on the rich combinations of cross-domain mini-batches that gradually adjust the model for adaptation [8, 15, 16, 34, 35, 36, 112, 160], which the online streaming setting cannot afford to provide. In particular, many domain adversarial-based methods [9, 11, 12, 161] depend on a slowly annealing adversarial mechanism that requires discriminating large number of source-target pairs to achieve the adaptation. Recently, state-of-the-art offline methods [37, 162, 163] show promising results by exploiting target-oriented clustering, which requires an offline access to the entire target domain. Therefore, the online UDA task needs new solutions to succeed at scarcity of the data from target domain.

We aim straight at the most fundamental challenge of the online task—the lack of diverse cross-domain data pairs—and propose a novel algorithm based on cross-domain bootstrapping for online domain adaptation. At each online query, we increase the data diversity across domains by bootstrapping the source domain to form diverse combinations with the current target query. To fully exploit the valuable discrepancies among the diverse combinations, we train a set of independent learners to preserve the differences. Inspired by [164], we later integrate the knowledge of learners by exchanging their predicted pseudo-labels on the current target query to co-supervise the learning on the target domain, but without sharing the weights to maintain the learners' divergence. We obtain more accurate prediction on the current target query by an average ensemble of the diverse expertise of all the learners. We call it **CroDoBo**: **Cr**oss-**Do**main **Bo**otstrapping for online domain adaptation, an overview of CroDoBo pipeline is shown in Figure 6.3.

We conduct extensive evaluations on our method, including the classic UDA benchmark *VisDA-C* [165], a practical medical imaging benchmark *COVID-DA* [166] and the large-scale distribution shift benchmark *WILDS* [167] subset *Camelyon*. Moreover, we propose a new adaptation scenario in this chapter from *Fashion-MNIST* [1] to *DeepFashion* [168]. On all the benchmarks, our method outperforms the state-of-the-art UDA methods that are eligible for the online setting. Further, without the reuse of any target sample, our method achieves comparable performance to the offline setting. We summarize the contributions as follows.

- To our best knowledge, we are the first to propose an online domain adaptation framework to implement *the right to be forgotten*.
- We study the fundamental drawback of the online setting compared to offline–the lack of data diversity, and designed a novel online domain adaptation method that improves, and exploits the data diversity.
- Our proposed algorithm achieves new state-of-the-art online results on four challenging benchmarks.
- Although designed for online setting, our method yields comparable performance to the offline setting, suggesting that it is a superior choice even just for time efficiency.

6.2 Related Work

The Right to Be Forgotten [18, 19, 20, 21, 22], also referred to as *right to vanish*, *right to erasure* and *courtesy vanishing*, is the right given to each individual to ask organizations to delete their personal data. RTBF is part of the General Data Protection Regulation (GDPR). As a legal document, the GDPR outlines the specific circumstances under which the right applies in *Article 17* GDPR [1]. The first item is: *The personal data is no longer necessary for the purpose an organization originally collected or processed it.* Yet, the exercise of this right has become a thorny issue in applications. Politou *et al.* [24] discussed that the technical challenges of aligning modern systems and processes with the GDPR provisions are numerous and in most cases insurmountable. In [23] they specifically examined the implications of erasure requests on current backup systems and highlight a number of challenges pertained to the widely known backup standards, data retention policies, backup mediums, search services, and ERP systems [169]. In the context of machine learning, Villaronga *et al.* [18] addressed that the core issue of the AI and Right to Be Forgotten problem is the dearth of interdisciplinary scholarship supporting privacy law and regulation. Graves *et al.* [170] proposed three defense mechanisms against a general threat model to enable deep neural networks to forget sensitive data while maintaining model efficacy. In this paper, we focus on how to obtain model efficacy while erasing data online to protect the user's right to be forgotten.

Online Adaptation to Shifting Domains was first investigated in Signal Processing [171] and later studied in Natural Language Processing [172] and Vision tasks [173, 174, 175, 176, 177, 178, 179, 180]. Jain *et al.* [174] assumed the original classifier output a continuous number

[1] *Article 17* GDPR - Right to be forgotten `https://gdpr.eu/article-17-right-to-be-forgotten/`

of which a threshold gives the class, and reclassify points near the original boundary using a Gaussian process regression scheme. The procedure is presented as a Viola-Jones cascade of classifiers. Moon *et al.* [179] proposed a four-stage method by assuming a transformation matrix between the source subspace and the mean-target subspace embedded in the Grassmann manifold. The method is designed for handcrafted features. In the context of deep neural network, we argue that one transformation matrix might not be sufficient to describe the correlation between source and target deep representations [181]. Taufique *et al.* [182] approached the task by selectively mixing the online target samples with those that were saved in a buffer. Without a further discussion of which samples can be saved in the buffer, we find this method limited in the exercise of the right to be forgotten.

Active Domain Adaptation [183, 184, 185, 186, 187, 188, 189, 190] also benefits the online learning of shifting domains. It bears a different setting: the target domain can actively acquire labeled data online. Rai *et al.* [183] presented an algorithm that harnessed the source domain data to learn a initializer hypothesis, which is later used for active learning on the target domain. Ma *et al.* [185] allowed a small budget of target data for the categories that appeared only in target domain and presented an algorithm that jointly trains two sub-networks of different learning strategies. Chen *et al.* [187] proposed an algorithm that can adaptively deal with interleaving spans of inputs from different domains by a tight trade-off that depends on the duration and dimensionality of the hidden domains.

Test-Time Domain Adaptation [191, 192, 193] is another related task. Similar to the "burn after reading", test-time DA also aims at a fast adaptation to the target samples. Differently, test-time DA is motivated by the unavailability of the source domain [192], which is a variant of source-free domain adaptation [162]. Thus, it is based on a continual setting. Meanwhile, test-time domain adaptation does not require target samples being deleted after training, although Wang *et al.* [192] and Sun *et al.* [193] both discussed the extension to an online setting in the experiments. Without the access to source samples, Varsavsky *et al.* [191] leverages a combination of adversarial learning and consistency under augmentation. Sun *et al.* [193] exploits the self-supervision with auxiliary rotation prediction. In this paper, we compare with test-time DA with a devised continual version of our method in the supplementary.

Ensemble Methods for Online Learning [194, 195, 196] such as bagging and boosting have shown advantages handling *concept drift* [197] and class imbalance, which are common challenges in the online learning task. Barros *et al.* [195] proposed to modify Oza and Russell's Online Boosting [198] based on heuristic modifications. They investigated the effects of weakening the requirements to allow the experts to vote and changing the concept drift detection method, and proposed an improved approach of boosting-like online learning ensemble. MinKu *et al.* [196] addressed the importance of ensemble diversity to improve accuracy in changing environments and proposed the measurement of ensemble diversity. Han *et al.* [199] proposed a regularization for online tracking with a subset of branches in the neural network that are randomly selected. Although online learning and online domain adaptation share similar streaming form of data input, we argue that the two tasks face fundamentally different challenges. For online learning, the challenge is to select the most trustworthy supervisions from the streaming data by differentiating the informative vs. misleading data points, also known as the *stability-plasticity dilemma* [200]. For online domain adaptation, the streaming data of target domain naturally comes unlabeled, and the challenge is the scarcity of supervision. Thus the goal is how to maximize the utilization of the supervision from a different but related labeled source domain.

6.3 Approach

In this section, we introduce the proposed method for "Burn After Reading" framework, in which the samples from the public source domain are fully accessible, while only one/a batch of the target samples is available at each iteration. The model "reads" the current target data, updates, then predicts, after which the target data is deleted permanently from the target domain. In Sec 6.3.1 we describe the difference between online and the offline setting. In Sec 6.3.2, we first introduce the cross-domain bootstrapping strategy and the theoretical insights behind. Then we describe the details of the co-supervision. The input includes the labeled data from a public source domain with labels, and a private target domain without labels. During training, the samples from the public source domain are fully accessible, while only one/a batch of the target samples is available at each iteration. The model "reads" the current target data, updates, then predicts, after which the target data is deleted permanently from the target domain. The process is repeated along the target streaming data until it is finished.

6.3.1 Offline vs. Online

Given the labeled source data $D_{\mathcal{S}} = \{(s_i, y_i)\}_{i=1}^{N_S}$ drawn from the source distribution $p_s(x, y)$, and the unlabeled target data $D_{\mathcal{T}} = \{t_i\}_{i=1}^{N_T}$ drawn from the target distribution $p_t(x, y)$, where N_S and N_T represent the number of source and target samples, both offline and online adaptation aim at learning a classifier that make accurate predictions on $D_{\mathcal{T}}$. The *offline adaptation* assumes access to every data point in both $D_{\mathcal{S}}$ and $D_{\mathcal{T}}$, synchronous [10, 15, 34, 35] or domain-wise asynchronous [162]. The inference on $D_{\mathcal{T}}$ happens after the model is trained on both $D_{\mathcal{S}}$ and $D_{\mathcal{T}}$ entirely. Differently for *online adaptation*, we assume the access to the entire $D_{\mathcal{S}}$, while the data from $D_{\mathcal{T}}$ arrives in a streaming data of random mini-batches $\{T_j = \{t_b\}_{b=1}^{B}\}_{j=1}^{M_{\mathcal{T}}}$. B is the batch size and $M_{\mathcal{T}}$ is the total number of target batches. Each mini-batch T is first adapted, tested and then erased from $D_{\mathcal{T}}$ without replacement, as shown in Figure 6.1 and 6.3. We refer each online batch of target data as a target *query*.

The fundamental challenge of our online task is the limited access to the training data at each inference query, compared to the offline task. For generality, we can assume there are 10^3 source and target batches, respectively. In an offline setting, the model is tested after training on at most 10^6 combinations of source-target data pairs, while in an online setting, an one-stream model can see at most $10^3 + 500$ combinations at the 500-th query. Undoubtedly, the online adaptation faces a significantly compromised data diversity. The training process of our task suffers from two major drawbacks: (I) The model is prone to underfitting on target domain due to the lack of seen target samples, especially at the early stage of training. (II) Due to the deletion of previous data, the model lacks the diverse combinations of source-target data pairs that enable the deep network to find the optimal cross-domain classifier [201].

The goal of the proposed method is to minimize the two drawbacks of the online setting. We first propose to increase the data diversity by cross-domain bootstrapping, and we preserve the discrepancy in independently trained learners. Then we fully exploit the valuable discrepancies of these learners by exchanging their expertise on the current target query to co-supervise each other.

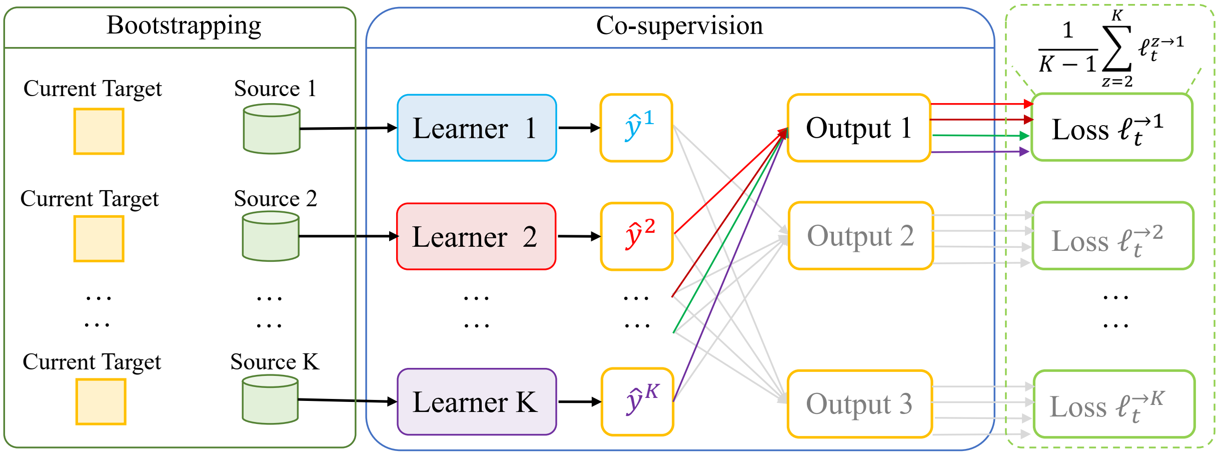

Figure 6.2: Illustration of computing co-supervision loss ($\ell_t^{z\to k}$ in Eq. 6.4), taking $\ell_t^{\to 1}$ for example. The co-supervision for learner 1 is from the *other* K-1 learners. The current target data is repeatedly paired with each bootstrapped source data to improve data diversity. Each learner takes a unique data combination and generates pseudo-label $\hat{y}^k$ of the current target data. Then $\ell_t^{\to 1}$ receives co-supervision averagely from the pseudo-labels $\{\hat{y}^2, \hat{y}^3, ..., \hat{y}^K\}$.

6.3.2 Proposed Method

6.3.2.1 Cross-domain Bootstrapping for Data Diversity

The diversity of cross-domain data pairs is crucial for most prior offline methods [8, 10, 35] to succeed. Since the target samples cannot be reused in the online setting, we propose to increase the data diversity across domains by bootstrapping the source domain to form diverse combinations with the current target domain query, as shown in Figure 6.2. Specifically, for each target query T_j, we randomly select a set of K mini-batches $\{S_j^k = \{(s_b)_{b=1}^B\}\}_{k=1}^K$ of the same size from the source domain with replacement. Correspondingly, we define a set of K base learners $\{\boldsymbol{w}^k\}_{k=1}^K$. At each iteration, a learner $\boldsymbol{w}^k$ makes prediction for query T_j after trained on

$\{T_j, S_j^k\}$, and updates via

$$\begin{aligned} \boldsymbol{w}^k &\leftarrow \boldsymbol{w}^k - \eta \left(\nabla \mathcal{L}(\boldsymbol{w}^k, \{T_j, S_j^k\}) \right), \\ p_j^k &= p\left(c | T_j; \boldsymbol{w}^k\right), \end{aligned} \tag{6.1}$$

where η is the learning rate, c is the number of classes, p_j^k is the predicted probability by the k-th learner, and $\mathcal{L}(,)$ is the objective function. The predicted class for T_j is the average of K predictions of the base learners. We justify our design choice from the perspective of uncertainty estimation in the following discussion.

Theoretical Insights As mentioned in Section 6.3.1, we aim at the best estimation of the current target query. We first consider a single learner situation. At the j-th query, the learner faces a fundamental trade-off: by minimizing the uncertainty of the j-th query, the learner can attain the best current estimation. Yet the risk of fully exploring the uncertainty is to spoil the existing knowledge from the previous j-1 target domain queries. However, if we don't treat the uncertainty, the single observation on j-th query is less informative for current query estimation. Confronting the dilemma, we should not ignore that the uncertainty captures the variability of a learner's posterior belief which *can* be resolved through statistical analysis of the appropriate data [202]. This gives us hope for a more accurate model via uncertainty estimation. One popular suggestion for resolving uncertainty is to use *Dropout* [113, 203, 204] sampling, where individual neurons are independently set to zero with a probability. As a sampling method on the neurons, *Dropout* works in a similar form of *bagging* [205, 206] of multiple decision trees. It might equally reduce the overall noise of the network regardless of domain shift but it does not address the problem of our task, which is the lack of diverse cross-domain combinations.

Alternatively, we employ another pragmatic approach *Bootstrap* for uncertainty estimation on the target domain that offsets the source dominance. With the scarcity of target samples, we propose to bootstrap source-target data pairs for a more balanced cross-domain simulation. At high-level, the bootstrap simulates multiple realizations of a specific target query given the diversity of source samples. Specifically, the bootstrapped source approximate a distribution over the current query T_j via the bootstrap.

The bootstrapping brings multi-view observations on a single target query by two means. First, given K sampling subsets from $D_{\mathcal{S}}$, let $\mathcal{F}$ be the ideal estimate of T_j, $\hat{\mathcal{F}}$ be the practical estimate of the dataset, and $\hat{\mathcal{F}}^*$ be the estimate from a bootstrapped source paired with the target query, $\hat{\mathcal{F}}^* = K^{-1}\sum_{k=1}^{K}\hat{\mathcal{F}}_k^*$ will be the average of the multi-view K estimates. Second, besides the learnable parameters, the *Batch-Normalization* layers of K learners generate result in a set of different means and variances $\{\mu_k, \sigma_k\}_{k=1}^{K}$ that serve as K different initializations that affects the learning of $\hat{\mathcal{F}}^*$.

6.3.2.2 Exploit the Discrepancies via Co-supervision

After the independent learners have preserved the valuable discrepancies of cross-domain pairs, the question now is how to fully exploit the discrepancies to improve the online predictions on the target queries. On one hand, we want to integrate the learners' expertise into one better prediction on the current target query, on the other we hope to maintain their differences. Inspired by [164], we train the K learners jointly by exchanging their knowledge on the target domain as a form of co-supervision. Specifically, the K learners are trained independently with bootstrapped source supervision, but they exchange the pseudo-labels generated for target queries. We followed

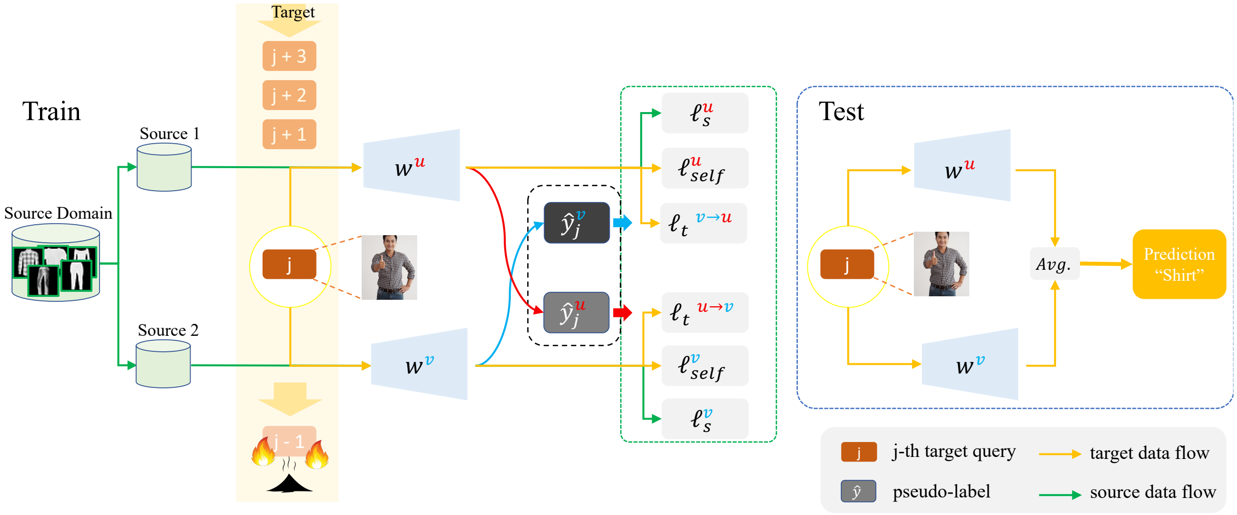

Figure 6.3: The full pipeline of the proposed CroDoBo K=2 method at j-th iteration. Only one target query j is currently available from target domain in this iteration. We bootstrap the source domain and combine with the current j-th query. The learners w^u (k=1) and w^v (k=2) exchange the generated pseudo-labels $\hat{y}_j^u$ and $\hat{y}_j^v$ as co-supervision. Each learner is updated by a supervised loss ℓ_s on source data, a self-supervised loss ℓ_{self} on the target data and a co-supervised loss ℓ_t. The test result is recorded by averaging the predictions of both learners. Once tested, query j is immediately deleted.

the *FixMatch* [145] to compute pseudo-labels on the target domain. We first consider K=2 for simplicity, we denote the learners as $\boldsymbol{w}^u$ for $k = 1$ and $\boldsymbol{w}^v$ for $k = 2$, respectively.

Given the current target query T_j, the loss function $\mathcal{L}$ consists a supervised loss term ℓ_s from the source domain with the bootstrapped samples, and a self-supervised loss term ℓ_t from the target domain with pseudo-labels $\hat{y}_b$ from the peer learner, as illustrated in Figure 6.3. We denote the cross-entropy between two probability distributions as $\mathcal{H}(;)$. Thus, the co-supervision objective ℓ_t is obtained via:

$$\begin{aligned}\ell_t^{v\to u} &= B^{-1}\sum_{b=1}^{B} \nVdash \left(p_b^v \geq \tau\right) \mathcal{H}\left(\hat{y}_b^v; p(c|\tilde{t}_b; \boldsymbol{w}^u)\right),\\ \ell_t^{u\to v} &= B^{-1}\sum_{b=1}^{B} \nVdash \left(p_b^u \geq \tau\right) \mathcal{H}\left(\hat{y}_b^u; p(c|\tilde{t}_b; \boldsymbol{w}^v)\right),\end{aligned} \tag{6.2}$$

Algorithm 3: The CroDoBoalgorithm

```
Input  : Number of learners $K$, learners $\{\boldsymbol{w}^k\}_{k=1}^K$, learning rate $\eta$, number of target
         queries $N_T$, confidence threshold $\tau$, batch size $B$, transform $F$, data $D_{\mathcal{S}}$, $D_{\mathcal{T}}$,
         number of class $c$;
for $j \leftarrow 1$ to $N_T$ do
    Given $T_j = \{t_b\}_{b=1}^B$ from $D_{\mathcal{T}}$, $\{\tilde{t_b}\} = \{F(t_b)\}$,
    Sample $S_j^k$ from $D_{\mathcal{S}}$, repeat $K$ times;
    for $k \leftarrow 1$ to $K$ do
        Update $\boldsymbol{w}^k \leftarrow \boldsymbol{w}^k - \eta \nabla \ell_s^k$,
        Obtain pseudo-labels $\{\hat{y}_b^k\}_{b=1}^B = \{\arg\max_c (p(c|t_b; w^k) > \tau)\}_{b=1}^B$;
    end
    for $k \leftarrow 1$ to $K$ do
        for $b \leftarrow 1$ to $B$ do
            Obtain $\{\ell_t^{z \to k}\}_{z=1}^{K-1} = \{\mathbf{1}\left(p_{\tilde{b}}^z \geq \tau\right) \mathcal{H}\left(\hat{y}_b^k; p_{\tilde{b}}^k\right)\}_{z=1}^{K-1}$,
            Obtain $\ell_{\text{self}} = \ell_{\text{ent}} + \lambda \ell_{\text{div}}$;
        end
    end
    Update $\boldsymbol{w}^k \leftarrow \boldsymbol{w}^k - \eta(\frac{1}{K-1} \sum_{z=1}^{K-1} \nabla \ell_t^{z \to k} + \nabla \ell_{\text{self}})$
    Output $\hat{y}_{\text{test}} = \arg\max_c \frac{1}{K-1} \sum_{k=1}^K p(c|T_j; \boldsymbol{w}^k)$.
end
```

p_b^u and p_b^v are the predicted probabilities of t_b by $\boldsymbol{w}^u$ and $\boldsymbol{w}^v$, respectively. τ is the threshold for pseudo-label selection, and $\tilde{t}_b$ is a strongly-augmented version of t_b using *Randaugment* [207]. However, we note that *RandAug* is a technique only employed to increase data diversity, but is **not** required for CroDoBo. We denote the version without any augmentation as CroDoBo, and we denote the version with *RandAug* as CroDoBo+.

To further exploit the supervision from the limited target query, from p_b^u and p_b^v we compute a self-supervised loss $\ell_{\text{self}} = \ell_{\text{ent}} + \lambda \ell_{\text{div}}$, in which ℓ_{ent} is standard entropy and ℓ_{div} is a balancing term for class-diversity, λ is a weighting factor. The ℓ_{self} is widely used in prior domain adaptation works [16, 32, 162]. Finally, we update the learners by

$$\begin{aligned} \boldsymbol{w}^u &\leftarrow \boldsymbol{w}^u - \eta(\ \nabla \ell_s(\boldsymbol{w}^u, S_j^u) + \nabla \ell_t^{v\to u} + \nabla \ell_{\text{self}}(\boldsymbol{w}^u, T_j)), \\ \boldsymbol{w}^v &\leftarrow \boldsymbol{w}^v - \eta(\ \nabla \ell_s(\boldsymbol{w}^v, S_j^v) + \nabla \ell_t^{u\to v} + \nabla \ell_{\text{self}}(\boldsymbol{w}^v, T_j)). \end{aligned} \tag{6.3}$$

An algorithm table is illustrated in Algorithm 3.

For $K > 2$, each learner $\boldsymbol{w}^k$ is updated with the co-supervision from the other $K - 1$ learners (Figure 6.2), weighted by $1/(K-1)$ for each $\ell_t^{z\to k}$ (z is the learner's index other than k). We update $\boldsymbol{w}^k$ by

$$\boldsymbol{w}^k \leftarrow \boldsymbol{w}^k - \eta(\nabla \ell_s(\boldsymbol{w}^k, S_j^u) + \frac{1}{K-1}\sum_{z=1}^{K-1} \nabla \ell_t^{z\to k} + \nabla \ell_{\text{self}}(\boldsymbol{w}^k, T_j)). \tag{6.4}$$

6.4 Experiments

We consider two metrics for evaluating online domain adaptation methods: *online average accuracy* and *one-pass accuracy*. The online average is an overall estimate of the streaming effectiveness. The one-pass accuracy measures after training on the finite-sample how much the online model has deviated from the beginning [208]. A *one-pass accuracy* much lower than *online average* indicates that the model might have overfitted to the fresh queries, but compromised its generalization ability to the early queries.

Dataset. We use **VisDA-C** [165], a classic benchmark adapting from synthetic images to real. We followed the data split used in prior offline settings [35, 162, 165]. We also use **COVID-DA** [166], adapting the CT images diagnosis from common pneumonia to the novel disease.

This is a typical scenario where online domain adaptation is valuable in practice. When a novel disease breaks out, without any prior knowledge, one has to exploit a different but correlated domain to assist the diagnosis of the new pandemic in a time-sensitive manner. We also evaluate on a large-scale medical dataset *Camelyon17* from the **WILDS** [167], a histopathology image datasets with patient population shift from source to the target. Camelyon17 has 455k samples of breast cancer patients from 5 hospitals. Another practical scenario is the online fashion where the user-generated content (UGC) might be time-sensitive and cannot be saved for training purposes. Due to the lack of cross-domain fashion prediction dataset, we propose to evaluate adapting from **Fashion-MNIST** [1]-to-**DeepFashion** [168] category prediction branch. We select 6 fashion categories shared between the two datasets, and design the task as adapting from $36,000$ grayscale samples of Fashion-MNIST to $200,486$ real-world commercial samples from DeepFashion.

Implementation details. We implement using Pytorch [139]. We follow [162, 163] to use ResNet-101 [137] on VisDA-C pretrained on ImageNet [141, 142]. We follow [166] to use pretrained ResNet-18 [137] on COVID-DA. We follow the leader-board on WILDS challenge [167] [2] to use DenseNet-121 [209] on Camelyon17 with random initialization, we use the official WILDS codebase (v1.1.0) for data split and evaluation. We use pretrained ResNet-101 [137] on Fashion-MNIST-to-DeepFashion. Our target query batch-size and bootstrapped source batch-size are both set as 64. The confidence threshold $\tau = 0.95$ and diversity weight $\lambda = 0.4$ are fixed throughout the experiments. Our method is not sensitive to hyperparameters, the results are reported in supplementary.

Baselines. We compare **CroDoBo** without data augmentation and **CroDoBo**$^+$ with *RandAug*

[2] https://wilds.stanford.edu/leaderboard/

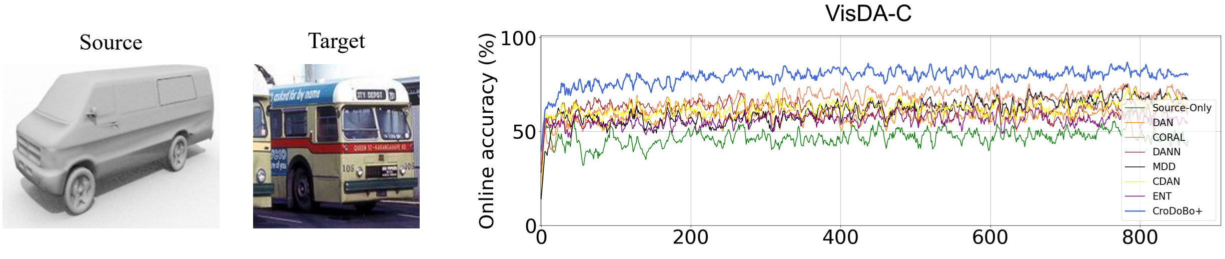

Figure 6.4: Results of online adaptation from synthetic source domain to real target domain on VisDA-C [165] with "Burn After Reading". The x-axis is the online streaming timestep. Each query contains 64 samples. Each approach takes the same randomly perturbed sequence of target queries. Source-Only is in green, the proposed CroDoBo is in blue. Smoothed with 1-D uniform filter with length=5. *Best viewed in color.*

Table 6.1: Accuracy on VisDA-C (%) using ResNet-101. In the online setting, individual class reports accuracy after one-pass, *one-pass* is the class average. Best offline (***italic bold***), best online (**bold**).

Methods (Syn → Real)		plane	bike	bus	car	horse	knife	motor	person	plant	skate	train	truck	Online	One-pass	Per-Class Acc.
Offline	Source-Only	67.7	27.4	50.0	61.7	69.5	13.7	85.9	11.5	64.4	34.4	84.2	19.2	-	-	49.1
	DAN [107]	84.4	50.9	68.4	66.8	82.0	17.0	82.3	22.0	73.3	47.4	81.2	18.3	-	-	57.8
	CORAL [210]	94.7	46.8	78.0	62.4	86.5	70.1	90.4	73.5	84.2	34.9	87.7	24.9	-	-	69.5
	DANN [10]	81.9	77.7	***82.8***	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	-	-	57.4
	ENT [32]	88.6	29.5	82.5	***75.8***	88.7	16.0	***93.2***	63.4	***94.2***	40.1	87.3	12.1	-	-	64.3
	MDD [160]	89.2	58.9	70.5	54.5	71.1	42.9	78.8	22.5	68.6	54.7	88.6	15.4	-	-	59.6
	CDAN [13]	89.4	40.3	74.6	65.2	81.5	62.2	90.1	69.3	73.3	58.6	84.8	19.1	-	-	67.4
	SHOT [162]	94.3	***88.5***	80.1	57.3	93.1	94.9	80.7	***80.3***	91.5	89.1	86.3	58.2	-	-	82.9
	ATDOC-NA [163]	***95.3***	84.7	82.4	75.6	***95.8***	***97.7***	88.7	76.6	94.0	***91.7***	***91.5***	***61.9***	-	-	***86.3***
Online	Source-Only	73.3	6.5	44.9	67.8	58.6	5.7	67.2	18.3	47.7	19.2	84.1	9.3	46.7	41.9	-
	DAN [107]	87.7	45.9	69.9	70.9	77.4	17.7	80.7	18.6	79.9	29.9	82.7	16.6	57.8	56.5	-
	CORAL [210]	94.7	51.0	79.6	63.2	88.2	69.4	**91.1**	73.1	87.7	41.8	88.4	24.2	66.7	71.0	-
	DANN [10]	84.5	39.2	70.2	60.4	77.1	28.6	90.9	20.5	67.7	39.9	**89.8**	10.5	49.0	56.6	-
	ENT [32]	87.1	14.8	87.9	71.9	87.8	**98.9**	90.3	0.0	5.2	15.0	80.4	0.2	55.8	53.3	-
	MDD [160]	**95.1**	52.2	87.9	57.9	90.3	94.8	88.4	45.7	76.2	50.5	77.7	25.7	60.4	70.1	-
	CDAN [13]	88.5	44.3	74.3	68.4	80.3	60.2	89.9	69.9	74.3	57.1	84.8	13.9	62.3	67.1	-
	CroDoBo	94.8	**87.5**	**90.5**	**76.0**	**94.9**	93.7	88.7	**80.1**	**94.8**	**89.4**	84.6	**30.7**	**79.4**	**84.0**	-

with eight state-of-the-art domain adaptation approaches, including **DAN** [107], **CORAL** [210], **DANN** [10], **ENT** [32, 96], **MDD** [160], **CDAN** [13], **SHOT** [162] and **ATDOC** [163]. ATDOC has multiple variants of the auxiliary regularizer, we compared with the *Neighborhood Aggregation* (ATDOC-NA) with the best performance in [163]. Among the compared approaches, SHOT and ATDOC-NA require a memory module that collects and stores information of all the target samples, thus only apply the offline setting. For the other six approaches, we compare both

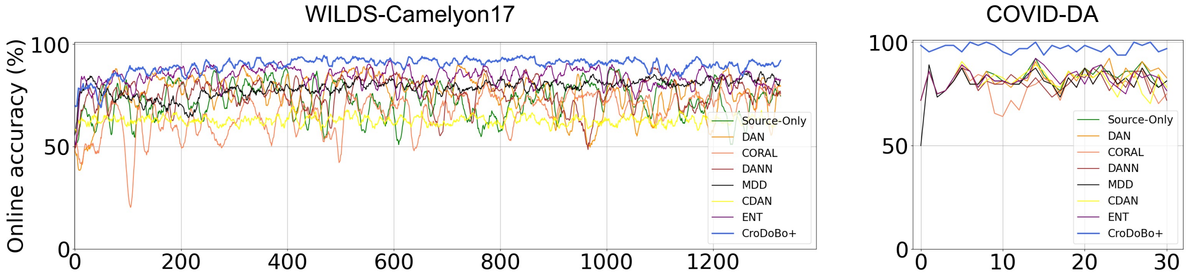

Figure 6.5: Results of online accuracy on *WILDS*-Camelyon17 [167] with hospital patient population shift, and COVID-DA [166] adapting from common pneumonia to COVID-19 medical images with "Burn After Reading". Source-Only is in green, the proposed CroDoBo is the solid blue line. Smoothed with 1-D uniform filter with length=5 for *WILDS*-Camelyon17.

offline and online results. Each offline model is trained for 10 epochs. Each online model is trained batch-by-batch for 1 epoch, during which the online test results are recorded after each model update. All the online baselines take the same randomly-perturbed target queries to make a fair comparison. The results of CroDoBo and CroDoBo+ reported in Table 6.1-6.4 have 2 learners (*i.e.* K=2), the results with $K \geq 3$ are reported in Table 6.7.

Main results. We summarize the results on VisDA-C [165] in Table 6.1, and plot the online results in Figure 6.4 We follow [37, 162, 163, 165] to provide the VisDA-C one-pass accuracy in class average. In Table 6.1: Online, the proposed CroDoBo largely outperforms other baselines. Without augmentation, our method outperforms the second by 11.5%. Our online result is on par with the state-of-the-art offline performance ATDOC-NA [162], outperforming many other offline baselines.

Comparing across the offline and online setting, the Source-Only baseline drops 2.4% in the online average and 7.2% in the one-pass accuracy, which indicates that the data diversity is also important in domain generalization. We observe that ENT [32], which is an entropy regularizer on the posterior probabilities of the unlabeled target samples, has a noticeable performance drop in the online setting, and illustrates more obvious imbalanced results over the categories (superior at class “knife” but poor at “person” and “truck”). We consider it a typical example of bad objective choice for the online setting when the dataset is imbalanced. Without sufficient rounds to provide data diversity, entropy minimization might easily overfit the current target query. The 2.5% drop

Table 6.2: Offline and online accuracy (%) on COVID-DA [166], adaptation from pneumonia to Covid. All the baselines use ResNet-18 as the backbone. COVID-DA* is the method proposed in [166] along with dataset.

Methods (Pneumonia → Covid)		Online	One-pass	Offline
Offline & Online	Source-Only	83.6	82.0	88.9
	DAN [107]	84.4	85.7	87.7
	CORAL [210]	67.6	45.4	65.4
	DANN [10]	83.0	87.1	87.7
	ENT [32]	84.3	87.3	89.8
	MDD [160]	83.2	86.2	81.0
	CDAN [13]	83.0	86.4	86.3
	SHOT [162]	-	-	93.2
	ATDOC-NA [163]	-	-	***98.1***
	COVID-DA* [166]	-	-	***98.1***
	CroDoBo (ours)	95.0	**97.1**	-
	CroDoBo$^+$(ours)	**96.5**	**97.1**	-

Table 6.3: Accuracy on *WILDS*-Camelyon17 [167] (%) using DenseNet-121. *Domain Generalization* results are reprinted from *WILDS* leaderboard (see Footnote 2).

Methods (Hospital 1,2,3 → Hospital 5)		Online	One-pass	Offline
Domain Generalization	ERM [167]	-	-	70.3
	Group DRO [167]	-	-	68.4
	IRM [167]	-	-	64.2
	FISH [211]	-	-	74.7
Offline & Online	Source-Only	71.7	60.1	63.6
	DAN [107]	76.3	78.0	69.0
	CORAL [210]	66.0	87.1	85.0
	DANN [10]	76.4	81.4	86.7
	ENT [32]	83.1	82.3	***87.5***
	MDD [160]	77.8	52.5	63.7
	CDAN [13]	62.7	60.1	58.5
	SHOT [162]	-	-	73.8
	ATDOC-NA [163]	-	-	86.3
	CroDoBo (ours)	87.5	89.2	-
	CroDoBo$^+$(ours)	**89.2**	**91.9**	-

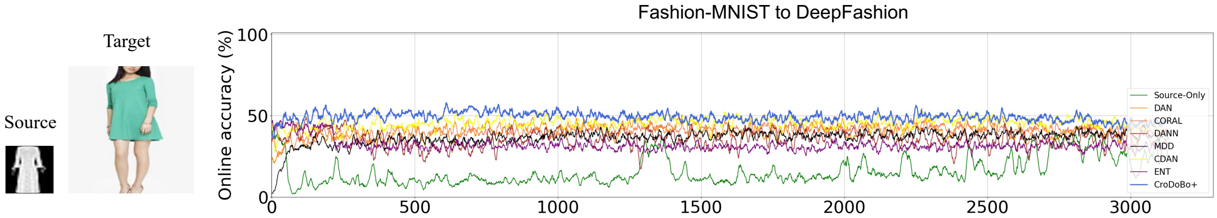

Figure 6.6: Results of online adaptation from Fashion-MNIST [1]to DeepFashion [168] with "Burn After Reading". Smoothed with 1-D uniform filter with length=10.

in one-pass from online further confirmed the model has deviated from the beginning.

Results on two medical imaging datasets COVID-DA [166] and *WILDS*-Camelyon17 [167] are respectively summarized in Table 6.2 and Table 6.3.The online streaming accuracy is presented in Figure 6.5. COVID-DA* is the method proposed along with the dataset in [166], which is a domain adversarial-based multi-classifier approach with focal loss regularization. Our method outperforms the other approaches on COVID-DA regarding the online and one-pass metric, and achieves competitive performance against the best offline accuracy. On the large-scale benchmark *WILDS*-Camelyon17, our CroDoBo is on par the the best offline result, and CroDoBo+ outperforms the offline results by 1.7%, which validates the effectiveness of the approach. The good performance on larger number of target queries indicates that CroDoBo can well exploit the underlying information from the target domain. Similar observations are made on the large-scale Fashion benchmark [1, 168]. Meanwhile, we reprint *Domain Generalization* results from the *WILDS* leaderboard for reference.

Results on large-scale Fashion dataset, from Fashion-MNIST [1] to DeepFashion [168] category prediction branch, is summarized in Table 6.4. We provide the online results in Figure 6.6. To the best of our knowledge, we are the first to report results on this meaningful adaptation scenario. The offline Source-Only merely achieves 23.1% accuracy, only 6.5% gain on the basis of the probability of guessing, which indicates the benchmark is challenging. The sharp drop of

Table 6.4: Accuracy on Fashion-MNIST [1] to DeepFashion [168] (%) using ResNet-101.

Methods (F-MNIST → DeepFashion)		Online	One-pass	Offline
Offline & Online	Source-Only	22.7	15.8	23.1
	DAN [107]	40.7	42.0	32.7
	CORAL [210]	40.4	40.7	39.6
	DANN [10]	35.6	26.5	40.5
	ENT [32]	31.9	31.2	31.1
	MDD [160]	36.5	38.0	39.0
	CDAN [13]	45.4	**47.6**	47.2
	SHOT [162]	-	-	42.3
	ATDOC-NA [163]	-	-	***47.4***
	CRODOBO	**49.1**	46.3	-

performance from Source-Only online accuracy to one-pass accuracy (-6.8%) indicates the large domain gap, and how easy the model is dominated by the source domain supervision. Similar observation is made on *WILDS*-Camelyon17 Source-Only results(-11.6% from online to one-pass), this usually happens when the source domain is less challenging than the target domain, and the distribution of the two domains are far from each other. Faced with this challenging benchmark, CRODOBO improves the online performance to a remarkable 49.1%, outperforming the best result in the offline setting. Our one-pass accuracy is slightly shy compared to CDAN [13], but is better in online metric.

Prior Online UDA approaches. In this chapter, we propose a novel cross-domain framework to

Table 6.5: Ablation study of cross-domain bootstrapping on four datasets (%). VisDA-C one-pass accuracy is in per-class. Number of learners $K = 2$ in both w/ CRODOBO and w/o CRODOBO

Method/Dataset		VisDA-C	COVID-DA	Camelyon17	Fashion
Online	w/o CroDoBo	78.5	94.4	86.2	42.3
	w/ CroDoBo	79.4	96.5	89.2	49.1
One-pass	w/o CroDoBo	84.0	97.1	89.4	39.9
	w/ CroDoBo	84.0	97.1	91.9	46.3

implement the right to be forgotten. However, we *do not* claim to have proposed the task of online unsupervised domain adaptation, which has existed before the emergence of deep learning [172, 174, 179]. The recent works are mostly engineered for a specific downstream task [175, 176, 180, 212, 213] that lacks generality. Yet, we try to compare to a more general but unpublished approach CONDA [182] despite its limited availability. The setting of CONDA is different from our approach. It allows a memory module that selectively buffers target queries in which the model can re-access previous target samples. As a result, CONDA is less challenging compared to "burn after reading". Meanwhile, CONDA has a continual setting, in which the model is pretrained on the source domain and then adapted to the target domain. Without any available

Table 6.6: Ablation study on the objectives on target domain on VisDA-C (%). T is the sharpening temperature in the MixMatch [101].

Method	Online	One-pass
default (w/o CroDoBo, τ=0.95, λ=0.4)	78.5 (-)	84.0 (-)
w/o ℓ_{ent}	63.7($\downarrow$)	53.1($\downarrow$)
w/o ℓ_{div}	72.6($\downarrow$)	73.0($\downarrow$)
replace $\ell_{\text{ent}} + \ell_{\text{div}}$ w/ Pseudo-labeling [114] (τ=0.95)	70.2($\downarrow$)	70.0($\downarrow$)
replace $\ell_{\text{ent}} + \ell_{\text{div}}$ w/ MixMatch [101] (T=0.5)	73.0($\downarrow$)	75.3($\downarrow$)
replace ℓ_t w/ MixMatch [101] (T=0.5)	76.3($\downarrow$)	81.5($\downarrow$)
use *Randaug* [207] on $\ell_{\text{ent}}, \ell_{\text{div}}$	77.6($\downarrow$)	83.7($\downarrow$)

Table 6.7: Accuracy on VisDA-C (%) using ResNet-101 with different number of learners K, and comparing the computation speed reported using 2 NVIDIA-P6000 GPUs.

CroDoBo$^+$	plane	bike	bus	car	horse	knife	motor	person	plant	skate	train	truck	Online	One-pass	samples/sec
$K = 2$	94.8	87.5	90.5	76.0	94.9	93.7	88.7	80.1	94.8	89.4	84.6	30.7	79.4	84.0	25
$K = 3$	95.0	85.6	84.2	73.3	94.4	95.7	88.5	82.2	94.4	83.4	89.3	36.6	79.2	83.5	16
$K = 4$	95.5	85.0	85.0	76.1	95.3	96.0	92.7	81.8	92.7	88.9	86.8	37.3	81.3	84.4	12
$K = 5$	96.3	82.3	86.7	83.0	93.7	95.6	91.6	83.2	96.3	87.0	85.2	43.0	82.0	85.3	10

Table 6.8: Accuracy on VisDA-C (%) with Multi-Source CroDoBo$^+$.

Methods (ResNet-101)	plane	bike	bus	car	horse	knife	motor	person	plant	skate	train	truck	Online	Class Avg.
Multi-Source CroDoBo$^+$	96.2	85.4	90.8	79.7	96.6	94.6	93.3	87.5	96.3	92.4	90.2	50.6	84.0	87.8
CroDoBo$^+$ K=2	94.8	87.5	90.5	76.0	94.9	93.7	88.7	80.1	94.8	89.4	84.6	30.7	79.4	84.0

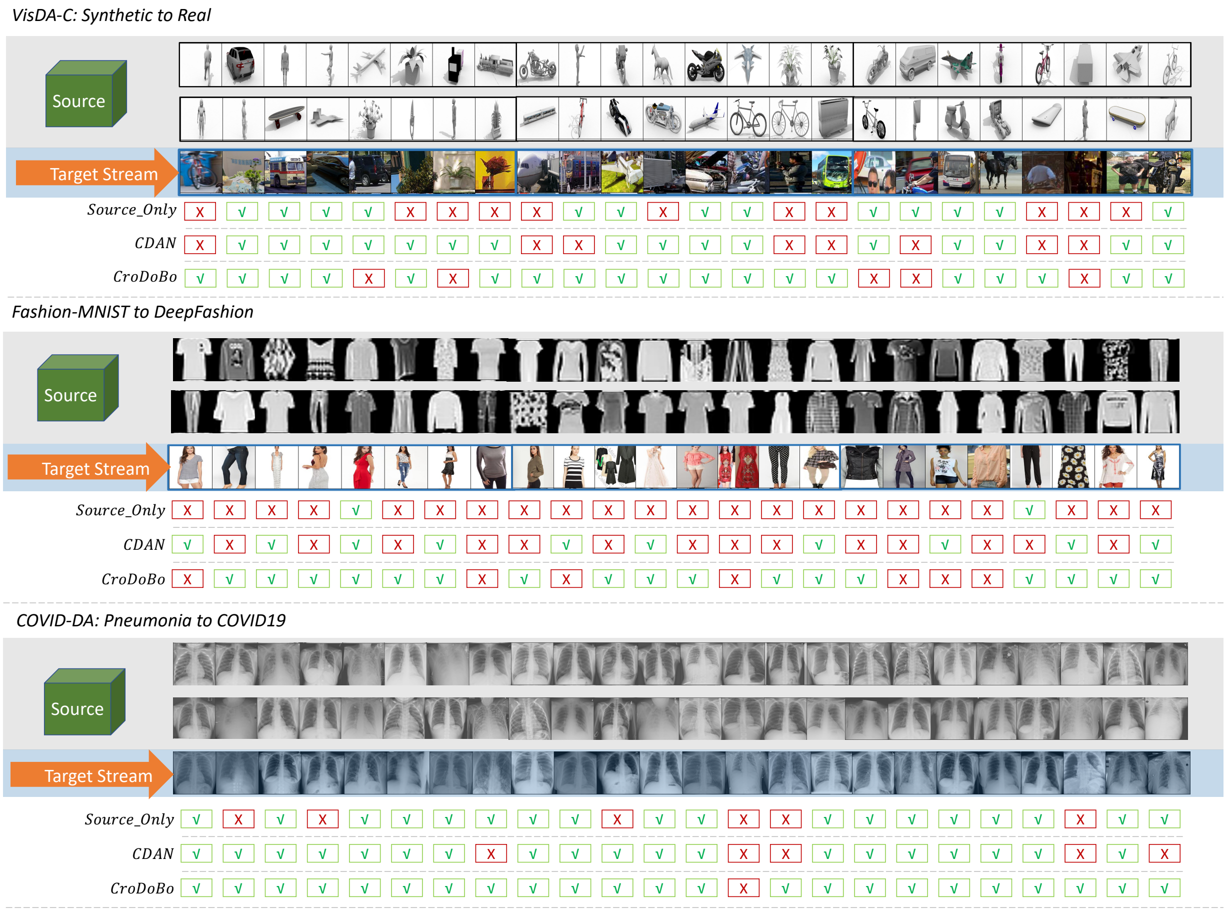
VisDA-C: Synthetic to Real
Source
Target Stream
Source_Only X √ √ √ √ X X X X √ √ X √ √ X X √ √ √ √ X X X √
CDAN X √ √ √ √ √ √ √ X X √ √ √ √ X X √ X √ √ X X √ √
CroDoBo √ √ √ √ X √ X √ √ √ √ √ √ √ √ √ X X √ √ √ X √ √
Fashion-MNIST to DeepFashion
Source
Target Stream
Source_Only X X X X √ X X X X X X X X X X X X X X X √ X X X
CDAN √ X √ X √ X √ X X √ X √ X X X √ X X √ X X √ X √
CroDoBo X √ √ √ √ √ √ X √ X √ √ √ X √ √ √ X X X √ √ √ √
COVID-DA: Pneumonia to COVID19
Source
Target Stream
Source_Only √ X √ X √ √ √ √ √ √ X √ √ X X √ √ √ √ √ √ X √ √
CDAN √ √ √ √ √ √ √ X √ √ √ √ √ X X √ √ √ √ √ √ X √ X
CroDoBo √ √ √ √ √ √ √ √ √ √ √ √ √ X √ √ √ √ √ √ √ √ √ √

Figure 6.7: Qualitative results of a randomly selected target query (size 24). We compare CroDoBo with two essential baselines Source-Only and CDAN [13]. We represent the bootstrapped source samples (top two rows under each benchmark), target samples (third row under each benchmark), and the prediction result of each target sample. *Best viewed in color.*

source code from CONDA [182], we employ the same backbone in [182], *HR-Net* [214], to make a fair comparison. We devise CroDoBoto a continual setting to make it comparable. Without simultaneous access to the source domain, cross-domain bootstrapping is not an option. So we employ the objectives on the target domain, we call it **Continual CroDoBo**. The results are in Table 6.9. We observe that, without any buffer mechanism or re-access to the previous queries, the continual CroDoBo still outperforms ConDA [182]. As mentioned in Section 6.2, we compare to another related task–Test-Time Domain Adaptation [192, 193]. We have analyzed the differences of the setting of Test-Time DA in Section 6.2, and here we provide the results of

Tent [192] compared with the Continual CRODOBO in Table 6.10. We observe that our proposed method largely outperforms Tent on VisDA-C.

Streaming Randomness As mentioned in the Section 6.3.2, in the online setting, each model takes the same target sequence for fair comparison. The target sequence is randomly-perturbed using the a fixed randomseed. Here, we discuss whether the model will be influenced by different random sequential orders. We perturb the original target sequence (arranged in the categorical order) using 5 different random seeds, and report the results of each seed on VisDA-C [165] and the large-scale Fashion-MNIST-to-DeepFashion [168] benchmark. We compare the randomness using CDAN [13] and CRODOBO. We choose CDAN [13] since it is a benchmark adversarial approach, essentially different from the proposed approach. The results are in Table 6.11. We observe that on VisDA-C the variance among different sequential orders is rather small ($<$ 0.25). On the more challenging Fashion benchmark, the variance of CRODOBO is larger but manageable ($<$ 2.0). We analyze that CRODOBO relies more on the target-oriented supervision (see Section 6.3.2) than CDAN [13], which makes it more sensitive towards the changes of the target samples. This is a drawback of CRODOBO that we will try to address in the future work. To conclude, the randomness in forming the order of target queries will not be a factor that influences the evaluation of the online model effectiveness.

Other Pseudo-labeling Approaches as Co-supervision The co-supervision in the proposed method can be replaced with any other pseudo-labeling approaches. One can simply replace the term on either/both $\{w^u,\ w^v\}$ to achieve better performance. We replace on either/both learners with another popular semi-supervised approach MixMatch [101] and report the results in Table 6.12. We observe that FixMatch [145] provides better co-supervision and the online performance drops $\sim$8% when replaced with MixMatch.

Table 6.9: Accuracy on VisDA-C (%) using HR-Net.

Methods (Syn → Real)	plane	bike	bus	car	horse	knife	motor	person	plant	skate	train	truck	Online	One-pass
ConDA [182]	97.0	90.4	80.9	50.0	95.2	95.7	80.3	81.9	94.9	94.2	91.1	63.9	N/P	84.6
Continual CroDoBo (Ours)	96.5	85.2	82.3	47.3	98.0	96.1	89.6	79.2	94.9	95.7	90.4	66.5	80.0	85.1
CroDoBo (Ours)	94.8	86.0	90.7	80.3	97.1	99.1	93.1	85.0	88.2	89.6	90.9	47.1	82.9	86.8

Table 6.10: Comparisons to *Tent* on VisDA-C using ResNet-101.

Methods (ResNet-101)	plane	bike	bus	car	horse	knife	motor	person	plant	skate	train	truck	Online	One-pass
Tent [192]	85.2	44.3	79.4	50.0	78.1	52.7	83.0	43.5	65.0	53.1	81.4	30.1	62.1	-
Continual CroDoBo (Ours)	93.3	75.8	83.6	70.6	92.8	21.8	86.5	80.5	86.6	90.0	79.6	43.6	74.0	75.4

Table 6.11: Online accuracy (%) on five different perturbations of target sequence on VisDA-C [165] and Fashion-MNIST [1]-to-DeepFashion [168].

VisDA-C

Methods	rand 0	rand 1	rand 2	rand 3	rand 4	mean	var
CDAN [13]	62.3	61.0	61.9	61.6	61.9	61.7	0.21
CroDoBo	79.4	78.6	79.6	79.2	79.4	79.2	0.15

Fashion-MNIST-to-DeepFashion

Methods	rand 0	rand 1	rand 2	rand 3	rand 4	mean	var
CDAN [13]	45.4	47.4	46.7	46.3	46.2	46.4	0.54
CroDoBo	49.1	48.9	46.3	46.5	48.9	47.9	1.99

Table 6.12: Replacing main paper Eq.3 with other pseudo-labeling methods(%) on VisDA-C.

Methods (Syn → Real)	plane	bike	bus	car	horse	knife	motor	person	plant	skate	train	truck	Online
w^u: MixMatch [101] w^v: FixMatch [145]	93.2	80.9	85.6	67.1	94.1	10.3	88.4	77.9	92.3	91.9	85.7	35.9	74.3
w^u, w^v: MixMatch [101]	94.7	83.3	81.0	62.4	90.7	13.8	84.8	78.7	95.6	94.6	82.9	45.4	71.6
CroDoBo	94.8	87.5	90.5	76.0	94.9	93.7	88.7	80.1	94.8	89.4	84.6	30.7	79.4

Table 6.13: Performance sensitivity (%) to hyperparameter λ (weight for diversity loss) on VisDA-C [165], τ=0.95.

Metric/λ	0.1	0.4	0.5	0.8	1.0	mean	var
Online	74.9	79.4	78.7	78.5	78.4	78.0	3.1
One-pass	80.2	84.0	83.4	83.6	83.5	82.9	2.4

Table 6.14: Performance sensitivity (%) to hyperparameter τ (confidence threshold for pseudo-label selection in main paper Eq.2 on VisDA-C [165], λ=0.4.

Metric/τ	0.5	0.6	0.7	0.8	0.9	0.95	mean	var
Online	75.0	76.7	77.3	77.9	78.4	79.4	77.5	2.3
One-pass	80.9	81.7	82.6	82.8	83.4	84.0	82.7	2.0

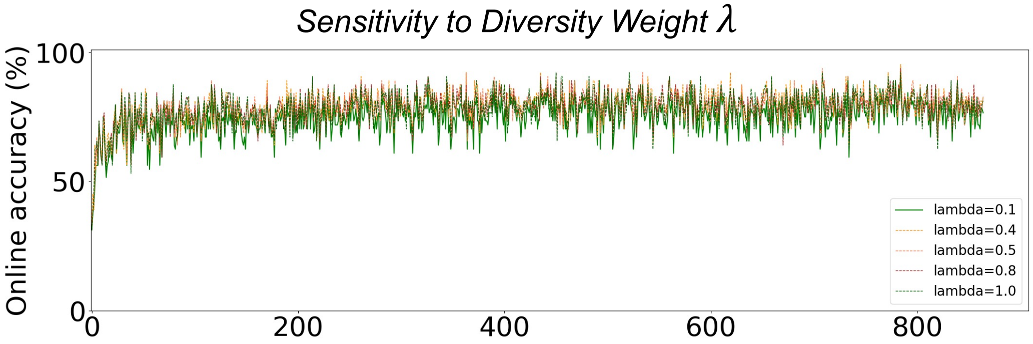

Figure 6.8: Results of online accuracy *w.r.t.* sensitivity to hyperparameter λ for the diversity term on VisDA-C [165] using ResNet-101.

Hyperparameters We have two hyperparameters in the proposed approach: λ for weighing the term ℓ_{div} and τ for the pseudo-label selection. We used λ=0.4 and τ=0.95 in all our experiments, here we report results on more settings of these hyperparameters. The results of λ={0.1, 0.4, 0.5, 0.8, 1.0} are shown in Table 6.13. As the results suggest, CroDoBo is not sensitive to hyperparameter λ. We observe similar performance of the model when λ is larger than 0.4.

The sensitivity to τ is shown in Table 6.14. When τ is smaller, more samples in each target query are selected as pseudo-labels to co-supervise the peer learner. However, the quality of these pseudo-labels is compromised since the model is less confident about the prediction. Thus, the co-supervision is less accurate to depend on. We observe the performance drop when the threshold τ is smaller than 0.6. Therefore, we suggest a larger threhold τ to achieve a more effective model. The online accuracy of the above settings are shown in Figure 6.8 and Figure 6.9.

Network Architecture. We follow the network architecture in [162, 163], a feature backbone followed by a bottleneck layer with dimension=256, and a Linear layer as the output layer. For the experiments on VisDA-C [165], COVID-DA [166] and Fashion-MNIST-to-DeepFashion [1, 168], the feature backbone is pretrained on ImageNet [141]. For the *WILDS*-Camelyon17 benchmark, we followed the leaderboard to use a randomly initialized DenseNet-121 [209]. We use Adam [215]

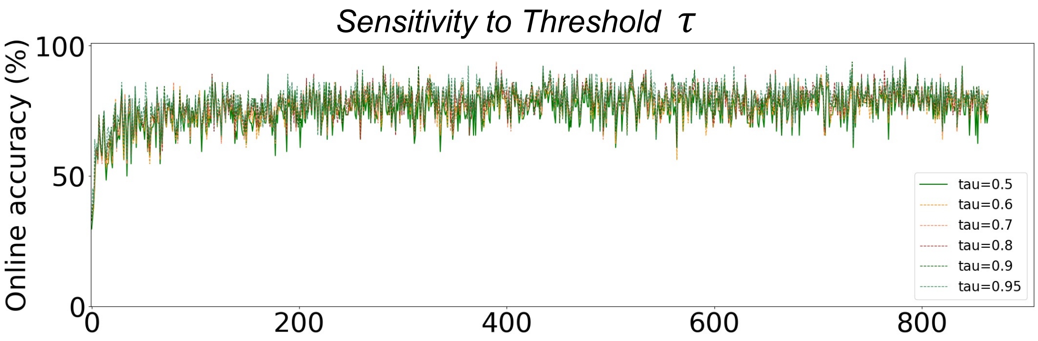

Figure 6.9: Results of online accuracy *w.r.t.* sensitivity to hyperparameter τ for pseudo-label threshold on VisDA-C [165] using ResNet-101.

Table 6.15: Ablation study of *RandAug* and the multiple forward of each target query on VisDA-C and Fashion dataset.

Dataset	VisDA-C		Fashion	
	No Aug	*RandAug*+multiple use of target	No Aug	*RandAug*+multiple use of target
DAN [107]	57.8	68.4	40.7	45.2
CORAL [210]	66.7	72.1	40.4	37.1
DANN [10]	49.0	49.9	35.6	37.2
ENT [32]	55.8	46.1	31.9	31.3
MDD [160]	60.4	67.0	36.5	39.0
CDAN [13]	62.3	62.8	45.4	41.0
CroDoBo K=2 (Ours)	**77.9**	**79.4**	**47.6**	**49.1**

with with an initial learning rate of 8e-4. The query size in our experiments is set as 64. We have not observed any major performance change using different batch-size. Results are reported based on an average of 5 runs.

Ablation study. We conduct ablation study on the impact of cross-domain bootstrapping in Table 6.5. Following Table 6.1, we provide the VisDA-C one-pass accuracy in class average. This study is to evaluate whether the improvement is introduced by cross-domain bootstrapping or simply the strong baseline with the objectives on the target domain (see Section 6.3.2.2). Thus, we devise a baseline by removing only the cross-domain bootstrapping, called w/o CroDoBo. The baseline model has one learner that is optimized by minimizing the objective $\ell_s + \ell_t + \ell_{\text{ent}} + \lambda\ell_{\text{div}}$, where $\ell_t = B^{-1}\sum_{b=1}^{B} \nVdash (p_b \geq \tau)\,\mathcal{H}\left(\hat{y}_b; p(c|\tilde{t}_b; \boldsymbol{w})\right)$, which is Eq. (6.2) without exchanging the pseudo-labels. In Table 6.5, we observe that w/ CroDoBo is consistently better than w/o in the online average accuracy on all the datasets. Regarding one-pass accuracy, the effectiveness of cross-domain bootstrapping is unapparent on smaller datasets VisDA-C and COVID-DA, yet clearly outperforms w/o on large-scale *WILDS*-Camelyon17 and Fashion-MNIST-to-DeepFashion.

We further conduct ablation study on the objective terms (see Section 6.3.2.2) and report the results in Table 6.6. To eliminate the benefit of cross-domain boosting, our default setting is the model w/o CRODOBO. We leave out ℓ_{ent} and observe significant performance drop. Without ℓ_{div}, the performance decrease slight in the online metric, but far more sharply on the one-pass metric (which is calculated per-class). We analyze that the diversity term is important for imbalanced dataset like VisDA-C to achieve high class-average accuracy. We also report the results by replacing ℓ_{ent} and ℓ_{div} with Pseudo-labeling [114]. We replace either $\{\ell_{\text{ent}}, \ell_{\text{div}}\}$ or ℓ_t with MixMatch, and observe decent performance when employed together with $\{\ell_{\text{ent}}, \ell_{\text{div}}\}$ (see Table 6.6 row6). The RandAugment [207] on the entropy and diversity terms does not enhance the performance.

Number of Learners $K \geq$ 3. We report the results of CroDoBo with varying number of learners $K \in \{2, 3, 4, 5\}$ on VisDA-C in Table 6.7. We observe that when K=3 the performance is consistent with K=2. However, from K=4 the performance is improved with more learners with discrepancies. This observation reflects the effectiveness to exploit the discrepant learners via bootstrapping and co-supervision. The choice of K is a trade-off between computation cost and performance. We find that K=2 is sufficient to yield state-of-the-art performance in most times, thus is a better choice considering its computation efficiency.

Multi-Source CroDoBo. Since the proposed method exploits the learners' discrepancy, a natural extension of the proposed method is to use multiple source to obtain more discrepant co-supervisions. We experimented on VisDA-C with one learner taking from an additional source domain from the Youtube Bounding Box dataset [165]. For fair comparison, we randomly select a subset of the source samples to have equal total number of source samples for both multi-source and single-source settings. The results are reported in Table 6.8. Multi-Source CroDoBo improves the class average accuracy to a remarkable 87.8%. The result further validates the effectiveness to increase data diversity.

As clarified in Section 6.3.2, *RandAug* is only employed to increase the data diversity, and is not required for the proposed method. We note that the use of *RandAug* and the multiple use of each target query in the proposed method might lead to confusion. To better evaluate the proposed method, besides providing CroDoBo without any augmentation, here we further provide the augmented baseline results, and with multiple use of each target query with two strong and two weak augmented versions. We search the best performing hyperparameters for each method using grid-search. We observe that (Table 6.15) either CroDoBo or CroDoBo+ outperforms the compared baselines.

6.5 Conclusion

In the context of the *the right to be forgotten*, we propose an online domain adaptation framework in which the target data is erased immediately after prediction. A novel online UDA algorithm is proposed to tackle the lack of data diversity, which is a fundamental drawback of the online setting. The proposed method achieves state-of-the-art online results and comparable results to the offline domain adaptation approaches. We would like to extend CroDoBo to more tasks like semantic segmentation [216, 217].

Chapter 7: Conclusion and Future Research

7.1 Summary

In this dissertation, we presented the several techniques for handling practical out-of-distribution shifts in deep learning systems via the problem of domain adaptation. We reviewed the background and necessary preliminaries for domain adaptation. We presented approaches in three scenarios in domain adaptation. We presented a curriculum-based adversarial learning approach for multi-source domain adaptation. We showed the effectiveness of using an adversarial agent with experiments, and illustrate the visualizations of how the agent works during training. We also presented a semi-supervised domain adaptation approach that optimize the utility of target ground-truths, via the deep co-training of the two created subtasks. In the end, we presented a practical setting for online domain adaptation that aims at privacy preservation, and an approach that aims to increase the data diversity via source bootstrapping.

7.2 Conclusions

Through studying representation learning under distribution shifts, in recent years we have witnessed an evolution of related tasks and methods. We first conclude our observations on the recent changes in domain adaptation, and then we share our analysis on how it is connected to a series of different but related fields.

First, the domain adaptation methods have evolved from the conventional objective which aims to enclose the distribution divergence, to a non-conservative objective which aims at the unsupervised learning of the structure of merely the target domain. Is is not hard to understand why in-domain supervisions are superior to across-domain supervisions. Therefore, if we have effective ways to explore the unlabeled target domain, source domain are less worth exploring. Consequently, we have observed less instance weighting-based techniques that try to maximize the utility of the source domain. In fact, many recent state-of-the-art techniques on domain adaptation benefit from the unsupervised and semi-supervised approaches [32, 162, 164, 192, 218]. Many direct extensions from the semi-supervised techniques to the domain adaptation setting have been successful. The mean teacher learning model [219], domain adaptation via Mix-Up [111], contrastive learning [220], self-training [221] and co-teaching [164]. The recent re-use of the infomax objective and deep clustering has enabled source-free domain adaptation [162], in which the source domain is merely used to initialize the network. We observe that the thriving unsupervised learning techniques have result in the recent shift in the field of domain adaptation, and will continue to lead in the near future.

Second, we observe similar phenomenon in the related fields like continual learning, also known as life-long learning. Apart from the effectiveness of the replay-based methods [222] to alleviate the catastrophic forgetting, recent continual learning also benefit from unsupervised and semi-supervised learning techniques such as the contrastive learning [223] and pseudo-labeling [224]. Similarly in the filed of federated learning, which also relies on knowledge transfer, we observe a success applying knowledge distillation [225, 226, 227] and constrastive learning [228]. In brief, we observe that the unique challenge of these tasks have been well explored initially, and there are more opportunities exploring the representation learning techniques in combination with the specific constraints of a task in the near future.

Third, we observe a common need for robustness and privacy in domain adaptation, continual learning and federated learning, which indicates the progress of these fields towards practicality. In domain adaptation, we used to assume the source and target data share the same classes, which may not be realistic. Recently, efforts have been made to relax this assumption. Partial domain adaptation [65] adapts from a larger source categories to the target sub-set categories. Open-set and universal [229] domain adaptation further relaxes this constraints as long as source and target domain share classes. In [226], the i.i.d. assumption for federated learning is relaxed. In [230], the privacy constraint has been escalated to individual device for a single person. In brief, in the context of neural network, we need flexibility. We need to enable our models to embrace changes and the differences in personalized needs.

In conclusion, the efficient exploration of one of these tasks requires horizontal comparison of the other related tasks, as well as guidance from the "upstream" tasks. In recent years, the development of new methods, new settings is rather rapid. To keep up, one also needs to keep rethinking what is in need for a task besides what yields the state-of-the-art performance.

7.3 Future Research Directions

Label-free Evaluation with Distribution Shift: Unsupervised domain adaptation assumes a label-rich source domain and a label-scarce target domain. Current methods that assume labels are available on the target domain at test-time, are not realistic in practice. Moreover, if there is a small bunch of target labels available, they are better exploited as supervision during training than being used as test labels. The label-free accuracy estimation is desirable in many practical situations, especially when the target distribution is constantly changing with time. However, this direction has not drawn much attention currently. Hence, it is of interest to develop better label-free evaluation metrics that enables better flexibility of domain adaptation in practice.

Test-time Domain Adaptation: Many domain adaptation works have underestimated the importance of time efficiency in domain adaptation. Instead, to be competitive in performance and to achieve state-of-the-art, many approaches were established based on tens of thousands of iterations. In reality, the target domain is without labels not just because of annotation budget limit, time is another important factor. In the problem of test-time domain adaptation, one can adapt from the model weights trained on the source domain to the target data at test time. Although the current approaches yield relatively poor performance to win the time efficiency, it is worth exploring due to its value in practice. Unfortunately, this direction has not drawn as much attention as the conventional domain adaptation settings. Hence, it is of interest to improve the test-time domain adaptation approaches, especially to explore better approaches that balance time, performance and data availability.

Noisy-label Learning: Noisy-label learning or noisy training, a seemingly different problem from domain adaptation, is actually related in many less obvious ways. In domain adaptation, the major mission is to alleviate the lack of annotation of the target domain. In noisy-label learning, the mission is similar: to alleviate the effect of bad annotations of a dataset, which can be the target domain as well. Like many domain adaptation works, noisy-label learning approaches are also interested in finding “trustworthy” samples that can serve as anchor points during training. The main difference is that most noisy-label learning settings don’t assume a cross-domain scenario, which is worth considering: can we use an auxiliary dataset with clean labels but with domain shift, to help clean a poorly annotated dataset. In reality, since there are many publicly available datasets with good annotations, this is often possible. Hence, it is of interest to explore the possibility of noisy-label learning under distribution shift.

Bibliography

[1] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[2] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[3] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019.

[4] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[5] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[6] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[7] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5542–5550, 2017.

[8] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019.

[9] Jin Chen, Xinxiao Wu, Lixin Duan, and Shenghua Gao. Domain adversarial reinforcement learning for partial domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

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

[11] Jian Hu, Hongya Tuo, Chao Wang, Lingfeng Qiao, Haowen Zhong, Junchi Yan, Zhongliang Jing, and Henry Leung. Discriminative partial domain adversarial network. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 632–648. Springer, 2020.

[12] Maxime W Lafarge, Josien PW Pluim, Koen AJ Eppenhof, Pim Moeskops, and Mitko Veta. Domain-adversarial neural networks to address the appearance variability of histopathology images. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 83–91. Springer, 2017.

[13] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *arXiv preprint arXiv:1705.10667*, 2017.

[14] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[15] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

[16] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019.

[17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[18] Eduard Fosch Villaronga, Peter Kieseberg, and Tiffany Li. Humans forget, machines remember: Artificial intelligence and the right to be forgotten. *Computer Law & Security Review*, 34(2):304–313, 2018.

[19] Sanjam Garg, Shafi Goldwasser, and Prashant Nalini Vasudevan. Formalizing data deletion in the context of the right to be forgotten. *Advances in Cryptology–EUROCRYPT 2020*, 12106:373, 2020.

[20] Ugo Pagallo and Massimo Durante. Human rights and the right to be forgotten. In *Human Rights, Digital Society and the Law*, pages 197–208. Routledge, 2019.

[21] Noam Tirosh. Reconsidering the 'right to be forgotten'–memory rights and the right to memory in the new media era. *Media, Culture & Society*, 39(5):644–660, 2017.

[22] Kristie Byrum. *The European Right to be Forgotten: The First Amendment Enemy*. Rowman & Littlefield, 2018.

[23] Eugenia Politou, Alexandra Michota, Efthimios Alepis, Matthias Pocs, and Constantinos Patsakis. Backups and the right to be forgotten in the gdpr: An uneasy relationship. *Computer Law & Security Review*, 34(6):1247–1257, 2018.

[24] Eugenia Politou, Efthimios Alepis, Maria Virvou, and Constantinos Patsakis. The "right to be forgotten" in the gdpr: Implementation challenges and potential solutions. In *Privacy and Data Protection Challenges in the Distributed Era*, pages 41–68. Springer, 2022.

[25] MM Hassan Mahmud. On universal transfer learning. *Theoretical Computer Science*, 410(19):1826–1846, 2009.

[26] MTCAJ Thomas and A Thomas Joy. *Elements of information theory*. Wiley-Interscience, 2006.

[27] Changjian Shui, Qi Chen, Jun Wen, Fan Zhou, Christian Gagné, and Boyu Wang. Beyond h-divergence: Domain adaptation theory with jensen-shannon divergence. 2020.

[28] Shengjia Zhao, Abhishek Sinha, Yutong He, Aidan Perreault, Jiaming Song, and Stefano Ermon. H-divergence: A decision-theoretic probability discrepancy measure. 2020.

[29] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[30] Matthias Schröder. Admissible representations for probability measures. *Mathematical Logic Quarterly*, 53(4-5):431–445, 2007.

[31] Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1134–1144, 2021.

[32] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8050–8058, 2019.

[33] Wouter M Kouw and Marco Loog. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):766–785, 2019.

[34] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.

[35] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018.

[36] Seungmin Lee, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong. Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 91–100, 2019.

[37] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

[38] Kaichao You, Ximei Wang, Mingsheng Long, and Michael Jordan. Towards accurate model selection in deep unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 7124–7133, 2019.

[39] Cheng Ouyang, Konstantinos Kamnitsas, Carlo Biffi, Jinming Duan, and Daniel Rueckert. Data efficient unsupervised domain adaptation for cross-modality image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 669–677. Springer, 2019.

[40] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 627–636, 2019.

[41] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2239–2247, 2019.

[42] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.

[43] Rae Jeong, Yusuf Aytar, David Khosid, Yuxiang Zhou, Jackie Kay, Thomas Lampe, Konstantinos Bousmalis, and Francesco Nori. Self-supervised sim-to-real adaptation for visual robotic manipulation. *arXiv preprint arXiv:1910.09470*, 2019.

[44] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *International Journal of Computer Vision*, pages 1–47, 2019.

[45] Jae Shin Yoon, Takaaki Shiratori, Shoou-I Yu, and Hyun Soo Park. Self-supervised adaptation of high-fidelity face models for monocular performance tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4601–4609, 2019.

[46] Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 6670–6680, 2017.

[47] Tao Wang, Xiaopeng Zhang, Li Yuan, and Jiashi Feng. Few-shot adaptive faster r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7173–7182, 2019.

[48] Junyi Zhang, Ziliang Chen, Junying Huang, Liang Lin, and Dongyu Zhang. Few-shot structured domain adaptation for virtual-to-real scene parsing. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

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

[50] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery.

[51] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[52] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[53] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision*, pages 624–639, 2018.

[54] Zhengming Ding, Nasser M Nasrabadi, and Yun Fu. Semi-supervised deep domain adaptation via coupled neural networks. *IEEE Transactions on Image Processing*, 27(11):5214–5224, 2018.

[55] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2507–2516, 2019.

[56] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018.

[57] Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J Gordon. On learning invariant representation for domain adaptation. *arXiv preprint arXiv:1901.09453*, 2019.

[58] Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8156–8164, 2018.

[59] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. *arXiv preprint arXiv:1812.01754*, 2018.

[60] Massimiliano Mancini, Lorenzo Porzi, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Boosting domain adaptation by discovering latent domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3771–3780, 2018.

[61] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *Advances in Neural Information Processing Systems*, pages 8559–8570, 2018.

[62] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3964–3973, 2018.

[63] Caiming Xiong, Scott McCloskey, Shao-Hang Hsieh, and Jason J Corso. Latent domains modeling for visual domain adaptation. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[64] Qingchao Chen, Yang Liu, Zhaowen Wang, Ian Wassell, and Kevin Chetty. Re-weighted adversarial adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7976–7985, 2018.

[65] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Partial transfer learning with selective adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2724–2732, 2018.

[66] Abhishek Kumar, Prasanna Sattigeri, Kahini Wadhawan, Leonid Karlinsky, Rogerio Feris, Bill Freeman, and Gregory Wornell. Co-regularized alignment for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 9345–9356, 2018.

[67] Zuxuan Wu, Xin Wang, Joseph E Gonzalez, Tom Goldstein, and Larry S Davis. Ace: Adapting to changing environments for semantic segmentation. *arXiv preprint arXiv:1904.06268*, 2019.

[68] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *International Conference on Machine Learning*, pages 4013–4022, 2019.

[69] Lixin Duan, Dong Xu, and Shih-Fu Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1338–1345. IEEE, 2012.

[70] Lixin Duan, Dong Xu, and Ivor Wai-Hung Tsang. Domain adaptation from multiple sources: A domain-dependent regularization approach. *IEEE Transactions on Neural Networks and Learning Systems*, 23(3):504–518, 2012.

[71] Lixin Duan, Ivor W Tsang, Dong Xu, and Tat-Seng Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 289–296. ACM, 2009.

[72] Himanshu S Bhatt, Arun Rajkumar, and Shourya Roy. Multi-source iterative adaptation for cross-domain classification. In *IJCAI*, pages 3691–3697, 2016.

[73] Massimiliano Mancini, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Adagraph: Unifying predictive and continuous domain adaptation through graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6568–6577, 2019.

[74] Yang Shu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Transferable curriculum for weakly-supervised domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4951–4958, 2019.

[75] Vinod Kumar Kurmi, Vipul Bajaj, Venkatesh K Subramanian, and Vinay P Namboodiri. Curriculum based dropout discriminator for domain adaptation. *arXiv preprint arXiv:1907.10628*, 2019.

[76] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 137–144. MIT Press, 2007.

[77] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 129–136. Curran Associates, Inc., 2008.

[78] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073. IEEE, 2012.

[79] Boqing Gong, Kristen Grauman, and Fei Sha. Learning kernels for unsupervised domain adaptation with applications to visual object recognition. *JMLR*, 109(1-2):3–27, 2014.

[80] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018.

[81] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018.

[82] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M Alvarez. Effective use of synthetic data for urban scene semantic segmentation. In *ECCV*. Springer, 2018.

[83] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.

[84] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016.

[85] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *CVPR*, 2018.

[86] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1992–2001, 2017.

[87] Yan Wang, Chen Xiangyu, You Yurong, Erran Li Li, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Chao Wei-Lun. Train in germany, test in the usa: Making 3d object detectors generalize. In *CVPR*, 2020.

[88] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018.

[89] Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *ECCV*, 2020.

[90] Can Qin, Lichen Wang, Qianqian Ma, Yu Yin, Huan Wang, and Yun Fu. Opposite structure learning for semi-supervised domain adaptation. *arXiv preprint arXiv:2002.02545*, 2020.

[91] Shuang Ao, Xiang Li, and Charles X Ling. Fast generalized distillation for semi-supervised domain adaptation. In *AAAI*, 2017.

[92] Limin Li and Zhenyue Zhang. Semi-supervised domain adaptation by covariance matching. *TPAMI*, 41(11):2724–2739, 2018.

[93] Da Li and Timothy Hospedales. Online meta-learning for multi-source and semi-supervised domain adaptation. *arXiv preprint arXiv:2004.04398*, 2020.

[94] Jeff Donahue, Judy Hoffman, Erik Rodner, Kate Saenko, and Trevor Darrell. Semi-supervised domain adaptation with instance constraints. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 668–675, 2013.

[95] Ting Yao, Yingwei Pan, Chong-Wah Ngo, Houqiang Li, and Tao Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *CVPR*, 2015.

[96] Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296, 2005.

[97] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

[98] Maria-Florina Balcan, Avrim Blum, and Ke Yang. Co-training and expansion: Towards bridging theory and practice. In *NIPS*, 2005.

[99] Minmin Chen, Kilian Q Weinberger, and Yixin Chen. Automatic feature decomposition for single view co-training. In *ICML*, 2011.

[100] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018.

[101] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.

[102] Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. Theoretical analysis of self-training with deep networks on unlabeled data. *arXiv preprint arXiv:2010.03622*, 2020.

[103] Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning*, pages 5468–5479. PMLR, 2020.

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

[105] Siyuan Qiao, W. Shen, Zhishuai Zhang, Bo Wang, and A. Yuille. Deep co-training for semi-supervised image recognition. In *ECCV*, 2018.

[106] Minmin Chen, Kilian Q Weinberger, and John Blitzer. Co-training for domain adaptation. In *NIPS*, 2011.

[107] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

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

[109] Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. Transferable attention for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5345–5352, 2019.

[110] Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5495–5504, 2018.

[111] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *AAAI*, 2020.

[112] Luyu Yang, Yogesh Balaji, Ser-Nam Lim, and Abhinav Shrivastava. Curriculum manager for source selection in multi-source domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 608–624. Springer, 2020.

[113] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. *arXiv preprint arXiv:1711.01575*, 2017.

[114] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[115] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *ACL*, 2006.

[116] David McClosky, Eugene Charniak, and Mark Johnson. Reranking and self-training for parser adaptation. In *ACL*, 2006.

[117] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th annual meeting on association for computational linguistics*, pages 26–33. Association for Computational Linguistics, 2001.

[118] Seunghyeon Kim, Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In *ICCV*, 2019.

[119] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018.

[120] Mehran Khodabandeh, Arash Vahdat, Mani Ranjbar, and William G Macready. A robust learning approach to domain adaptive object detection. In *ICCV*, 2019.

[121] Qingyi Tao, Hao Yang, and Jianfei Cai. Zero-annotation object detection with web knowledge transfer. In *ECCV*, 2018.

[122] Jian Liang, Ran He, Zhenan Sun, and Tieniu Tan. Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation. In *CVPR*, 2019.

[123] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *CVPR*, 2018.

[124] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[125] Yuhua Tang, Zhipeng Lin, Haotian Wang, and Liyang Xu. Adversarial mixup synthesis training for unsupervised domain adaptation. In *ICASSP*, 2020.

[126] Xudong Mao, Yun Ma, Zhenguo Yang, Yangbin Chen, and Qing Li. Virtual mixup training for unsupervised domain adaptation. *arXiv preprint arXiv:1905.04215*, 2019.

[127] Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*, 2020.

[128] Judy Hoffman, Erik Rodner, Jeff Donahue, Trevor Darrell, and Kate Saenko. Efficient learning of domain-invariant image representations. In *ICLR*, 2013.

[129] Luis AM Pereira and Ricardo da Silva Torres. Semi-supervised transfer subspace for domain adaptation. *Pattern Recognition*, 75:235–249, 2018.

[130] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015.

[131] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[132] Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, Subhransu Maji, and Manmohan Chandraker. Active adversarial domain adaptation. In *WACV*, 2020.

[133] Viraj Prabhu, Arjun Chandrasekaran, Kate Saenko, and Judy Hoffman. Active domain adaptation via clustering uncertainty-weighted embeddings. *arXiv preprint arXiv:2010.08666*, 2020.

[134] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[135] Zhijun Mai, Guosheng Hu, Dexiong Chen, Fumin Shen, and Heng Tao Shen. Metamixup: Learning adaptive interpolation policy of mixup with meta-learning. *arXiv preprint arXiv:1908.10059*, 2019.

[136] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[137] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[138] Zhiyong Huang, Kekai Sheng, Weiming Dong, Xing Mei, Chongyang Ma, Feiyue Huang, Dengwen Zhou, and Changsheng Xu. Effective label propagation for discriminative semi-supervised domain adaptation. *arXiv preprint arXiv:2012.02621*, 2020.

[139] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[140] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[141] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[142] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[143] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.

[144] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR, 2019.

[145] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

[146] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[147] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

[148] Jeffrey Rosen. The right to be forgotten. *Stan. L. Rev. Online*, 64:88, 2011.

[149] Huifeng Guo, Bo Chen, Ruiming Tang, Weinan Zhang, Zhenguo Li, and Xiuqiang He. An embedding learning framework for numerical features in ctr prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2910–2918, 2021.

[150] Ningyu Zhang, Qianghuai Jia, Shumin Deng, Xiang Chen, Hongbin Ye, Hui Chen, Huaixiao Tou, Gang Huang, Zhao Wang, Nengwei Hua, et al. Alicg: Fine-grained and evolvable conceptual graph construction for semantic search at alibaba. *arXiv preprint arXiv:2106.01686*, 2021.

[151] Xiaoyu Zhang, Xiaofeng Chen, Joseph K Liu, and Yang Xiang. Deeppar and deepdpa: privacy preserving and asynchronous deep learning for industrial iot. *IEEE Transactions on Industrial Informatics*, 16(3):2081–2090, 2019.

[152] Alexander Ziller, Dmitrii Usynin, Rickmer Braren, Marcus Makowski, Daniel Rueckert, and Georgios Kaissis. Medical imaging deep learning with differential privacy. *Scientific Reports*, 11(1):1–8, 2021.

[153] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[154] Georgios A Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, 2020.

[155] Ligeng Zhu and Song Han. Deep leakage from gradients. In *Federated learning*, pages 17–31. Springer, 2020.

[156] Xiao Jin, Pin-Yu Chen, Chia-Yi Hsu, Chia-Mu Yu, and Tianyi Chen. Cafe: Catastrophic data leakage in vertical federated learning. *arXiv preprint arXiv:2110.15122*, 2021.

[157] Xiaoyun Xu, Jingzheng Wu, Mutian Yang, Tianyue Luo, Xu Duan, Weiheng Li, Yanjun Wu, and Bin Wu. Information leakage by model weights on federated learning. In *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*, pages 31–36, 2020.

[158] Doyen Sahoo, Quang Pham, Jing Lu, and Steven CH Hoi. Online deep learning: Learning deep neural networks on the fly. *arXiv preprint arXiv:1711.03705*, 2017.

[159] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.

[160] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413. PMLR, 2019.

[161] Qing Wang, Wei Rao, Sining Sun, Leib Xie, Eng Siong Chng, and Haizhou Li. Unsupervised domain adaptation via domain adversarial training for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4889–4893. IEEE, 2018.

[162] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020.

[163] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2021.

[164] Luyu Yang, Yan Wang, Mingfei Gao, Abhinav Shrivastava, Kilian Q Weinberger, Wei-Lun Chao, and Ser-Nam Lim. Deep co-training with task decomposition for semi-supervised domain adaptation. *arXiv preprint arXiv:2007.12684*, 2020.

[165] Xingchao Peng, Ben Usman, Neela Kaushik, Dequan Wang, Judy Hoffman, and Kate Saenko. Visda: A synthetic-to-real benchmark for visual domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2021–2026, 2018.

[166] Yifan Zhang, Shuaicheng Niu, Zhen Qiu, Ying Wei, Peilin Zhao, Jianhua Yao, Junzhou Huang, Qingyao Wu, and Mingkui Tan. Covid-da: Deep domain adaptation from typical pneumonia to covid-19. *arXiv preprint arXiv:2005.01577*, 2020.

[167] Pang Wei Koh, Shiori Sagawa, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[168] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1096–1104, 2016.

[169] Mahmood Ali and Lloyd Miller. Erp system implementation in large enterprises–a systematic literature review. *Journal of Enterprise Information Management*, 2017.

[170] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Does ai remember? neural networks and the right to be forgotten. 2020.

[171] Stephen J Elliott and Boaz Rafaely. Frequency-domain adaptation of causal digital filters. *IEEE Transactions on Signal processing*, 48(5):1354–1364, 2000.

[172] Mark Dredze and Koby Crammer. Online methods for multi-domain learning and adaptation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 689–697, 2008.

[173] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, and Hong-Jiang Zhang. Two-dimensional multilabel active learning with an efficient online adaptation model for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1880–1897, 2008.

[174] Vidit Jain and Erik Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *CVPR 2011*, pages 577–584. IEEE, 2011.

[175] Massimiliano Mancini, Hakan Karaoguz, Elisa Ricci, Patric Jensfelt, and Barbara Caputo. Kitting in the wild through online domain adaptation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1103–1109. IEEE, 2018.

[176] Adrien Gaidon and Eleonora Vig. Online domain adaptation for multi-object tracking. *arXiv preprint arXiv:1508.00776*, 2015.

[177] Jiaolong Xu, David Vázquez, Krystian Mikolajczyk, and Antonio M López. Hierarchical online domain adaptation of deformable part-based models. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5536–5541. IEEE, 2016.

[178] Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

[179] JH Moon, Debasmit Das, and CS George Lee. Multi-step online unsupervised domain adaptation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 41172–41576. IEEE, 2020.

[180] Rita Delussu, Lorenzo Putzu, Giorgio Fumera, and Fabio Roli. Online domain adaptation for person re-identification with a human in the loop. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3829–3836. IEEE, 2021.

[181] Loris Nanni, Stefano Ghidoni, and Sheryl Brahnam. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, 71:158–172, 2017.

[182] Abu Md Niamul Taufique, Chowdhury Sadman Jahan, and Andreas Savakis. Conda: Continual unsupervised domain adaptation. *arXiv preprint arXiv:2103.11056*, 2021.

[183] Piyush Rai, Avishek Saha, Hal Daumé III, and Suresh Venkatasubramanian. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, pages 27–32, 2010.

[184] Avishek Saha, Piyush Rai, Hal Daumé, Suresh Venkatasubramanian, and Scott L DuVall. Active supervised domain adaptation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 97–112. Springer, 2011.

[185] Xinhong Ma, Junyu Gao, and Changsheng Xu. Active universal domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8968–8977, 2021.

[186] Giona Matasci, Devis Tuia, and Mikhail Kanevski. Svm-based boosting of active learning strategies for efficient domain adaptation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(5):1335–1343, 2012.

[187] Yining Chen, Haipeng Luo, Tengyu Ma, and Chicheng Zhang. Active online learning with hidden shifting domains. In *International Conference on Artificial Intelligence and Statistics*, pages 2053–2061. PMLR, 2021.

[188] Claudio Persello and Lorenzo Bruzzone. Active learning for domain adaptation in the supervised classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 50(11):4468–4483, 2012.

[189] Cheng Deng, Xianglong Liu, Chao Li, and Dacheng Tao. Active multi-kernel domain adaptation for hyperspectral image classification. *Pattern Recognition*, 77:306–315, 2018.

[190] Viraj Prabhu, Arjun Chandrasekaran, Kate Saenko, and Judy Hoffman. Active domain adaptation via clustering uncertainty-weighted embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8505–8514, 2021.

[191] Thomas Varsavsky, Mauricio Orbes-Arteaga, Carole H Sudre, Mark S Graham, Parashkev Nachev, and M Jorge Cardoso. Test-time unsupervised domain adaptation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 428–436. Springer, 2020.

[192] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2020.

[193] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pages 9229–9248. PMLR, 2020.

[194] Shuo Wang, Leandro L Minku, and Xin Yao. Resampling-based ensemble methods for online class imbalance learning. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1356–1368, 2014.

[195] Roberto Souto Maior de Barros, Silas Garrido T de Carvalho Santos, and Paulo Mauricio Gonçalves Júnior. A boosting-like online learning ensemble. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1871–1878. IEEE, 2016.

[196] Leandro L Minku, Allan P White, and Xin Yao. The impact of diversity on online ensemble learning in the presence of concept drift. *IEEE Transactions on knowledge and Data Engineering*, 22(5):730–742, 2009.

[197] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2018.

[198] Nikunj C Oza and Stuart J Russell. Online bagging and boosting. In *International Workshop on Artificial Intelligence and Statistics*, pages 229–236. PMLR, 2001.

[199] Bohyung Han, Jack Sim, and Hartwig Adam. Branchout: Regularization for online ensemble tracking with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3356–3365, 2017.

[200] Ghazal Jaber, Antoine Cornuéjols, and Philippe Tarroux. Online learning: Searching for the best forgetting strategy under concept drift. In *International Conference on Neural Information Processing*, pages 400–408. Springer, 2013.

[201] Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. *arXiv preprint arXiv:1808.01204*, 2018.

[202] Ian Osband. Risk versus uncertainty in deep learning: Bayes, bootstrap and the dangers of dropout. In *NIPS workshop on bayesian deep learning*, volume 192, 2016.

[203] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[204] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. *arXiv preprint arXiv:1705.07832*, 2017.

[205] David Warde-Farley, Ian J Goodfellow, Aaron Courville, and Yoshua Bengio. An empirical analysis of dropout in piecewise linear networks. *arXiv preprint arXiv:1312.6197*, 2013.

[206] Alexander Schmitz, Yusuke Bansho, Kuniaki Noda, Hiroyasu Iwata, Tetsuya Ogata, and Shigeki Sugano. Tactile object recognition using deep learning and dropout. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 1044–1050. IEEE, 2014.

[207] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[208] Preetum Nakkiran, Behnam Neyshabur, and Hanie Sedghi. The deep bootstrap framework: Good online learners are good offline generalizers. *arXiv preprint arXiv:2010.08127*, 2020.

[209] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

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

[211] Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

[212] Dongrui Wu. Online and offline domain adaptation for reducing bci calibration effort. *IEEE Transactions on human-machine Systems*, 47(4):550–563, 2016.

[213] Sahand Hajifar and Hongyue Sun. Online domain adaptation for continuous cross-subject liver viability evaluation based on irregular thermal data. *IISE Transactions*, pages 1–12, 2021.

[214] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[215] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[217] Sicheng Zhao, Bo Li, Xiangyu Yue, Yang Gu, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source domain adaptation for semantic segmentation. *arXiv preprint arXiv:1910.12181*, 2019.

[218] Luyu Yang, Mingfei Gao, Zeyuan Chen, Ran Xu, Abhinav Shrivastava, and Chetan Ramaiah. Burn after reading: Online adaptation for cross-domain streaming data. *arXiv preprint arXiv:2112.04345*, 2021.

[219] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *ICLR*, 2018.

[220] Kendrick Shen, Robbie M Jones, Ananya Kumar, Sang Michael Xie, Jeff Z HaoChen, Tengyu Ma, and Percy Liang. Connect, not collapse: Explaining contrastive learning for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 19847–19878. PMLR, 2022.

[221] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. *Advances in Neural Information Processing Systems*, 34:22968–22981, 2021.

[222] Yogesh Balaji, Mehrdad Farajtabar, Dong Yin, Alex Mott, and Ang Li. The effectiveness of memory replay in large scale continual learning. *arXiv preprint arXiv:2010.02418*, 2020.

[223] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9516–9525, 2021.

[224] Andrea Maracani, Umberto Michieli, Marco Toldo, and Pietro Zanuttigh. Recall: Replay-based continual learning in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7026–7035, 2021.

[225] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning*, pages 12878–12889. PMLR, 2021.

[226] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10174–10183, 2022.

[227] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

[228] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10713–10722, 2021.

[229] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. *arXiv preprint arXiv:2002.07953*, 2020.

[230] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.