A PROTOTYPE-ORIENTED CLUSTERING FOR DOMAIN SHIFT WITH SOURCE PRIVACY

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised clustering under domain shift (UCDS) studies how to transfer the knowledge from abundant unlabeled data from multiple source domains to learn the representation of the unlabeled data in a target domain. In this paper, we introduce Prototype-oriented Clustering with Distillation (PCD) to not only improve the performance and applicability of existing methods for UCDS, but also address the concerns on protecting the privacy of both the data and model of the source domains. PCD first constructs a source clustering model by aligning the distributions of prototypes and data. It then distills the knowledge to the target model through cluster labels provided by the source model while simultaneously clustering the target data. Finally, it refines the target model on the target domain data without guidance from the source model. Experiments across multiple benchmarks show the effectiveness and generalizability of our source-private clustering method.

1 INTRODUCTION

Supervised learning methods require a tremendous amount of labeled data, limiting their use cases in many situations (Adadi, 2021). By contrast, unsupervised clustering seeks to group similar data points into clusters without labels (Hartigan, 1972). Clustering has become one of the most popular methods in various applications, such as computer vision (Coleman and Andrews, 1979; Lei et al., 2018; Mittal et al., 2021), natural language processing (Biemann, 2006; Yoon et al., 2019), reinforcement learning (Mannor et al., 2004; Xu et al., 2014; Ahmadi et al., 2021), and multi-modal learning (Hu et al., 2019; Chen et al., 2021). In many of these applications, data naturally come from multiple sources and may not contain labels since they are expensive to acquire (Girshick et al., 2014; Lin et al., 2014). As an example, medical institutions collaborate to achieve a large and diverse dataset (Mojab et al., 2020). However, this partnership faces privacy and ownership challenges (Sheller et al., 2020). Across different domains, users may also have varying amounts of resources and data (Salehi et al., 2019). Another example is the inference-as-a-service paradigm, a business scheme where providers serve models trained on multiple sources of data as APIs ($e.g.$, Google AI platforms, Amazon Web Services, GPT-3 (Brown et al., 2020)) without giving clients direct access to them. To exploit the rich data from multiple domains for limited-data-and-resource users while also taking into account privacy challenges, one may consider applying methods from Unsupervised Domain Adaptation (UDA) (Shimodaira, 2000; Farhadi and Tabrizi, 2008; Saenko et al., 2010). These methods nonetheless require labeled data in the source domains, making them not applicable in many scenarios.

To overcome the assumption of UDA, Menapace et al. (2020) have recently introduced Unsupervised Clustering under Domain Shift (UCDS), a learning scenario where both the source and target domains have no labels. The goal of this problem setting is to transfer the knowledge from the abundant unlabeled data from multiple source domains to a target domain with limited data. To solve this problem, Menapace et al. (2020) propose Adaptive Clustering of Images under Domain Shift (ACIDS), a method that uses an information-theoretic loss (Ji et al., 2019) for clustering and batch normalization alignment (Li et al., 2016) for target adaptation. However, it has two major drawbacks. First, it assumes that we have full access to the source model parameters to initialize the target model before clustering, limiting its use in privacy-sensitive situations where access to the source model is restricted. Second, it requires batch normalization, a specific architectural design of the source model that may not be applicable in some recently proposed state-of-the-art models such as Vision Transformer (Dosovitskiy et al., 2020).

Table 1: Overview of different domain transfer settings.

	Source labels	Target labels	Source data access	Source model's parameters access
Unsupervised Domain adaptation	✓	✗	✓	✓
Source-free Unsupervised Domain Adaptation	✓	✗	✗	✓
Unsupervised Clustering under Domain Shift	✗	✗	✗	✓
Ours	✗	✗	✗	✗

In this paper, we consider a more practical problem that is a variant of UCDS (see Table 1): in addition to the data privacy, we also consider model privacy. Target data owners have no direct access to the source model but can query it to obtain cluster labels during target adaptation. This requirement is important because, given full access to the model, target users or other adversaries may exploit it to recover the source data, jeopardizing source data privacy Chen et al. (2019); Luo et al. (2020). To address this important and challenging problem, we propose Prototype-oriented Clustering with Distillation (PCD), a holistic method that consists of three stages. First, we construct a source clustering model from multiple-domain data. To achieve this, we use optimal transport (Kantorovich, 2006; Peyré and Cuturi, 2019) to align the distributions of data and prototypes, as well as a mutual-information maximization to assist the learning of the feature encoder and prototypes (Krause et al., 2010; Shi and Sha, 2012; Liang et al., 2020). Second, we use the target cluster assignments provided by the source model to distill the knowledge to the target model while simultaneously clustering the target data. Finally, we perform clustering on the target data alone to further refine the target model. Figure 1 illustrates the schematic diagram of our approach.

PCD achieves the following benefits. Our approach can be directly applied to the inference-as-a-service paradigm, which is becoming increasingly popular (Soifer et al., 2019). Many providers currently serve users with API services without sharing direct access to their models. Our method also protects the privacy of both the data and model in the source domains, which is especially critical in practical applications such as healthcare. Moreover, we no longer require the source and target models to share the same architecture, allowing for more flexibility in the training process. Unlike source data owners, target users may have limited resources and cannot afford to train large models.

Our main contributions include: **1)** We propose a generalized approach for tackling the problem of data-and-model private unsupervised clustering under domain shift. PCD integrates a prototype-oriented clustering algorithm and knowledge distillation into a unified method. Our clustering algorithm synergistically combines optimal transport with the mutual-information objective for prototype and data alignment. **2)** We verify the effectiveness and general applicability of the proposed method in practical settings: model transfer as well as limited-data and cluster-imbalanced scenarios. **3)** We provide comprehensive study and experiments on multiple datasets and demonstrate consistent gains over the baselines.

2 METHOD

To address the clustering problem under domain shift and privacy concerns, we provide a general recipe that consists of three main parts: **1)** source model learning: learn a transferable model that can guide the target model; **2)** target model clustering: train a target model with the knowledge from the source model as well as the target data; and **3)** target model refinement: refine the target model on the target data alone. The resulting strategy, referred to as PCD, can effectively solve the clustering problem under domain shift while fully preserving the privacy of the source data and model. We include the pseudocode in Algorithm 1 in Appendix E.

2.1 BACKGROUND

In unsupervised clustering under domain shift, we are given D unlabeled datasets from the source domains, denoted as $\mathcal{X}^s = \{\mathcal{X}_d^s\}_{d=1}^D$ where $\mathcal{X}_d^s = \{\boldsymbol{x}_{dj}^s\}_{j=1}^{n_d^s}$ represents a dataset from a source domain d with n_d^s samples. We are also given an unlabeled dataset from the target domain, denoted as $\mathcal{X}^t = \{\boldsymbol{x}_j^t\}_{j=1}^{n_t}$ with n_t target samples. There are K underlying clusters in both the source and target domains with similar semantic content, but there is a shift between the source and target data distributions. The clustering model consists of a feature encoder, $F_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}^{d_f}$, parameterized by $\boldsymbol{\theta}$, and a linear clustering head $C_{\boldsymbol{\mu}} : \mathbb{R}^{d_f} \to \mathbb{R}^K$, parameterized by $\boldsymbol{\mu}$. To simplify the notation, $G = C_{\boldsymbol{\mu}}(F_{\boldsymbol{\theta}}(\cdot))$ will denote the composition of the feature encoder and linear clustering head. We denote G^s and G^t as the source and target models, respectively. The goal is to learn a model that can discover the underlying clusters of target samples under domain shift. Although the existing

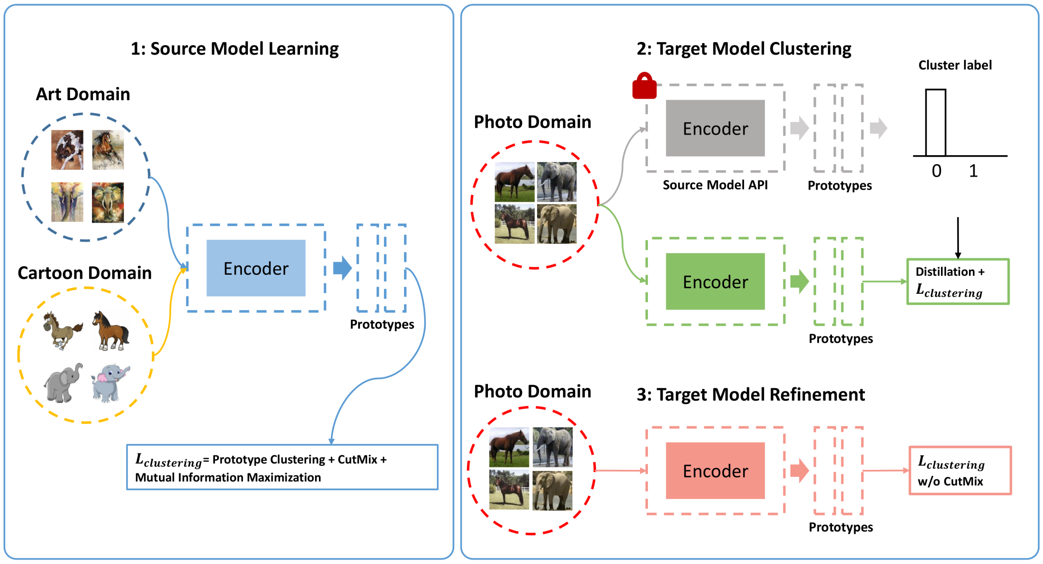

1: Source Model Learning
Art Domain
Cartoon Domain
Encoder
Prototypes
$L_{clustering}$= Prototype Clustering + CutMix + Mutual Information Maximization
2: Target Model Clustering
Photo Domain
Encoder
Source Model API
Prototypes
Cluster label
0
1
Encoder
Prototypes
Distillation + $L_{clustering}$
Photo Domain
3: Target Model Refinement
Encoder
Prototypes
$L_{clustering}$ w/o CutMix

Figure 1: The illustration of the proposed clustering framework under domain shift and privacy concerns. The semantic content of the source (Art and Cartoon) and target (Photo) data stays the same. However, the bias of the data in each domain leads to a distribution shift. During the adaptation phase, target users are only allowed to query from the source model, protecting the privacy of the source domain information.

approach by Menapace et al. (2020) can achieve this objective, it directly uses G_s to initialize G_t, compromising the privacy of the source domain and requiring G_s and G_t to have the same architecture. We now discuss how the different components of our method address these issues.

2.2 Source Model Learning

To effectively capture the feature distribution of the source data and avoid clustering based on domain information, we propose a clustering algorithm that consists of three components: prototype-oriented clustering, mutual-information maximization, and regularization via CutMix. The first two components help capture the feature distribution, while the last one curtails clustering based on domain information.

2.2.1 Prototype-oriented clustering

Our goal is to learn global representations of prototypes that capture the source data distributions and a feature encoder that maps the data from different domains to the prototypes. In our model, we have a linear clustering head, $C_{\boldsymbol{\mu}} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K] \in \mathbb{R}^{d_f \times K}$, where d_f denotes the dimension of both the prototype and the output of the feature encoder. The vector $\boldsymbol{\mu}_k$ represents a prototype of the kth cluster in the latent space. To discover the underlying clusters, we want to align the distribution of the global prototypes with the distribution of the feature representations in each domain.

We represent the distribution of the feature in each domain using the empirical distribution which is expressed as: $P_d = \sum_{j=1}^{n_d^s} \frac{1}{n_d^s} \delta_{\boldsymbol{f}_{dj}^s}$ where $\boldsymbol{f}_{dj}^s = F_{\boldsymbol{\theta}}^s(\boldsymbol{x}_{dj}^s)$ denotes the output of the feature encoder. While we use a set of global prototypes to learn domain-invariant representations, we carefully construct the distribution of prototypes in each domain such that the prototypes can align well with the data. Since the proportion of clusters in each domain may vary, we consider the domain-specific distribution of prototypes, Q_d, which is defined as: $Q_d = \sum_{k=1}^{K} \mathbf{B}_{dk} \delta_{\boldsymbol{\mu}_k}$, where $\mathbf{B}_{dk}$ denotes the proportion of cluster k in domain d ($\mathbf{B}_{dk} \geq 0$ and $\sum_{k=1}^{K} \mathbf{B}_{dk} = 1\ \forall d$). We emphasize here that the prototypes are shared across different domains, but the proportion of the prototypes is domain-specific.

To align the distributions of prototypes and data, we want to quantify their difference. A principled way to compare two discrete distributions is to consider the optimal transport problem (Kantorovich, 2006; Peyré and Cuturi, 2019). Thus, we consider the entropic regularized optimal transport formulation (Cuturi, 2013) that is defined as:

$$OT(P_d, Q_d) = \min_{\mathbf{T}_d \in \Pi(\boldsymbol{u}, \boldsymbol{v})} \mathrm{Tr}((\mathbf{T}_d)^T \mathbf{C}_d) + \epsilon h(\mathbf{T}_d), \qquad (1)$$

where $\mathbf{C}_d \in \mathbb{R}_{\geq 0}^{n_d^s \times K}$ stands for the transport cost matrix in domain d, Tr denotes the trace operation, $h(\mathbf{T}_d) = -\sum_{j,k} (\mathbf{T}_d)_{jk} \log(\mathbf{T}_d)_{jk}$ is the entropy of the transport plan, ϵ controls the strength of the regularization term, and $\mathbf{T}_d \in \mathbb{R}_{>0}^{n_d^s \times K}$ is a doubly stochastic matrix in domain d such that $\Pi(\boldsymbol{u}, \boldsymbol{v}) = \{\mathbf{T}_d | \mathbf{T}_d \mathbf{1} = \boldsymbol{u}, \mathbf{1}^T \mathbf{T}_d = \boldsymbol{v}\}$. The probability vectors $\boldsymbol{u} = \frac{\mathbf{1}}{n_d^s} \in \boldsymbol{\Sigma}^{n_d^s}$ and $\boldsymbol{v} = \mathbf{B}_d \in \boldsymbol{\Sigma}^K$, where $\boldsymbol{\Sigma}^M$ stands for the probability simplex of $\mathbb{R}^M$, denote the respective probabilities for P_d and Q_d. We define the point-wise transport cost $(\mathbf{C}_d)_{jk}$ as the cosine dissimilarity: $(\mathbf{C}_d)_{jk} = 1 - \frac{\boldsymbol{\mu}_k^T \boldsymbol{f}_{dj}^s}{||\boldsymbol{\mu}_k||\ ||\boldsymbol{f}_{dj}^s||}$, where $\boldsymbol{f}_{dj}^s = F_{\boldsymbol{\theta}}^s(\boldsymbol{x}_{dj}^s)$ denotes the output of the feature encoder. The intuition here is that if $(\mathbf{C}_d)_{jk}$ is high, it is less likely for sample j to be transported to cluster k.

To summarize, for a fixed $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$, we can solve Eq. (1) to obtain $\mathbf{T}_d$, the probabilities of moving prototypes to data points in each domain. After obtaining the transport plans, we update the parameters of the encoder $\boldsymbol{\theta}$ and prototypes $\boldsymbol{\mu}$ to minimize the total transport cost for the given transport plan using mini-batch stochastic gradient descent. The final transport loss is expressed as: $\mathcal{L}_{transport}\left(G^s; \mathcal{X}^s\right) = \frac{1}{D} \sum_{d=1}^{D} OT(P_d, Q_d)$. The connections of our method with other deep clustering algorithms (Caron et al., 2018; Asano et al., 2019) are provided in Appendix D.

2.2.2 Learning domain-specific cluster proportions

In the previous section, we utilize cluster proportions, $\mathbf{B}_d$, as the marginal constraint when solving the optimal transport problems. Assuming that each cluster contains roughly the same number of samples, we can use a uniform distribution for $\mathbf{B}_d$. However, this assumption is not valid in practice. Since each domain may have different distributions over the clusters, we propose a way to estimate domain-specific cluster proportions, $\mathbf{B}_d$. To infer these quantities, we first initialize them with a uniform prior over clusters $\mathbf{B}_{dk} = \frac{1}{K}$ and iteratively refine them using an EM-like update (Saerens et al., 2002; Kang et al., 2018; Alexandari et al., 2020):

$$\tilde{\mathbf{B}}_{dk}^{l+1} = \tfrac{1}{M_d} \textstyle\sum_{j=1}^{M_d} \pi_{\boldsymbol{\theta}}^l(\boldsymbol{\mu}_k \mid \boldsymbol{f}_{dj}^s), \quad \text{where} \quad \pi_{\boldsymbol{\theta}}^l(\boldsymbol{\mu}_k \mid \boldsymbol{f}_{dj}^s) = \frac{\exp(\boldsymbol{\mu}_k^T \boldsymbol{f}_{dj}^s) \mathbf{B}_{dk}^l}{\sum_{k'=1}^{K} \exp(\boldsymbol{\mu}_{k'}^T \boldsymbol{f}_{dj}^s) \mathbf{B}_{dk'}^l}, \tag{2}$$

where M_d stands for the number of samples in domain d in a mini-batch, $\mathbf{B}_{dk}^{l+1}$ refers to the proportion of cluster k in domain d at the $l{+}1$ th iteration, $\pi_{\boldsymbol{\theta}}^l(\boldsymbol{\mu}_k \mid \boldsymbol{f}_{dj}^s)$ denotes the predicted cluster probabilities at the l th iteration, and $\boldsymbol{f}_{dj}^s$ indicates the j th feature sample in domain d. To obtain a reliable estimates of the full dataset, we iteratively update the proportions with $\mathbf{B}_{dk}^{l+1} \leftarrow \beta^l \mathbf{B}_{dk}^l + (1 - \beta^l) \tilde{\mathbf{B}}_{dk}^{l+1}$, where β^l follows a cosine learning rate schedule.

2.2.3 Global alignment with mutual-information maximization

The transport loss introduced in the previous section aligns the local distributions of data and prototypes. To assist the learning of the feature encoder and prototypes on a global level, we utilize the widely-adopted mutual-information objective (Krause et al., 2010; Shi and Sha, 2012). This objective ensures that the feature representations are tightly clustered around each prototype. If the data are close to the prototypes, we expect the posterior probabilities to be close to one-hot vectors. To make this more likely, we minimize the entropy of the conditional distribution of cluster labels given the data. However, minimizing this loss alone could lead to a degenerate solution since the model can assign all the samples to one cluster (Morerio et al., 2017; Wu et al., 2020). To prevent such a solution, we maximize the marginal entropy of the cluster label distribution. The mutual-information objective is thus expressed as:

$$\begin{aligned} \mathcal{L}_{mi}\left(G^s; \mathcal{X}^s\right) &= -[H\left(\mathcal{Y}^s\right) - H\left(\mathcal{Y}^s \mid \mathcal{X}^s\right)] \\ &= -[h\left(\mathbb{E}_{\boldsymbol{x}^s \in \mathcal{X}^s} G^s\left(\boldsymbol{x}^s\right)\right) - \mathbb{E}_{\boldsymbol{x}^s \in \mathcal{X}^s} h\left(G^s\left(\boldsymbol{x}^s\right)\right)], \end{aligned} \tag{3}$$

where $H\left(\mathcal{Y}^s\right)$ and $H\left(\mathcal{Y}^s \mid \mathcal{X}^s\right)$ denote the marginal entropy and conditional entropy of the cluster labels $\mathcal{Y}^s$, which are latent variables, respectively and $h(p) = -\sum_i p_i \log p_i$.

To avoid clustering based on domain information, we add the CutMix (Yun et al., 2019) regularization, which mixes two samples by interpolating images and labels. Since the data have no labels, the predicted cluster probabilities are utilized as the pseudo-labels. The CutMix regularization is defined as: $\mathcal{L}_{cutmix} = \mathbb{E}_{\boldsymbol{x}_i^s, \boldsymbol{x}_j^s \in \mathcal{X}^s} L(G^s(\tilde{x}), \tilde{y})$, where $L(\cdot, \cdot)$ is the cross-entropy loss and $(\tilde{\boldsymbol{x}}, \tilde{y})$ are the interpolated samples from the pair $(\boldsymbol{x}_i^s, G_*^s(\boldsymbol{x}_i^s))$ and $(\boldsymbol{x}_j^s, G_*^s(\boldsymbol{x}_j^s))$, with G_*^s indicating no gradient optimization. We construct the final objective function to update the prototypes and feature encoder.

$$\mathcal{L}_{clustering}(G^s; \mathcal{X}^s) = \mathcal{L}_{transport}(G^s; \mathcal{X}^s) + \mathcal{L}_{mi}(G^s; \mathcal{X}^s) + \mathcal{L}_{cutmix}(G^s; \mathcal{X}^s). \tag{4}$$

2.3 Target model learning

Because of the domain shift, we divide our target model learning into two stages—target model clustering and target model refinement—to ensure that the knowledge transferred from the source domain does not interfere with the learning in the target domain (Shu et al., 2018). The first phase aims to transfer the knowledge from the source model to the target model while protecting the privacy of the source domain. The second phase focuses on refining the target model so that target samples are tightly clustered around each prototype.

2.3.1 Target model clustering

In many practical applications, it is crucial to preserve the privacy of both the source model and data (Ziller et al., 2020; 2021). Thus, directly using the source model to initialize the target model is not ideal. Instead, we consider the practical problem where the source model can only provide a cluster label for each target example. The source model is simply an API, and we have access to neither its architecture nor model parameters. With the predicted cluster assignments given by the source model, we want to learn a well-trained clustering model on the target data.

Source knowledge transfer with knowledge distillation. Given unlabeled target samples, $\{\boldsymbol{x}_i^t\}_{i=1}^{n_t}$, we can obtain cluster assignments, $G^s(\boldsymbol{x}_i^t)$, through the source model. Our algorithm can work for both hard and soft labels; however, it is more practical to consider hard labels from the source domain since soft labels may not be available for all models (Sanyal et al., 2022). Thus, we consider hard label assignments from the source domain in our experiments. To transfer the knowledge from the source to target models, we utilize a knowledge distillation loss (Hinton et al., 2015) to train the target model to mimic the predicted output from the source. The loss can be formulated as follows: $\mathcal{L}_{kd}\left(G^t; \mathcal{X}^t, G^s\right) = \mathbb{E}_{x^t \in \mathcal{X}^t} \mathcal{D}_{kl}\left(G^s(\boldsymbol{x}^t) \| G^t(\boldsymbol{x}^t)\right)$, where $\mathcal{D}_{kl}(G^s(\boldsymbol{x}^t) \| G^t(\boldsymbol{x}^t)) = \sum_{k=1}^{K} G^s(\boldsymbol{x}^t)_k \log \frac{G^s(\boldsymbol{x}^t)_k}{G^t(\boldsymbol{x}^t)_k}$ stands for the Kullback–Leibler divergence between two distributions and G^t is initialized with a pre-trained feature encoder.

Because of the domain shift, the source model may not always cluster target samples based on their semantic content. Thus, we propose to refine the predicted target assignments using two simple strategies: label smoothing (Pereyra et al., 2017) and self-ensemble (Laine and Aila, 2016; Kim et al., 2021). Müller et al. (2019) discover that label smoothing can help the penultimate layer representation form tight clusters, allowing the model to discover underlying clusters more easily. To utilize label smoothing, we interpolate the hard assignments with a uniform distribution to obtain soft labels: $\hat{y}_k^{LS} = (1-\gamma) G^s(\boldsymbol{x}^t)_k + \frac{\gamma}{K}$, where γ is the weight of the uniform distribution. As the target model improves, we can leverage its predicted cluster probabilities across different iterations to form a temporal ensemble: $(\hat{y}^t)^l \leftarrow \tau (\hat{y}^t)^{l-1} + (1-\tau) G^t(\boldsymbol{x}^t)^l$, where τ determines how much weight we give to past assignments, $(\hat{y}_t)^{l-1}$ is the assignment at the $l-1$ th iteration, and $G^t(\boldsymbol{x}^t)^l$ is the current assignment. We initialize $(\hat{y}^t)^0$ with the smooth assignments from the source model. The refined cluster assignments from the source model $\hat{y}^t$ then replaces $G^s(x^t)$ in the distillation loss. Thus, for target model clustering, the training includes the following losses: $\mathcal{L}_{target\_clustering}(G^t; \mathcal{X}^t, G^s) = \mathbb{E}_{x_t \in \mathcal{X}^t} \mathcal{D}_{kl}\left(\hat{y}^t \| \, G^t\left(\boldsymbol{x}^t\right)\right) + \mathcal{L}_{clustering}(G^t; \mathcal{X}^t)$.

2.3.2 Target model refinement

In the previous section, we use both source and target domain knowledge to learn our clustering model. While the source domain knowledge can assist target domain learning, the bias in distribution due to domain shift could lead the target model to learn noisy domain information from the source model. Similar to the observation by Shu et al. (2018), we find that the target model could benefit from further clustering on the target data alone. We utilize the clustering objective in Eq. (4) with target data and model as arguments and without the CutMix loss. The CutMix regularization term is not included since there is no source knowledge transfer and the target data come from a single domain. Also, the regularizer makes the predicted probabilities unconfident. During this stage, we want the target feature representations to be clustered tightly around the target prototypes (confident network outputs). The target refinement loss is thus formulated as: $\mathcal{L}_{target\_refinement}(G^t; \mathcal{X}^t) = \mathcal{L}_{transport}(G^t; \mathcal{X}^t) + \mathcal{L}_{mi}(G^t; \mathcal{X}^t)$.

3 Related work

Clustering. For a complete picture of the field, readers may refer to the survey by Min et al. (2018). We emphasize deep-clustering-based approaches, which attempt to learn the feature representation of the data while simultaneously discovering the underlying clusters: K-means (Yang et al., 2017; Caron et al., 2018), information maximization (Menapace et al., 2020; Ji et al., 2019; Kim and Ha, 2021; Do et al., 2021), transport alignment(Asano et al., 2019; Caron et al., 2020; Wang et al., 2022), neighborhood-clustering (Xie et al., 2016; Huang et al., 2019; Dang et al., 2021), contrastive learning (Pan and Kang, 2021; Shen et al., 2021), probabilistic approaches (Yang et al., 2020; Monnier et al., 2020; Falck et al., 2021; Manduchi et al., 2021), and kernel density (Yang and Li, 2021). These works primarily focus on clustering data for downstream tasks for a single domain, whereas our clustering algorithm is designed to cluster the data from multiple domains. Moreover, our method solves the problem of transferring the knowledge from the data-rich source domain to the target domain. Distinct from ACIDS (Menapace et al., 2020) which maximizes the mutual information between different views of the same image, our method maximizes the mutual information between cluster labels and images. In addition to data privacy, we also consider model privacy.

Source-free knowledge transfer. Early domain adaptation methods (Ben-David et al., 2006; Blitzer et al., 2006; Tzeng et al., 2014; Ganin and Lempitsky, 2015; Long et al., 2017; 2018; 2015; Tzeng et al., 2017; Courty et al., 2017) focus on reducing the distributional discrepancies between the source and target domain data. These methods, however, require access to the source and target data simultaneously during the adaptation process, compromising the privacy of the source domain. To overcome this issue, several methods (Kuzborskij and Orabona, 2013; Du et al., 2017; Liang et al., 2020; Li et al., 2020; Kundu et al., 2020; Kurmi et al., 2021; Yeh et al., 2021; Tanwisuth et al., 2021) have been developed for source data-free domain adaptation. For a more thorough literature review of this field, we refer the reader to the survey paper by Yang et al. (2021). In contrast to those methods, we consider a more challenging adaptation setting, as used in previous works (Lipton et al., 2018; Deng et al., 2021; Liang et al., 2021; Zhang et al., 2021), where the privacy of both data and models is the main concern. Different from these lines of work, our approach relies on labeled data in neither the source nor target domain.

4 Experiments

In this section, we evaluate our method on Office-31, Office-Home, and PACS datasets under three different transfer learning scenarios. The first setting (standard setting) includes only input distribution shift. The second setup (model transfer setting) contains both input and model shifts. The last scenario (limited-data and cluster-imbalanced setting) involves both input and cluster-proportion shifts.

4.1 Experimental setup

Comparable methods. We benchmark against existing clustering approaches—*DeepCluster of Caron et al. (2018), Invariant Information Clustering (IIC) of Ji et al. (2019)*, and *Adaptive Clustering of Images under Domain Shift (ACIDS) of Menapace et al. (2020)*—in the UCDS setting when the results are available. Unless specified otherwise, the reported baseline results are directly taken from Menapace et al. (2020). IIC and DeepCluster train on target data only while ACIDS trains a source model and then adapts on the target data. We also compare our approach to the following alternative methods, which are different components of our framework: *Pre-trained Only (PO)*, which uses a pre-trained network to cluster target data directly; *Source training Only (SO)*, which trains a model on all the source data using Eq. (4) and directly tests on the target data; *Target Training Only (TO)*, which trains a model on the target data using the loss in Section 2.3.2 without source knowledge transfer; *Adaptation Only (AO)*, which performs the first two stages of our framework, source model training and target model clustering, without further refining on the target data; *PCD (Ours)* refers to using all three stages of our approach: source model learning, target model clustering, and target model refinement. SO allows us to see the significance of the source model training. Compared with PCD, TO enables us to evaluate the importance of the source knowledge transfer, while AO helps us see the improvement from target refinement.

Pre-trained networks. To verify the compatibility of our approach with different models, we consider multiple types of pre-trained network architectures and pre-training schemes in our experiments. For pre-training schemes, we explore supervised and self-supervised pre-trainings on ImageNet (Russakovsky et al., 2015). For network architectures, we experiment with supervised ResNet-18 as well as self-supervised ResNet-50 (He et al., 2016) and Vision Transformer (ViT) (Dosovitskiy et al., 2020). In particular, we adopt the network trained by SWAV (Caron et al., 2020) for ResNet-50 and that trained by DINO (Caron et al., 2021) for Vision Transformer for our self-supervised pre-training.

Datasets and evaluation metric. We use the following datasets in our experiments: Office-31 (Saenko et al., 2010), Office-Home (Venkateswara et al., 2017), and PACS (Li et al., 2017). The Office-31 dataset has three domains (Amazon, Webcam, DSLR) with 4,652 images. The Office-Home dataset consists of 15,500 images with four domains (Art, Clipart, Product, and Real-world). The PACS dataset contains four domains (Art, Painting, Cartoon, and Sketch) with 9,991 images. Following prior works (Ji et al., 2019; Menapace et al., 2020), we evaluate all methods using clustering accuracy on the target dataset. The metric is calculated by first solving a linear assignment problem to match the clusters to ground-truth classes. We set K, the number of clusters, equal to the number of classes in each dataset for evaluation purposes.

Implementation details. We follow the standard protocols for source-free domain adaptation (Liang et al., 2020). Specifically, we use mini-batch SGD with a momentum of 0.9 and weight decay of 0.001. Both source and target encoders are initialized with ImageNet pre-trained networks (Russakovsky et al., 2015), but the prototypes and the projection layer of the encoder are initialized with a random linear layer. The initial learning rates are set to 0.001 for the pre-trained encoders and 0.01 for the randomly initialized layer. The learning rates, η, follows the following schedule: $\eta = \eta_0(1 + 10p)^{-0.75}$ where η_0 is the initial learning rate. We use the batch size of 64 in both source and target learning. All three loss terms are equally weighted, while other choices are possible. We report the sensitivity of the coefficients in front of the loss terms in Appendix B. The initial value of β_0 to learn domain-specific proportions is set to 0.9999 for source clustering and 0.99 for target clustering in all settings. We run our method with three different random seeds to calculate the standard deviation. Full implementation details are included in Appendix F

Table 2: Clustering accuracy $(\%)$ on different datasets for ResNet-18-based methods.

Settings	Office-31				Office-Home					PACS				
	$\mathcal{R} \to$ A	$\mathcal{R} \to$ W	$\mathcal{R} \to$ D	Avg	$\mathcal{R} \to$ Ar	$\mathcal{R} \to$ Cl	$\mathcal{R} \to$ Pr	$\mathcal{R} \to$ Rw	Avg	$\mathcal{R} \to$ P	$\mathcal{R} \to$ A	$\mathcal{R} \to$ C	$\mathcal{R} \to$ S	Avg
DeepCluster (Caron et al., 2018)	19.6	18.9	18.7	19.1	8.9	11.1	16.9	13.3	12.6	27.9	22.2	24.4	27.1	25.4
IIC (Ji et al., 2019)	31.9	37.0	34.0	34.4	12.0	15.2	22.5	15.9	16.4	70.6	39.8	39.6	46.6	49.2
ACIDS (Menapace et al., 2020)	33.4	37.5	36.1	35.7	12.0	16.2	23.9	15.7	17.0	64.4	42.1	44.5	51.1	50.5
PO	14.1	17.9	18.3	16.8	11.4	9.0	12.9	10.8	11.0	30.5	24.1	19.8	20.8	23.8
SO	34.5	46.7	43.0	41.4	23.6	15.6	23.1	21.8	21.0	30.8	35.7	27.6	26.0	30.0
TO	38.0	46.6	45.3	43.3	21.3	12.2	30.6	24.2	22.1	88.4	**56.5**	56.5	49.1	62.6
AO	42.8	58.4	55.8	52.3	30.0	22.7	29.3	24.4	26.6	91.5	47.7	52.3	49.1	60.2
PCD	**46.8**	**60.0**	**57.8**	**54.9**	**33.3**	**24.4**	**31.4**	**28.1**	**29.3**	**92.6**	49.7	**56.7**	**53.4**	**63.4**

4.2 MAIN RESULTS

Standard setting. In real-world applications, the source and target data distributions often differ. To test our method under input distribution shift, we evaluate our method on Office-31, Office-Home, and PACS datasets. For each experiment, we select one domain as the target and all the other, denoted as $\mathcal{R}$, as the source domains. We use the same model architecture in both the source and target domains. We report the results for ResNet-18 (supervised pre-training) in Table 2. The full results with standard error are shown in Appendix A. Compared with the results reported by Menapace et al. (2020), our algorithm outperforms ACIDS consistently in all three datasets (see Table 2): 19.2% on Office-31, 12.3% on Office-Home, and 12.9% on PACS. Though ACIDS does not address the problem of our setting with the same pre-training scheme and backbones as our method, we report the results for comparison. The results of ACIDS with this pre-training scheme are included in Appendix A in Table 7. We observe that our approach still outperforms ACIDS on three out of four tasks with 4% higher in the average accuracy, emphasizing the general applicability and strong performance of PCD. With no adaptation, TO achieves higher clustering accuracy than both IIC and DeepCluster, demonstrating the effectiveness of our clustering method.

Compared with our own alternative methods (*i.e.*, PO, SO, TO, and AO), PCD achieves steady gains in performance except for one task. Notably, on the task $\mathcal{R} \to$ A of the PACS dataset, we notice a negative transfer (Wang et al., 2019) as TO performs the best (56.5% vs. 49.7%). We hypothesize that the Art domain looks quite distinct from the source domain data, and the supervised-pretraining backbone is strong enough to yield good performance using target training only. SO improves upon PO on all the tasks, showing that the knowledge from the source domain can benefit the target domain learning. Likewise, we see consistent improvements around $2-3\%$ over AO . This result illustrates the importance of target model refinement. We observe similar patterns using self-supervised ResNet-50 as the backbone (see Appendix A).

Model transfer setting. In many applications, source and target data owners may have different resource requirements. As an example, unlike source providers such as Google, target clients may

Table 3: Clustering accuracy $(\%)$ on Office-31 for different model transfer settings. *ssl* and *sup* denote self-supervised and supervised pre-trainings, respectively. (source) → (target).

Settings	ViT-B/16 (ssl) → ResNet-50 (ssl)				ViT-B/16 (ssl) → ResNet-50 (sup)				ViT-B/16 (ssl) → ResNet-18 (sup)			
	$\mathcal{R} \to$ A	$\mathcal{R} \to$ W	$\mathcal{R} \to$ D	Avg	$\mathcal{R} \to$ A	$\mathcal{R} \to$ W	$\mathcal{R} \to$ D	Avg	$\mathcal{R} \to$ A	$\mathcal{R} \to$ W	$\mathcal{R} \to$ D	Avg
PO	20.1/13.5	26.7/16.7	27.2/19.3	24.7/16.5	20.1/15.7	26.7/24.2	27.2/18.8	24.7/19.6	20.1/14.1	26.7/17.9	27.2/18.3	24.7/16.8
SO	43.2	46.4	37.2	42.3	43.2	46.4	37.2	42.3	43.2	46.4	37.2	42.3
TO	32.6	34.3	33.7	33.5	43.7	55.8	**52.0**	50.5	38.0	45.3	46.6	43.3
AO	50.6	49.7	36.4	45.6	52.5	53.7	44.2	50.1	53.0	47.2	43.9	48.0
PCD	**51.7**	**51.7**	**41.8**	**48.4**	**54.4**	**60.8**	49.2	**54.8**	**54.6**	**53.6**	**46.7**	**51.6**

have limited resources. Thus, they may not be able to use the same model architecture as the source provider. To illustrate the flexibility and demonstrate the generalizability of our framework under model shift, we experiment with different model architectures and pre-training schemes in the source and target domains. We explore three different combinations of source and target model architectures and pre-training schemes: ViT-B/16 (self-supervised) → ResNet-50 (self-supervised), ViT-B/16 (self-supervised) → ResNet-50 (supervised), and ViT-B/16 (self-supervised) → ResNet-18 (supervised). The results are reported in Table 3. In both settings, we continually see improvements in average performance. This finding shows that our method still performs well even though the source and target domain architectures differ, providing strong evidence for the generalizability and compatibility of different components of our framework.

Limited-data and cluster-imbalanced setting. In real-world scenarios, target domain data are often scarce and imbalanced. To further show the benefit of our clustering loss under this setting, we follow the experimental procedures in Tachet des Combes et al. (2020). Specifically, we drop 70% of the target data in the first $\lfloor K/2 \rfloor$ clusters to create this scenario. The experiments are done on the Office-31 dataset. To illustrate the use of our method in a label-free pipeline, we utilize self-supervised ResNet-50 as the feature encoder for both source and target domains. This scenario is extremely challenging for transfer learning methods since there are shifts in both image and cluster-label distributions. However, as we see in Table 4, PCD still outperforms TO by 4%. We note that TO also adaptively learns the target proportions but does not have to deal with distribution shifts. We also observe consistent improvements over other alternative methods. This result highlights the use of our method in practical settings with limited and imbalanced data.

Table 4: Clustering accuracy $(\%)$ on sub-sampled version of Office-31 for ResNet-50-based methods.

Settings	$\mathcal{R} \to$ sub-A	$\mathcal{R} \to$ sub-W	$\mathcal{R} \to$ sub-D	Avg
PO	14.6	16.7	21.5	17.6
SO	21.1	32.5	36.0	29.9
TO	31.4	41.9	45.1	39.5
AO	34.7	40.8	43.9	39.8
PCD	**37.8**	**46.4**	**47.0**	**43.7**

5 ANALYSIS

Ablation study. To see the contribution of each component, we remove one part at a time from the whole framework and present the results in Table 5. Overall, PCD achieves higher clustering accuracy than all other alternative versions with privacy constraints. We observe that the clustering accuracy drops dramatically (10.3%) without the prototype clustering, illustrating the importance of this element. The mutual-information objective is also significant since omitting it leads to a drop in clustering accuracy of 7.3%. This observation shows that the two losses are complementary to each other. The temporal ensemble of the cluster labels produced by the source model still improves the model but does not significantly hurt the performance if removed. We also report the result of directly initializing the target model with the source model (w/o model privacy). We notice around 3% improvement. When pooling all the source domains together into a single source domain for clustering (pooled source), we see a drop of 2.2%. This result indicates that we should respect the local structures of data in each domain.

Table 5: Clustering accuracy (%) on the task $\mathcal{R} \to$ W (Office-31) under different variants (ResNet-18).

Full	w/o prototype clustering	w/o MI	w/o CutMix	w/o Temporal Ensemble	w/o model privacy	pooled source
60.0	49.7	52.7	54.5	58.3	63.1	57.8

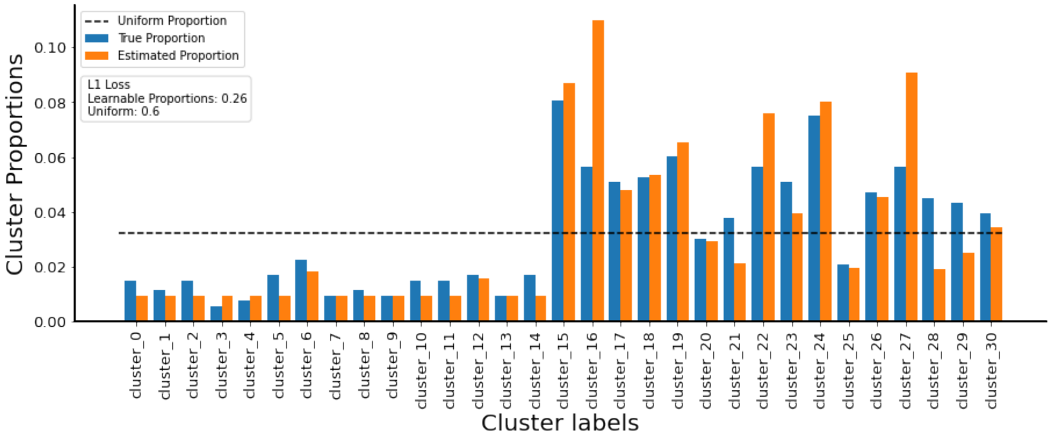
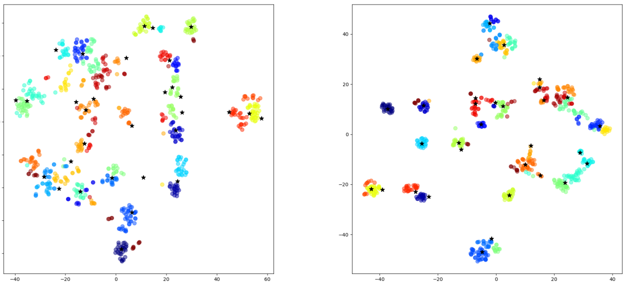


Figure 2: Visualization of the cluster proportions for the sub-sampled version of the task $\mathcal{R} \rightarrow$ sub-W on the Office-31 dataset. To create this plot, we first match the predicted clusters with the true labels using optimal assignment. The blue bars exhibit the true cluster proportions, whereas the orange bars depict the estimated cluster proportions. The L1 loss of the estimated cluster distribution is lower than that of the uniform proportion (0.26 vs. 0.6), demonstrating the success of our estimation.



(a) (b)

Figure 3: t-SNE visualizations of the encoder's outputs on the task $\mathcal{R} \rightarrow$ W. Different colors represent semantic classes from the ground-truth labels. Figure (a) shows the outputs trained with Target training Only (TO), while Figure (b) depicts those trained with the whole framework. Samples with similar semantic content are more tightly clustered around the prototypes ($\star$) in Figure (b).

Results analysis. *Visualization.* In Figure 2, we visualize the estimated target proportions versus the true proportions, which are calculated from the ground-truth labels. The learned cluster distribution achieves lower L1 loss than the uniform distribution, meaning that the estimated values reflect the data distribution better than the uniform proportions. We plot the t-SNE visualization of the outputs of the feature encoder for the model trained with target training only (Section 2.3.2) in Figure 3a and the one trained with the whole framework (Algorithm 1) in Figure 3b. Using the whole framework, we can see that the samples are more tightly clustered around the prototypes, illustrating that the knowledge from the source domain benefits the target model learning. *Running time and parameter size.* We report the number of parameters and running time per step for comparison in Appendix C, where we see that our method is more efficient in both time and memory than ACIDS.

6 Conclusion

We study a practical transfer learning setting that does not rely on labels in the source and target domains and considers the privacy of both the source data and model. To solve this problem, we provide a novel solution that utilizes prototype clustering, mutual-information maximization, data augmentation, and knowledge distillation. Experiments show that our clustering approach consistently outperforms the baselines and works well across different datasets and model architectures.

REFERENCES

Amina Adadi. A survey on data-efficient algorithms in big data era. *Journal of Big Data*, 8(1):1–54, 2021.

John A Hartigan. Direct clustering of a data matrix. *Journal of the american statistical association*, 67(337):123–129, 1972.

Guy Barrett Coleman and Harry C Andrews. Image segmentation by clustering. *Proceedings of the IEEE*, 67(5):773–785, 1979.

Tao Lei, Xiaohong Jia, Yanning Zhang, Shigang Liu, Hongying Meng, and Asoke K Nandi. Superpixel-based fast fuzzy c-means clustering for color image segmentation. *IEEE Transactions on Fuzzy Systems*, 27(9):1753–1766, 2018.

Himanshu Mittal, Avinash Chandra Pandey, Mukesh Saraswat, Sumit Kumar, Raju Pal, and Garv Modwel. A comprehensive survey of image segmentation: clustering methods, performance parameters, and benchmark datasets. *Multimedia Tools and Applications*, pages 1–26, 2021.

Chris Biemann. Chinese whispers-an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing*, pages 73–80, 2006.

Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. A compare-aggregate model with latent clustering for answer selection. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2093–2096, 2019.

Shie Mannor, Ishai Menache, Amit Hoze, and Uri Klein. Dynamic abstraction in reinforcement learning via clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 71, 2004.

Xin Xu, Zhenhua Huang, Daniel Graves, and Witold Pedrycz. A clustering-based graph laplacian framework for value function approximation in reinforcement learning. *IEEE Transactions on Cybernetics*, 44(12):2613–2625, 2014.

Mohsen Ahmadi, Ali Taghavirashidizadeh, Danial Javaheri, Armin Masoumian, Saeid Jafarzadeh Ghoushchi, and Yaghoub Pourasad. Dqre-scnet: a novel hybrid approach for selecting users in federated learning with deep-q-reinforcement learning based on spectral clustering. *Journal of King Saud University-Computer and Information Sciences*, 2021.

Di Hu, Feiping Nie, and Xuelong Li. Deep multimodal clustering for unsupervised audiovisual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9248–9257, 2019.

Brian Chen, Andrew Rouditchenko, Kevin Duarte, Hilde Kuehne, Samuel Thomas, Angie Boggust, Rameswar Panda, Brian Kingsbury, Rogerio Feris, David Harwath, et al. Multimodal clustering networks for self-supervised learning from unlabeled videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8012–8021, 2021.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

Nooshin Mojab, Vahid Noroozi, Darvin Yi, Manoj P Nallabothula, Abdullah Aleem, S Yu Philip, and Joelle A Hallak. Real-world multi-domain data applications for generalizations to clinical settings. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 677–684. IEEE, 2020.

Micah J Sheller, Brandon Edwards, G Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, Daniel Marcus, Rivka R Colen, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, 10(1):1–12, 2020.

Ahmad S Salehi, Carsten Rudolph, and Marthie Grobler. A dynamic cross-domain access control model for collaborative healthcare application. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 643–648. IEEE, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Ali Farhadi and Mostafa Kamali Tabrizi. Learning to recognize activities from the wrong view point. In *European conference on computer vision*, pages 154–166. Springer, 2008.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

Willi Menapace, Stéphane Lathuilière, and Elisa Ricci. Learning to cluster under domain shift. In *European Conference on Computer Vision*, pages 736–752. Springer, 2020.

Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874, 2019.

Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3514–3522, 2019.

Liangchen Luo, Mark Sandler, Zi Lin, Andrey Zhmoginov, and Andrew Howard. Large-scale generative data-free distillation. *arXiv preprint arXiv:2012.05578*, 2020.

Leonid V Kantorovich. On the translocation of masses. *Journal of Mathematical Sciences*, 133(4): 1381–1382, 2006.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

Andreas Krause, Pietro Perona, and Ryan Gomes. Discriminative clustering by regularized information maximization. *Advances in neural information processing systems*, 23, 2010.

Yuan Shi and Fei Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. *arXiv preprint arXiv:1206.6438*, 2012.

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.

Jonathan Soifer, Jason Li, Mingqin Li, Jeffrey Zhu, Yingnan Li, Yuxiong He, Elton Zheng, Adi Oltean, Maya Mosyak, Chris Barnes, et al. Deep learning inference service at microsoft. In *2019 USENIX Conference on Operational Machine Learning (OpML 19)*, pages 15–17, 2019.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41, 2002.

Guoliang Kang, Liang Zheng, Yan Yan, and Yi Yang. Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 401–416, 2018.

Amr Alexandari, Anshul Kundaje, and Avanti Shrikumar. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *International Conference on Machine Learning*, pages 222–232. PMLR, 2020.

Pietro Morerio, Jacopo Cavazza, and Vittorio Murino. Minimal-entropy correlation alignment for unsupervised deep domain adaptation. *arXiv preprint arXiv:1711.10288*, 2017.

Xiaofu Wu, Quan Zhou, Zhen Yang, Chunming Zhao, Longin Jan Latecki, et al. Entropy minimization vs. diversity maximization for domain adaptation. *arXiv preprint arXiv:2002.01690*, 2020.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

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

Alexander Ziller, Jonathan Passerat-Palmbach, Théo Ryffel, Dmitrii Usynin, Andrew Trask, Ionésio Da Lima Costa Junior, Jason Mancuso, Marcus Makowski, Daniel Rueckert, Rickmer Braren, et al. Privacy-preserving medical image analysis. *arXiv preprint arXiv:2012.06354*, 2020.

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.

Sunandini Sanyal, Sravanti Addepalli, and R Venkatesh Babu. Towards data-free model stealing in a hard label setting. *arXiv preprint arXiv:2204.11022*, 2022.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Kyungyul Kim, ByeongMoon Ji, Doyoung Yoon, and Sangheum Hwang. Self-knowledge distillation with progressive refinement of targets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6567–6576, 2021.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

Erxue Min, Xifeng Guo, Qiang Liu, Gen Zhang, Jianjing Cui, and Jun Long. A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, 6:39501–39514, 2018.

Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870. PMLR, 2017.

Yunji Kim and Jung-Woo Ha. Contrastive fine-grained class clustering via generative adversarial networks. *arXiv preprint arXiv:2112.14971*, 2021.

Kien Do, Truyen Tran, and Svetha Venkatesh. Clustering by maximizing mutual information across views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9928–9938, 2021.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

Dongsheng Wang, Dandan Guo, He Zhao, Huangjie Zheng, Korawat Tanwisuth, Bo Chen, and Mingyuan Zhou. Representing mixtures of word embeddings with mixtures of topic embeddings. *arXiv preprint arXiv:2203.01570*, 2022.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *International Conference on Machine Learning*, pages 2849–2858. PMLR, 2019.

Zhiyuan Dang, Cheng Deng, Xu Yang, Kun Wei, and Heng Huang. Nearest neighbor matching for deep clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13693–13702, 2021.

Erlin Pan and Zhao Kang. Multi-view contrastive graph clustering. *Advances in Neural Information Processing Systems*, 34, 2021.

Yuming Shen, Ziyi Shen, Menghan Wang, Jie Qin, Philip Torr, and Ling Shao. You never cluster alone. *Advances in Neural Information Processing Systems*, 34, 2021.

Xu Yang, Cheng Deng, Kun Wei, Junchi Yan, and Wei Liu. Adversarial learning for robust deep clustering. *Advances in Neural Information Processing Systems*, 33:9098–9108, 2020.

Tom Monnier, Thibault Groueix, and Mathieu Aubry. Deep transformation-invariant clustering. *Advances in Neural Information Processing Systems*, 33:7945–7955, 2020.

Fabian Falck, Haoting Zhang, Matthew Willetts, George Nicholson, Christopher Yau, and Chris C Holmes. Multi-facet clustering variational autoencoders. *Advances in Neural Information Processing Systems*, 34, 2021.

Laura Manduchi, Ričards Marcinkevičs, Michela C Massi, Thomas Weikert, Alexander Sauter, Verena Gotta, Timothy Müller, Flavio Vasella, Marian C Neidert, Marc Pfister, et al. A deep variational approach to clustering survival data. *arXiv preprint arXiv:2106.05763*, 2021.

Yingzhen Yang and Ping Li. Discriminative similarity for data clustering. *arXiv preprint arXiv:2109.08675*, 2021.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128, 2006.

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

Yaroslav Ganin and Victor S. Lempitsky. Unsupervised domain adaptation by backpropagation. In Francis R. Bach and David M. Blei, editors, *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1180–1189. JMLR.org, 2015.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2208–2217. PMLR, 06–11 Aug 2017.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 97–105, 2015.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Ilja Kuzborskij and Francesco Orabona. Stability and hypothesis transfer learning. In *International Conference on Machine Learning*, pages 942–950. PMLR, 2013.

Simon S Du, Jayanth Koushik, Aarti Singh, and Barnabás Póczos. Hypothesis transfer learning via transformation functions. *Advances in neural information processing systems*, 30, 2017.

Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020.

Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553, 2020.

Vinod K Kurmi, Venkatesh K Subramanian, and Vinay P Namboodiri. Domain impression: A source data free domain adaptation method. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 615–625, 2021.

Hao-Wei Yeh, Baoyao Yang, Pong C Yuen, and Tatsuya Harada. Sofa: Source-data-free feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 474–483, 2021.

Korawat Tanwisuth, Xinjie Fan, Huangjie Zheng, Shujian Zhang, Hao Zhang, Bo Chen, and Mingyuan Zhou. A prototype-oriented framework for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 34, 2021.

Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8978–8987, 2021.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pages 3122–3130. PMLR, 2018.

Bin Deng, Yabin Zhang, Hui Tang, Changxing Ding, and Kui Jia. On universal black-box domain adaptation. *arXiv preprint arXiv:2104.04665*, 2021.

Jian Liang, Dapeng Hu, Ran He, and Jiashi Feng. Distill and fine-tune: Effective adaptation from a black-box source model. *arXiv preprint arXiv:2104.01539*, 2021.

Haojian Zhang, Yabin Zhang, Kui Jia, and Lei Zhang. Unsupervised domain adaptation of black-box source models. *arXiv preprint arXiv:2101.02839*, 2021.

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.

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.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

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.

Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11293–11302, 2019.

Remi Tachet des Combes, Han Zhao, Yu-Xiang Wang, and Geoff Gordon. Domain adaptation with conditional distribution matching and generalized label shift. In *Advances in Neural Information Processing Systems*, 2020.

A Prototype-oriented Clustering for Domain Shift with Source Privacy: Appendix

A Full experimental results

A.1 Standard setting

Table 6: Clustering accuracy (%) on different datasets for ResNet-18-based methods (supervised pre-training for all methods below the mid line) and (random initialization for all methods above the mid line).

Settings	Office-31				Office-Home					PACS				
	$\mathcal{R} \to$ A	$\mathcal{R} \to$ W	$\mathcal{R} \to$ D	Avg	$\mathcal{R} \to$ Ar	$\mathcal{R} \to$ Cl	$\mathcal{R} \to$ Pr	$\mathcal{R} \to$ Rw	Avg	$\mathcal{R} \to$ P	$\mathcal{R} \to$ A	$\mathcal{R} \to$ C	$\mathcal{R} \to$ S	Avg
DeepCluster	19.6	18.9	18.7	19.1	8.9	11.1	16.9	13.3	12.6	27.9	22.2	24.4	27.1	25.4
IIC	31.9	37.0	34.0	34.4	12.0	15.2	22.5	15.9	16.4	70.6	39.8	39.6	46.6	49.2
ACIDS	33.4	37.5	36.1	35.7	12.0	16.2	23.9	15.7	17.0	64.4	42.1	44.5	51.1	50.5
PO	14.1 ± 1.6	17.9 ± 2.0	18.3 ± 2.9	16.8	11.4 ± 1.6	9.0 ± 1.6	12.9 ± 2.8	10.8 ± 1.7	11.0	30.5 ± 3.1	24.1 ± 0.6	19.8 ± 3.7	20.8 ± 1.7	23.8
SO	34.5 ± 0.5	46.7 ± 2.9	43.0 ± 2.9	41.4	23.6 ± 1.6	15.6 ± 1.9	23.1 ± 3.7	21.8 ± 2.9	21.0	30.8 ± 8.2	35.7 ± 3.9	27.6 ± 8.3	26.0 ± 3.7	30.0
TO	38.0 ± 3.2	46.6 ± 1.6	45.3 ± 1.5	43.3	21.3 ± 2.6	12.2 ± 0.7	30.6 ± 4.1	24.2 ± 0.7	22.1	88.4 ± 3.9	**56.5** ± 4.1	56.5 ± 11.1	49.1 ± 2.8	62.6
AO	42.8 ± 0.9	58.4 ± 3.7	55.8 ± 1.9	52.3	30.0 ± 1.7	22.7 ± 1.6	29.3 ± 4.1	24.4 ± 2.6	26.6	91.5 ± 5.9	47.7 ± 5.7	52.3 ± 1.2	49.1 ± 3.0	60.2
PCD	**46.8** ± 1.7	**60.0** ± 2.6	**57.8** ± 5.9	**54.9**	**33.3** ± 1.0	**24.4** ± 1.5	**31.4** ± 4.7	**28.1** ± 2.5	**29.3**	**92.6** ± 2.4	49.7 ± 5.0	**56.7** ± 2.6	**53.4** ± 5.9	**63.4**

Table 7: Clustering accuracy (%) on different datasets for ResNet-18-based methods (supervised pre-training).

Settings	PACS				
	$\mathcal{R} \to$ P	$\mathcal{R} \to$ A	$\mathcal{R} \to$ C	$\mathcal{R} \to$ S	Avg
ACIDS	80.9	48.2	50.5	**56.7**	59.1
PCD	**92.6**	**49.7**	**56.7**	53.4	**63.4**

Table 8: Clustering accuracy (%) on different initialization strategies for ResNet-50-based methods (supervised pre-training).

Settings	PACS				
	$\mathcal{R} \to$ P	$\mathcal{R} \to$ A	$\mathcal{R} \to$ C	$\mathcal{R} \to$ S	Avg
Self-supervised pre-training	82.1	53.4	50.8	43.6	57.5
Supervised pre-training	**93.3**	**54.6**	**59.1**	**56.8**	**66.0**

Table 9: Clustering accuracy (%) on different datasets for ResNet-50-based methods (self-supervised pre-training).

Settings	Office-31				Office-Home					PACS				
	$\mathcal{R} \to$ A	$\mathcal{R} \to$ W	$\mathcal{R} \to$ D	Avg	$\mathcal{R} \to$ Ar	$\mathcal{R} \to$ Cl	$\mathcal{R} \to$ Pr	$\mathcal{R} \to$ Rw	Avg	$\mathcal{R} \to$ P	$\mathcal{R} \to$ A	$\mathcal{R} \to$ C	$\mathcal{R} \to$ S	Avg
PO	13.5 ± 0.9	16.7 ± 0.3	19.3 ± 2.4	16.5	10.5 ± 0.6	8.4 ± 0.2	10.5 ± 0.7	9.1 ± 0.9	9.6	28.3 ± 7.4	22.9 ± 2.6	24.0 ± 1.2	29.1 ± 2.3	26.1
SO	19.0 ± 5.0	26.3 ± 2.0	27.5 ± 3.0	24.3	18.3 ± 1.2	11.2 ± 0.3	16.4 ± 1.3	16.7 ± 1.7	15.7	40.7 ± 12.2	25.0 ± 1.4	29.7 ± 5.7	35.0 ± 5.0	32.6
TO	31.6 ± 1.8	34.3 ± 4.3	33.7 ± 2.8	33.2	17.9 ± 2.0	10.1 ± 0.1	20.7 ± 1.9	16.6 ± 1.8	16.3	80.4 ± 6.8	51.9 ± 2.4	44.8 ± 1.3	32.8 ± 1.3	52.5
AO	33.3 ± 0.6	37.6 ± 5.3	41.9 ± 2.7	37.6	21.7 ± 2.2	17.9 ± 0.8	20.8 ± 3.7	27.5 ± 3.4	22.0	80.0 ± 3.8	38.9 ± 3.6	55.6 ± 3.4	43.5 ± 3.7	54.5
PCD	**37.8** ± 1.5	**48.2** ± 5.4	**51.0** ± 4.8	**45.6**	**23.8** ± 0.9	**18.4** ± 0.4	**30.6** ± 1.5	**27.6** ± 1.2	**25.1**	**82.1** ± 4.0	**53.4** ± 4.4	**50.8** ± 4.5	**43.6** ± 4.7	**57.5**

A.2 Model transfer setting

Table 10: Clustering accuracy (%) on Office-31 for different model transfer methods.

Settings	ViT-B/16 (ssl) → ResNet-50 (ssl)				ViT-B/16 (ssl) → ResNet-50 (sup)				ViT-B/16 (ssl) → ResNet-18 (sup)			
	$\mathcal{R} \to$ A	$\mathcal{R} \to$ W	$\mathcal{R} \to$ D	Avg	$\mathcal{R} \to$ A	$\mathcal{R} \to$ W	$\mathcal{R} \to$ D	Avg	$\mathcal{R} \to$ A	$\mathcal{R} \to$ W	$\mathcal{R} \to$ D	Avg
PO	20.1 ± 0.2/13.5 ± 0.9	26.7 ± 0.8/16.7 ± 0.3	27.2 ± 0.3/19.3 ± 2.4	24.7/16.5	20.1 ± 0.2/15.7 ± 0.7	26.7 ± 0.8/24.2 ± 3.5	27.2 ± 0.3/18.8 ± 2.2	24.7/19.6	20.1 ± 0.2/14.1 ± 1.6	26.7 ± 0.8/17.9 ± 2.0	27.2 ± 0.3/18.3 ± 2.9	24.7/16.8
SO	43.2 ± 5.0	46.4 ± 4.5	37.2 ± 9.0	42.3	43.2 ± 5.0	46.4 ± 4.5	37.2 ± 9.0	42.3	43.2 ± 5.0	46.4 ± 4.5	37.2 ± 9.0	42.3 ±
TO	32.6 ± 1.8	34.3 ± 4.3	33.7 ± 2.8	33.5	43.7 ± 1.6	55.8 ± 2.1	**52.0** ± 3.8	50.5	38.0 ± 3.2	45.3 ± 1.6	46.6 ± 1.5	43.3
AO	50.6 ± 3.7	49.7 ± 4.0	36.4 ± 5.0	45.6	52.5 ± 3.5	53.7 ± 1.9	44.2 ± 4.1	50.1	53.0 ± 3.8	47.2 ± 3.3	43.9 ± 4.9	48.0
PCD	**51.7** ± 2.9	**51.7** ± 2.0	**41.8** ± 3.2	**48.4**	**54.4** ± 2.4	**60.8** ± 2.1	49.2 ± 3.3	**54.8**	**54.6** ± 2.5	**53.6** ± 5.4	**46.7** ± 4.8	**51.6**

A.3 Limited-data and cluster-imbalanced and setting

Due to space constraints, we provide additional results for the sub-sampled versions of all three datasets in the appendix. PCD again outperforms other alternative methods consistently. AO, on

Table 11: Clustering accuracy $(\%)$ on sub-sampled versions of different datasets for ResNet-50-based methods (self-supervised pre-training).

Settings	Office-31				Office-Home					PACS				
	$\mathcal{R} \rightarrow$ sub-A	$\mathcal{R} \rightarrow$ sub-W	$\mathcal{R} \rightarrow$ sub-D	Avg	$\mathcal{R} \rightarrow$ sub-Ar	$\mathcal{R} \rightarrow$ sub-Cl	$\mathcal{R} \rightarrow$ sub-Pr	$\mathcal{R} \rightarrow$ sub-Rw	Avg	$\mathcal{R} \rightarrow$ sub-P	$\mathcal{R} \rightarrow$ sub-A	$\mathcal{R} \rightarrow$ sub-C	$\mathcal{R} \rightarrow$ sub-S	Avg
PO	14.6 ± 1.2	16.7 ± 1.4	21.5 ± 1.5	17.6	12.5 ± 0.8	9.3 ± 0.8	14.4 ± 1.1	12.2 ± 1.3	12.1	43.0 ± 8.9	32.3 ± 5.9	29.1 ± 1.1	39.8 ± 3.7	36.1
SO	21.1 ± 3.0	32.5 ± 2.6	36.0 ± 3.2	29.9	26.2 ± 1.3	19.5 ± 0.3	23.9 ± 1.0	26.9 ± 1.1	24.1	54.8 ± 4.4	37.8 ± 4.2	41.0 ± 5.5	47.2 ± 6.8	45.2
TO	31.4 ± 3.0	41.9 ± 3.6	45.1 ± 3.1	39.5	21.6 ± 1.8	11.9 ± 1.0	28.7 ± 1.3	22.8 ± 5.3	21.2	65.1 ± 2.8	46.4 ± 2.3	47.8 ± 5.4	40.9 ± 1.2	50.0
AO	34.7 ± 3.5	40.8 ± 3.9	43.9 ± 3.6	39.8	28.1 ± 0.6	21.8 ± 0.6	30.3 ± 2.6	29.4 ± 2.1	27.4	65.4 ± 5.4	43.2 ± 6.7	51.1 ± 1.8	$\mathbf{43.4} \pm 2.3$	50.7
PCD	$\mathbf{37.8} \pm 3.6$	$\mathbf{46.4} \pm 3.3$	$\mathbf{47.0} \pm 3.7$	**43.7**	$\mathbf{28.7} \pm 1.0$	$\mathbf{22.3} \pm 0.4$	$\mathbf{32.7} \pm 2.6$	$\mathbf{31.2} \pm 3.2$	**28.7**	$\mathbf{66.1} \pm 4.7$	$\mathbf{48.0} \pm 1.5$	$\mathbf{51.8} \pm 2.0$	$\mathbf{43.4} \pm 1.9$	**52.4**

average, performs better than TO, meaning that knowledge from the source can benefit target training. Similarly, SO improves upon PO in all cases. PCD achieves higher clustering accuracy than AO $(1 - 4\%)$, illustrating that target model refinement is crucial for PCD's success.

A.4 Ablation study

Table 12: Full ablation study on Office-31 dataset.

1lightgraylightgray

Settings	$\mathcal{R} \rightarrow$W	diff	$\mathcal{R} \rightarrow$ A	diff	$\mathcal{R} \rightarrow$ D	diff
Full	60.0	0.0	46.8	0.0	57.8	0.0
w/o prototype clustering	49.7	-10.3	43.2	-3.6	53.2	-4.6
w/o MI	52.7	-7.3	39.1	-7.7	54.79	-3.01
w/o CutMix	54.5	-5.5	46.1	-0.7	54.6	-3.2
w/o Temporal Ensemble	58.3	-1.7	46.6	-0.2	57.3	-0.5
w/o model privacy	63.1	3.1	49.2	2.4	61.4	3.6
pooled source	57.8	-2.2	45.6	-1.2	57.0	-0.8

Table 13: Clustering accuracy (%) on the task $\mathcal{R} \rightarrow$ W (Office-31) under different variants (ResNet-18).

Full	w/o prototype clustering	w/o MI	w/o CutMix	w/o Temporal Ensemble	w/o model privacy	pooled source
60.0 ± 2.6	49.7 ± 2.8	52.7 ± 3.0	54.5 ± 5.2	58.3 ± 2.4	63.1 ± 2.1	57.8 ± 2.5

B Sensitivity plot

In Figure 4, we plot the sensitivity of the target clustering accuracy when we vary the coefficient in front of the loss. We can see that our method is not sensitive to different values of the coefficients except for when the λ_{mix} coefficient is set to 5. This result is expected since the λ_{mix} is used as a regularization term and should not be set too high. We also observe that the performance can get even better via oracle validation by setting the λ_{mi} to 2 or 5. However, we set the coefficient to 1 for all three losses for all experiments.

C Running time and parameter size

Table 14: Number of parameters and average running time per step for different clustering approaches for ResNet-18-based models.

Methods	Parameter size (millions)	Running time (s/step)
ACIDS	11.94 M	head1 - 0.52 s/step / head2 - 0.44 s/step
PCD	11.32 M	0.16 s/step

D Connection with DeepCluster and SeLa

Caron et al. (2018) propose DeepCluster to perform clustering and representation learning simultaneously. This method alternates between K-means for clustering and cross-entropy minimization

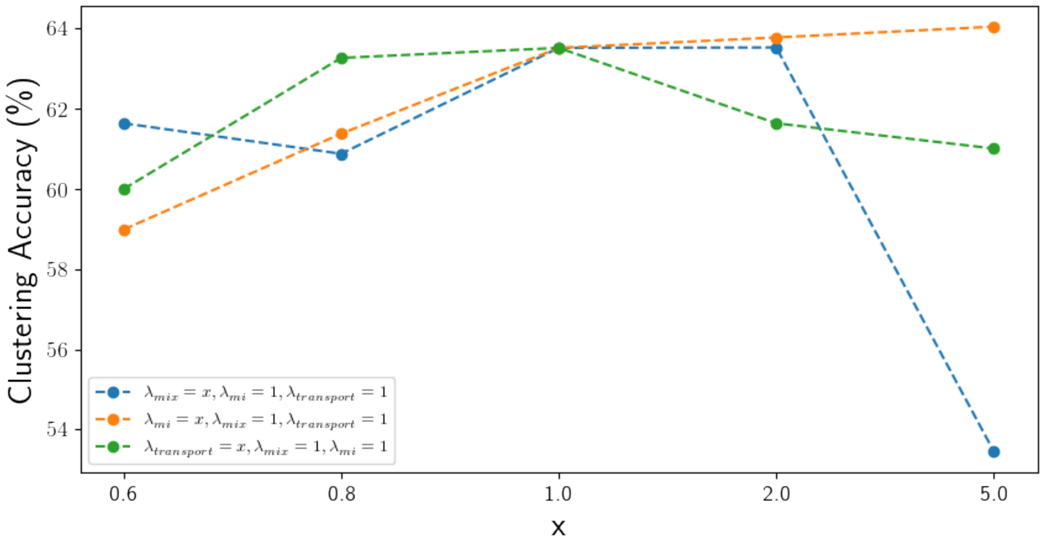


Figure 4: Sensitivity plot for the coefficient of the losses. We fix the coefficient of the two losses to 1.0 while varying the third loss from 0.6 to 5.0 and plot the clustering accuracy on the target data.

Table 15: Average running time per step for different clustering approaches for ResNet-18-based models.

ACIDS	PCD
head1 - 0.52 s/step / head2 - 0.44 s/step	0.16 s/step

for representation learning. While compatible with deep learning frameworks, the approach does have an obvious degenerate solution where all the samples get assigned to one cluster, yielding a constant representation. To overcome this, Asano et al. (2019) invent SeLa, which is similar to DeepCluster in the cross-entropy minimization step but differs from it in the pseudo-label assignment step. The authors explain that solving the K-means problem with equal partitioning constraints can avoid the degenerate solution. Asano et al. (2019) further recognize this as an instance of an optimal transport problem. Our clustering method is similar to SeLa in that we also solve the optimal transport problem during the pseudo-label assignment step. Unlike SeLa, we do not use the simplistic assumption that each cluster contains an equal number of data points. Instead, we dynamically update the cluster proportions using the predicted cluster probabilities. We also offer the interpretation of our method from the distribution alignment perspective. Moreover, our method is designed specifically for multi-domain data, and we also explore the use of our framework under the domain shift scenario.

E Pseudo-code

F Full implementation details

We follow the standard protocols for source-free unsupervised domain adaptation (Liang et al., 2020). Specifically, we use mini-batch SGD with a momentum of 0.9 and weight decay of 0.001. Both source and target encoders are initialized with ImageNet pre-trained networks (Russakovsky et al., 2015), but the prototypes are initialized with a random linear layer. The initial learning rates are set to 0.001 for the pre-trained encoders and 0.01 for the randomly initialized layer. The learning rates, η, follows the following schedule: $\eta = \eta_0(1 + 10p)^{-0.75}$ where η_0 is the initial learning rate. We use the batch size of 64 in both source and target learning. The initial value β_0 to learn domain-specific proportions is set to 0.9999 for source clustering and 0.99 for target clustering in all settings. We set the entropic regularization parameter, ϵ, to 0.01. The concentration parameter, α, in the CutMix

Algorithm 1: Pseudo code for our framework.

```
1. Source model training
Input: source data - $\mathcal{X}^s = \{\mathcal{X}^s_d\}_{d=1}^D$, source model - $G^s = C_{\boldsymbol{\mu}^s}(F_{\boldsymbol{\theta}^s}(.))$ (a randomly initialized
$C_{\boldsymbol{\mu}^s}$ and a pre-trained $F_{\boldsymbol{\theta}^s}$)
Output: updated $\boldsymbol{\theta}^s$, $\boldsymbol{\mu}^s$
for $t = 1$ to $T$ do
  • Sample a mini-batch of source data
  • Update the proportions $B$ with Eq. (2)
  • Solve the optimal transport problem in Eq. (1) to obtain the transport map for each domain
  • Update the encoder and prototypes using Eq. (4) with the transport map from the previous step
end for
2. Target model clustering
Input: target data - $\mathcal{X}^t = \{\boldsymbol{x}^t_j\}_{j=1}^{n_t}$, cluster labels from the source model - $G^s(x^t)$, target model -
$G^t = C_{\boldsymbol{\mu}^t}(F_{\boldsymbol{\theta}^t}(.))$ (a randomly initialized $C_{\boldsymbol{\mu}^t}$ and a pre-trained $F_{\boldsymbol{\theta}^t}$)
Output: updated $\boldsymbol{\theta}^t$, $\boldsymbol{\mu}^t$
for $t = 1$ to $T$ do
  • Sample a mini-batch of target data
  • Refine the hard-label with label smoothing and temporal ensemble
  • Update the the target model with the loss in Eq. (2.3.1)
end for
3. Target model refinement
Input: $\mathcal{X}^t = \{\boldsymbol{x}^t_j\}_{j=1}^{n_t}$, target model - $G^t$ ($C_{\boldsymbol{\mu}^t}$ and $F_{\boldsymbol{\theta}^t}$ from step 2's output)
Output: updated $\boldsymbol{\theta}^t$, $\boldsymbol{\mu}^t$
for $t = 1$ to $T$ do
  • Sample a mini-batch of target data
  • Update the proportions $B$ with Eq. (2)
  • Solve the optimal transport problem in Eq. (1) to obtain the transport map for the target
    domain
  • Update the encoder and prototype using Eq. (2.3.2) with the transport map from the previous
    step
end for
```

loss is set to 0.3. The temporal ensemble coefficient, τ, is equal to 0.6. The source model and hyper-parameters are selected using the validation set of the source domain. The target model is trained using all the target data. We run our method with three different random seeds to calculate the standard deviation. We implement our method in PyTorch.