Prototypical Partial Optimal Transport for Universal Domain Adaptation

Yucheng Yang[*], **Xiang Gu**[*], **Jian Sun**[†]

School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China
{ycyang, xianggu}@stu.xjtu.edu.cn, jiansun@xjtu.edu.cn

Abstract

Universal domain adaptation (UniDA) aims to transfer knowledge from a labeled source domain to an unlabeled target domain without requiring the same label sets of both domains. The existence of domain and category shift makes the task challenging and requires us to distinguish "known" samples (*i.e.*, samples whose labels exist in both domains) and "unknown" samples (*i.e.*, samples whose labels exist in only one domain) in both domains before reducing the domain gap. In this paper, we consider the problem from the point of view of distribution matching which we only need to align two distributions partially. A novel approach, dubbed mini-batch Prototypical Partial Optimal Transport (m-PPOT), is proposed to conduct partial distribution alignment for UniDA. In training phase, besides minimizing m-PPOT, we also leverage the transport plan of m-PPOT to reweight source prototypes and target samples, and design reweighted entropy loss and reweighted cross-entropy loss to distinguish "known" and "unknown" samples. Experiments on four benchmarks show that our method outperforms the previous state-of-the-art UniDA methods.

1 Introduction

Deep Learning has achieved significant progress in image recognition (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2014). However, deep learning based methods heavily rely on in-domain labeled data for training, to be generalized to target domain data. Considering that collecting annotated data for every possible domain is labour-intensive and time-consuming, a feasible solution is unsupervised domain adaptation (UDA) (Ben-David et al. 2010; Ganin and Lempitsky 2015; Long et al. 2018), which transfers the knowledge from labeled source domain to unlabeled target domain by alleviating distribution discrepancy between them. The most common setting in UDA is closed-set DA which assumes the source class set C_s is identical to the target class set C_t. This may be impractical in real-world applications, because it is difficult to ensure that the target dataset always has the same classes as the source dataset.

To tackle this problem, some works consider more general domain adaptation tasks. For example, partial domain adaptation (PDA) (Cao et al. 2018a) assumes that target class set is a subset of source class set, *i.e.*, $C_t \subseteq C_s$. Open-set domain adaptation (OSDA) (Saito et al. 2018) exploits the situation where source class set is a subset of target class set $C_s \subseteq C_t$. Universal domain adaptation (UniDA) (You et al. 2019) is a more general setting that both source and target domains possibly have common and private classes. The setting of UniDA includes PDA, OSDA, and a mixture of PDA and OSDA, *i.e.*, open-partial DA (OPDA) (Panareda Busto and Gall 2017), in which both source and target domains have private classes. This paper focuses on the general UniDA setting. The goal of UniDA is to classify target domain common class samples and detect target-private class samples, meanwhile reducing the negative transfer possibly caused by source private classes. To reduce the domain gap in UniDA, we may use distribution alignment techniques as in UDA methods (Courty et al. 2017; Ganin and Lempitsky 2015), to align the distributions of two domains. However, matching all data of two domains may lead to the mismatch of the common class data of one domain to the private class data in the other domain, and cause negative transfer.

In this work, we propose a novel Prototypical Partial Optimal Transport (PPOT) approach to tackle UniDA. Specifically, we model distribution alignment in UniDA as a partial optimal transport (POT) problem, to align a fraction of data (mainly from common classes), between two domains using POT. We design a prototype-based POT, in which the source data are represented as prototypes in POT formulation, which is further formulated as a mini-batch-based version, dubbed m-PPOT. We prove that POT can be bounded by m-PPOT and the distances between source samples and their corresponding prototypes, inspiring us to design a deep learning model for UniDA by using m-PPOT as one training loss. Meanwhile, the transport plan of m-PPOT can be regarded as a matching matrix, enabling us to utilize the row sum and column sum of the transport plan to reweight the source prototypes and target samples for distinguishing "known" and "unknown" samples. Based on the transport plan of m-PPOT, we further design reweighted cross-entropy loss on source labeled data and reweighted entropy loss on target data to learn a transferable recognition model.

In experiments, we evaluate our method on four UniDA benchmarks. Experimental results show that our method performs favorably compared with the state-of-the-art methods

[*] These authors contributed equally.

[†] Corresponding Author.

for UniDA. Code is available at https://github.com/ycyang-xjtu/PPOT.

2 Related Work

2.1 Domain Adaptation

Unsupervised domain adaptation aims to reduce the gap between source and target domains. Previous works (Ganin and Lempitsky 2015; Long et al. 2015, 2018) mainly focus on distribution alignment to mitigate domain gaps. The theoretical analysis in (Ben-David et al. 2010) shows that minimizing the discrepancy between source and target distributions may reduce the target prediction error. Previous works often minimize distribution discrepancy between two domains by adversarial learning (Ganin and Lempitsky 2015; Long et al. 2018; Zhang et al. 2019) and moment matching (Long et al. 2015; Sun and Saenko 2016; Pan et al. 2019). Partial DA (Cao et al. 2018b) tackles the scenario that only the source domain contains private classes. The methods in (Cao et al. 2018a; Zhang et al. 2018; Gu et al. 2021) are mainly based on reweighting source data for reducing negative transfer caused by source private class samples. Open-set DA (Saito et al. 2018) assumes that the label set of the source domain is a subset of that of the target domain. (Saito et al. 2018; Liu et al. 2019; Bucci, Loghmani, and Tommasi 2020) propose diverse methods to classify "known" samples meanwhile rejecting "unknown" samples.

2.2 Universal Domain Adaptation

Universal DA does not have any prior knowledge on the label space of two domains, which means that both source and target domains may or may not have private classes. UAN (You et al. 2019) computes the transferability of samples by entropy and domain similarity to separate "known" and "unknown" samples. CMU (Fu et al. 2020) improves UAN to measure transferability by a mixture of entropy, confidence, and consistency from ensemble model. DANCE (Saito et al. 2020) designs an entropy-based method by increasing the confidence of common class samples while decreasing it for private class samples to better distinguish known and unknown samples. DCC (Li et al. 2021) tries to exploit the domain consensus knowledge to discover matched clusters for separating common classes in cluster-level. OVANet (Saito and Saenko 2021) trains a "one-vs-all" discriminator for each class to recognize private class samples. GATE (Chen et al. 2022) explores the intrinsic geometrical relationship between the two domains and designs a universal incremental classifier to separate "unknown" samples. Different from the above methods, we model the UniDA as a partial distribution alignment problem and propose a novel m-PPOT model to solve it.

2.3 Optimal Transport

Optimal transport (OT) (Villani 2009; Peyré, Cuturi et al. 2019) is a mathematical tool for transporting/matching distributions. OT has been applied to diverse tasks such as generative adversarial training (Arjovsky, Chintala, and Bottou 2017), clustering (Ho et al. 2017), domain adaptation (Courty et al. 2017), object detection (Ge et al. 2021), etc. The partial OT (Caffarelli and McCann 2010; Figalli 2010) is a special OT problem that only transports a portion of the mass. To reduce computational cost of OT, the Sinkhorn OT (Cuturi 2013) can be efficiently solved by the Sinkhorn algorithm, and is further extended to partial OT in (Benamou et al. 2015). In (Flamary et al. 2016; Courty et al. 2017; Damodaran et al. 2018), OT was applied to domain adaptation to align distributions of source and target domains in input space or feature space. They use OT in mini-batch to reduce computational overhead, however, suffering from sampling bias that the mini-batch data partially reflect the original data distribution. (Fatras et al. 2021; Nguyen et al. 2022) replace mini-batch OT with more robust OT models, such as unbalanced mini-batch OT and partial mini-batch OT, and achieve better performance. (Xu et al. 2021) designs joint partial optimal transport which only transports a fraction of the mass for avoiding negative transfer, and extends the task into open-set DA.

In this work, we consider the UniDA task. We propose a novel mini-batch based prototypical POT model, which partially aligns the source prototypes and target features to solve the problem of UniDA. Experiments show that our method achieves state-of-the-art results for UniDA.

3 Preliminaries on Optimal Transport

We consider two sets of data points, $\{x_i^s\}_{i=1}^m$ and $\{x_j^t\}_{j=1}^n$, of which the empirical distributions are denoted as $\boldsymbol{\mu} = \sum_{i=1}^m \mu_i \delta_{x_i^s}$ and $\boldsymbol{\nu} = \sum_{j=1}^n \nu_j \delta_{x_j^t}$ respectively, where $\sum_{i=1}^m \mu_i = 1$, $\sum_{j=1}^n \nu_j = 1$ and δ_x is the Dirac function at position x. With a slight abuse of notations, we denote $\boldsymbol{\mu} = (\mu_1, \mu_2, \cdots, \mu_m)^\top$, $\boldsymbol{\nu} = (\nu_1, \nu_2, \cdots, \nu_n)^\top$ and define a cost matrix as $C \in \mathbb{R}^{m \times n}$, $C_{ij} = c(x_i^s, x_j^t)$.

Kantorovich problem. The Kantorovich problem (Kantorovitch 1958) aims to derive a transport plan from $\boldsymbol{\mu}$ to $\boldsymbol{\nu}$, modeled as the following linear programming problem:

$$\begin{aligned} &\mathrm{OT}(\boldsymbol{\mu}, \boldsymbol{\nu}) \triangleq \min_{\pi \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \langle \pi, C \rangle_F \\ &s.t.\ \ \Pi(\boldsymbol{\mu}, \boldsymbol{\nu}) = \{\pi \in \mathbb{R}_+^{m \times n} | \pi \mathbb{1}_n = \boldsymbol{\mu}, \pi^\top \mathbb{1}_m = \boldsymbol{\nu}\}, \end{aligned} \tag{1}$$

where $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product.

Mini-batch OT is designed to reduce computational cost and make OT more suitable for deep learning. We denote the collection of empirical distributions of b random samples in $\{x_i^s\}_{i=1}^m$ (resp. $\{x_j^t\}_{j=1}^n$) as $\mathcal{P}_b(\boldsymbol{\mu})$ (resp. $\mathcal{P}_b(\boldsymbol{\nu})$), where b is the batch size, and k is the number of mini-batches. The mini-batch OT is defined as

$$\text{m-OT}_k(\boldsymbol{\mu}, \boldsymbol{\nu}) = \frac{1}{k} \sum_{i=1}^k \mathrm{OT}(A_i, B_i), \tag{2}$$

where $A_i \in \mathcal{P}_b(\boldsymbol{\mu})$, $B_i \in \mathcal{P}_b(\boldsymbol{\nu})$ for any $i = 1, 2, ..., k$.

Partial OT aims to transport only α mass ($0 \leqslant \alpha \leqslant \min(\|\boldsymbol{\mu}\|_1, \|\boldsymbol{\nu}\|_1)$) between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ with the lowest cost. The partial OT is defined as

$$\mathrm{POT}^\alpha(\boldsymbol{\mu}, \boldsymbol{\nu}) \triangleq \min_{\pi \in \Pi^\alpha(\boldsymbol{\mu}, \boldsymbol{\nu})} \langle \pi, C \rangle_F, \tag{3}$$

where $\Pi^\alpha(\boldsymbol{\mu}, \boldsymbol{\nu}) = \{\pi \in \mathbb{R}_+^{m \times n} | \pi \mathbb{1}_n \leqslant \boldsymbol{\mu}, \pi^\top \mathbb{1}_m \leqslant \boldsymbol{\nu}, \mathbb{1}_m^\top \pi \mathbb{1}_n = \alpha\}$.

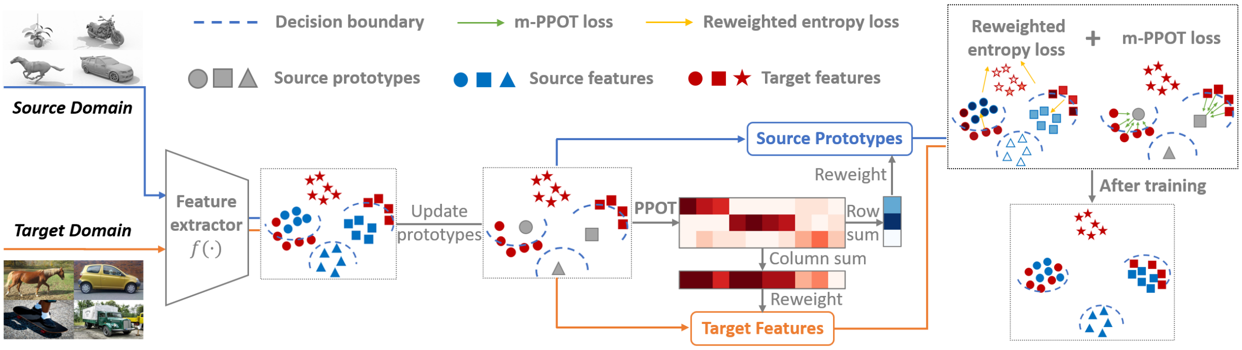


Figure 1: **Illustration of our model.** Source and target data share the same feature extractor that embeds data in feature space. PPOT is to match target features and source prototypes which are updated by the source features, and the row/column sum of transport plan is applied for reweighting. We design reweighted entropy loss to align common class features of two domains, while pushing away the unknown features.

4 Method

In this section, we model UniDA as a partial distribution alignment problem. To partially align source and target distributions, the mini-batch prototypical partial optimal transport (m-PPOT) is proposed. The m-PPOT focuses on the discrete partial OT problem between source prototypes and target samples for mini-batch. Based on m-PPOT, we design a novel model for UniDA. We also use contrastive pre-training to have a better initialization of network parameters.

In UniDA, we are given labeled source data $\boldsymbol{D_s} = \{x_i^s, y_i\}_{i=1}^m$ and unlabeled target data $\boldsymbol{D_t} = \{x_j^t\}_{j=1}^n$. UniDA aims to label the target sample with a label from source class set C_s or discriminate it as an "unknown" sample. We denote the number of source domain classes as $L = |C_s|$. Our deep recognition model consists of two modules, including a feature extractor f mapping input x into feature z, and an L-way classification head h. The source and target empirical distributions in feature space are denoted as $\bar{\boldsymbol{p}} = \sum_{i=1}^m p_i \delta_{f(x_i^s)}$ and $\bar{\boldsymbol{q}} = \sum_{j=1}^n q_j \delta_{f(x_j^t)}$ respectively, where $\sum_{i=1}^m p_i = 1, \sum_{j=1}^n q_j = 1$. With a slight abuse of notations, we denote the vector of data mass as $\bar{\boldsymbol{p}} = (p_1, p_2, ..., p_m)^\top$ and $\bar{\boldsymbol{q}} = (q_1, q_2, ..., q_n)^\top$ and we set $p_i = \frac{1}{m}$, $q_j = \frac{1}{n}$ for any i, j in this paper. Furthermore, the element of cost matrix C is defined as $C_{ij} = d(f(x_i^s), f(x_j^t))$, where d is the L_2-distance.

4.1 Modeling UniDA as Partial OT

(Ben-David et al. 2006, 2010) presented theoretical analysis on domain adaptation, emphasizing the importance of minimizing distribution discrepancy. However, it can not be simply extended to UniDA because the source/target data may belong to source/target private classes in UniDA setting. Directly aligning source distribution $\bar{\boldsymbol{p}}$ and target distribution $\bar{\boldsymbol{q}}$ will lead to data mismatch due to the existence of "unknown" samples in both domains. For UniDA task, we first decompose $\bar{\boldsymbol{p}}$, $\bar{\boldsymbol{q}}$ as

$$\bar{\boldsymbol{p}} = (1-\beta)\boldsymbol{p}_p + \beta \boldsymbol{p}_c, \quad \bar{\boldsymbol{q}} = (1-\alpha)\boldsymbol{q}_p + \alpha \boldsymbol{q}_c,$$

where $\boldsymbol{p}_p$ (resp. $\boldsymbol{q}_p$) denotes distribution of source (resp. target) private class data in feature space, $\boldsymbol{p}_c$ and $\boldsymbol{q}_c$ are denoted as source and target common class data distributions, α and β are the ratio of common class samples in the source and target domain respectively. Our goal is to minimize the discrepancy between $\boldsymbol{p}_c$ and $\boldsymbol{q}_c$, formulated as an OT problem:

$$\min_{f,\pi} \langle \pi, \bar{C} \rangle_F = \min_f \mathrm{OT}(\boldsymbol{p}_c, \boldsymbol{q}_c), \tag{4}$$

where $\bar{C} \in \mathbb{R}^{|\boldsymbol{p}_c| \times |\boldsymbol{q}_c|}$ is a submatrix of C, corresponding to the common class samples.

Obviously, we can not directly get these two distributions. Therefore, we consider to find an approximation of Eqn. (4). Following the assumption in (You et al. 2019) that $\boldsymbol{q}_c$ is closer to $\boldsymbol{p}_c$ than $\boldsymbol{q}_p$, meaning that the cost of transport between two domains' common class samples is generally less than the cost between two private class samples of these two domains or the private and common class samples of them. Note that partial OT only transports a fraction of the mass having lowest cost to transport. With the above assumption, the partial transport between two domains will prefer to transfer the common class samples of them. Therefore, we approximately solve Eqn. (4) by optimizing

$$\min_f \mathrm{POT}^\alpha(\frac{\alpha}{\beta}\bar{\boldsymbol{p}}, \bar{\boldsymbol{q}}) \tag{5}$$

where coefficient (α/β) is to ensure that the mass of common class samples in $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{q}}$ equals. The superscript α denotes the total mass to transport. For convenience of presentation, $(\alpha/\beta) \cdot \bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{q}}$ are denoted as $\boldsymbol{p}$ and $\boldsymbol{q}$ respectively.

4.2 Prototypical Partial Optimal Transport

We have turned the distribution alignment between $\boldsymbol{p}_c$ and $\boldsymbol{q}_c$ into a partial OT problem in Eqn. (5). The remaining challenge is to embed partial OT into a deep learning framework. In this paper, we design a mini-batch based prototypical partial optimal transport problem for UniDA. We first define the Prototypical Partial Optimal Transport (PPOT).

Definition 1. *(Prototypical Partial Optimal Transport) Let* $\{c_i\}_{i=1}^L$ *be the set of source domain prototypes, defined as*

$$c_i = \sum_{j:y_j=i} \frac{f(x_j^s)}{\sum_{l=1}^m \mathbf{1}(y_l = i)}.$$

The element of cost matrix C_{ij} is defined as $d(c_i, f(x_j^t))$ and the PPOT transportation cost between $\boldsymbol{p}$ and $\boldsymbol{q}$ is defined as

$$\mathrm{PPOT}^{\alpha}(\boldsymbol{p},\boldsymbol{q}) \triangleq \mathrm{POT}^{\alpha}(\boldsymbol{c},\boldsymbol{q}) = \min_{\pi\in\Pi^{\alpha}(\boldsymbol{c},\boldsymbol{q})} \langle \pi, C\rangle_F, \quad (6)$$

where $\boldsymbol{c} = \sum_{i=1}^{L} r_i \delta_{c_i}$ is the empirical distribution of source domain prototypes, and $r_i = \sum_{j:y_j=i} p_j$.

PPOT is suitable for the DA task because, first, it fits the mini-batch based deep learning implementation in which all of the prototypes, instead of batch of source samples, are regarded as source measures in POT. This change could reduce the mismatch caused by the lack of full coverage of source samples in a batch, and second, it requires less computational resources than original POT.

Mini-batch based PPOT. We further extend the PPOT to the mini-batch version m-PPOT, here we assume batch size b satisfy $b \mid n$ and set $k = n/b$. Let $\mathcal{B}_i$ be the i-th index set of b random target samples and their corresponding empirical distribution in feature space is denoted as $\boldsymbol{q}_{\mathcal{B}_i}$. We define $\mathcal{B} \triangleq \{\mathcal{B}_i\}_{i=1}^{k}$ as a partition if they satisfy:

- $\mathcal{B}_i \bigcap \mathcal{B}_j = \emptyset : \forall\ 0 \leqslant i < j \leqslant k$
- $\bigcup_{i=1}^{k} \mathcal{B}_i = \{1, 2, ..., n\}$

and the m-PPOT is defined as

$$\text{m-PPOT}_{\mathcal{B}}^{\alpha}(\boldsymbol{p},\boldsymbol{q}) \triangleq \frac{1}{k}\sum_{i=1}^{k} \mathrm{POT}^{\alpha}(\boldsymbol{c}, \boldsymbol{q}_{\mathcal{B}_i}), \ \ \mathcal{B} \in \Gamma \quad (7)$$

where Γ is the set of all partitions of $\{1, 2, ..., n\}$, *i.e.*, the index set of target data. Note that these assumptions are easily satisfied by the dataloader module in pytorch. Furthermore, we denote the optimal transportation in i-th batch as π_i^{α}. To show that m-PPOT is closely related to PPOT, we give the following proposition 1.

Proposition 1. *We extend π_i^{α} to a $L \times n$ matrix Π_i^{α} that pad zero entries to the column whose index does not belong to $\mathcal{B}_i$, then we have*

$$\frac{1}{k}\sum_{i=1}^{k} \Pi_i^{\alpha} \in \Pi^{\alpha}(\boldsymbol{c},\boldsymbol{q})$$

and

$$\mathrm{PPOT}^{\alpha}(\boldsymbol{p},\boldsymbol{q}) \leqslant \text{m-PPOT}_{\mathcal{B}}^{\alpha}(\boldsymbol{p}.\boldsymbol{q}). \quad (8)$$

Proposition 1 implies that m-PPOT$_{\mathcal{B}}^{\alpha}(\boldsymbol{p},\boldsymbol{q})$ is an upper bound for PPOT$^{\alpha}(\boldsymbol{p},\boldsymbol{q})$. The following theorem shows that POT is bounded by the sum of m-PPOT and the distances of source samples to their corresponding prototypes.

Theorem 1. *Considering two distributions $\boldsymbol{p}$ and $\boldsymbol{q}$, the distance between $f(x_i^s)$ and corresponding prototype c_{y_i} is denoted as $d_i \triangleq d(f(x_i^s), c_{y_i})$. The row sum of the optimal transport plan of PPOT$^{\alpha}(\boldsymbol{p},\boldsymbol{q})$ is denoted as $\boldsymbol{w} = (w_1, w_2, ..., w_L)^{\top}$, $r_i = \sum_{j:y_j=i} p_j$. Then we have*

$$\mathrm{POT}^{\alpha}(\boldsymbol{p},\boldsymbol{q}) \leqslant \sum_{i=1}^{m} \frac{w_{y_i}}{r_{y_i}} p_i d_i + \text{m-PPOT}_{\mathcal{B}}^{\alpha}(\boldsymbol{p},\boldsymbol{q}). \quad (9)$$

The proofs of theorem 1 and proposition 1 are included in Appendix.

4.3 UniDA based on m-PPOT

Our motivation is to minimize discrepancy between distributions of source and target common class data, meanwhile separating "known" and "unknown" data in both domains in training. We design the following losses for training.

m-PPOT loss. Based on theorem 1, to minimize the discrepancy between $\boldsymbol{p}_c$ and $\boldsymbol{q}_c$, we first design the m-PPOT loss to minimize the second term in the bound of theorem 1. We introduce the m-PPOT$_{\mathcal{B}}^{\alpha}(\boldsymbol{p},\boldsymbol{q})$ as a loss:

$$\mathcal{L}_{ot} = \mathop{\mathbb{E}}_{\mathcal{B}\in\Gamma}\left(\text{m-PPOT}_{\mathcal{B}}^{\alpha}(\boldsymbol{p},\boldsymbol{q})\right), \quad (10)$$

where $\mathbb{E}$ denotes the expectation over all target domain data index partitions in Γ. Using the mini-batch based optimization method, this term can be approximated by the partial OT problem POT$^{\alpha}(\boldsymbol{c}, \boldsymbol{q}_{\mathcal{B}_i})$ over each mini-batch, according to Eqn. (7). The set of prototypes $\boldsymbol{c}$ is updated by exponential moving average as in (Xie et al. 2018). We use the entropy regularized POT algorithm proposed by (Benamou et al. 2015) to solve POT on mini-batch.

Reweighted entropy loss. We further design entropy-based loss on target domain data to increase the prediction certainty. The solution π^* to the m-PPOT$_{\mathcal{B}}^{\alpha}(\boldsymbol{p},\boldsymbol{q})$ is a matrix measuring the matching between source prototypes and target features. Since the more easily a prototype (feature) can be transported, the more likely it belongs to a common class ("known" sample), we leverage the row/column sum of π^* as indicator to identify unknown samples. Specifically, we first get the column sum of π^* and multiply a constant n/α to make $\boldsymbol{w}^t \in \mathbb{R}^n$ satisfy $\|\boldsymbol{w}^t\|_1 = n$. A reweighted entropy loss is formulated as

$$\mathcal{L}_{pe} = -\sum_{i=1}^{n}\sum_{j=1}^{L} w_i^t p_{ij}\log(p_{ij}), w_i^t = \frac{n}{\alpha}\sum_{j=1}^{L}\pi_{ij}^*, \quad (11)$$

where $p_{ij} \triangleq \sigma(h \circ f(x_i^t))_j$. We take this loss to increase the confidence of prediction for those target samples seen as "known" samples.

Furthermore, we follow (Saito et al. 2020; Saito and Saenko 2021) to suppress the model to generate overconfident predictions for target "unknown" samples by loss

$$\mathcal{L}_{ne} = -\sum_{i=1}^{n}\sum_{j=1}^{L} w_i^u p_{ij}\log(p_{ij}), w_i^u = [1 - w_i^t]_+, \quad (12)$$

where w_i^u depends on w_i^t, and higher w_i^u for a sample means higher confidence to be an "unknown" sample. Therefore, we use $\mathcal{L}_{ne}$ to reduce the confidence of those samples which are likely to be "unknown" samples.

Reweighted cross-entropy loss. This loss is the classification loss defined in the source domain, based on the cross-entropy using labels of source domain data. Different to standard classification loss, we use the column sum of π^* to compute weights $\boldsymbol{w}^s \in \mathbb{R}^L$ for measuring the confidence of the "known" source domain prototypes. Then we design the reweighted cross-entropy loss

$$\mathcal{L}_{rce} = -\sum_{i=1}^{m}\sum_{j=1}^{L} w_j^s \mathbf{1}(y_i = j)\log(\sigma(h \circ f(x_i^s))_j) \quad (13)$$

where $w_j^s = \dfrac{L}{\alpha}\sum_{i=1}^n \pi_{ij}^*$ is the weight of j-th source prototype representing j-th class center. The weights satisfy $\sum_{j=1}^L w_j^s = L$ and each of them represents the possibility that each category belongs to a common class. (Papyan, Han, and Donoho 2020) shows that the cross-entropy based loss could minimize the distance of features to class prototype. This implies that the reweighted cross-entropy loss approximately minimizes the first term in bound of theorem 1, in which we use the row sum $\boldsymbol{w}^s$ of m-PPOT to approximate the row sum $\boldsymbol{w}$ of PPOT, and use the class-balanced sampling in implementation to enforce that $r_j, \forall j$, are equal.

Training loss and details. Our model is jointly optimized with the above loss terms, and the total training loss is

$$\mathcal{L} = \mathcal{L}_{rce} + \mathcal{L}_{ent} + \eta_1 \mathcal{L}_{ot}, \tag{14}$$

where $\mathcal{L}_{ent} = \eta_2 \mathcal{L}_{pe} - \eta_3 \mathcal{L}_{ne}$. In implementation, we set $\eta_1 = 5$, $\eta_2 = 0.01$, and $\eta_3 = 2$ for all datasets. The training process of our method is shown in Fig. 1. In the beginning, we map data in both domains into feature space by the feature extractor. Source prototypes are updated by source features in every batch and then we compute the m-PPOT between the empirical distributions of source prototypes and target samples, which we also leverage the row sum and column sum of corresponding transport plan to reweight in the losses. $\mathcal{L}_{ot}$ aims to reduce the gap between the distribution of "known" samples in both domains, meanwhile $\mathcal{L}_{ent}$ enforces the "known" samples to have higher prediction confidence by decreasing their entropy, and the "unknown" samples to have lower prediction confidence by increasing the entropy, in the target domain. Since the classifiers are learned over the source domain data, this may align the "known" target domain data to the source domain data distribution, while pushing the "unknown" target domain data away from the source domain data distribution.

Parameter initialization by contrastive pre-training. Motivated by (Shen et al. 2022), we use contrastive learning to pre-train our feature extractor. Specifically, we send both source and target unlabeled data into our feature extractor and use the contrastive learning method (MocoV2 (Chen et al. 2020)) to pre-train our feature extractor, then fine-tune the entire model on labeled source data, and take these parameters as our model's initial parameters. We empirically find that contrastive pre-training also works in UniDA setting in our experiments.

4.4 Hyper-parameters

We notice that α and β are nearly impossible to calculate precisely in practice, so we propose a method to compute them approximately. We denote two scalars as τ_1 and τ_2, where $\tau_1 \in (0, 1]$, $\tau_2 > 0$. To simplify the notation, we use $s(x) = \max \sigma(h \circ f(x))$ to denote the prediction confidence of x. We define α and β as

$$\alpha = \sum_{j=1}^n \frac{\mathbf{1}(s(x_j^t) \geqslant \tau_1)}{n}, \beta = \sum_{i=1}^L \frac{\mathbf{1}(w_i^s \geqslant \tau_2)}{L}. \tag{15}$$

The motivation is that we use the proportion of high-confidence samples to estimate the ratio of "known" samples in the target domain, and similarly use the proportion of categories with high weights to approximate the ratio of common classes in the source domain.

In experiments, we set $\tau_1 = 0.9$ and $\tau_2 = 1$. In the i-th iteration of training phase, we first calculate α^i by Eqn. (15), and update α by exponential moving average:

$$\alpha^i \leftarrow \lambda_1 \alpha^i + (1 - \lambda_1)\alpha^{i-1}.$$

Then we use α^i as transport ratio and α^i/β^{i-1} as coefficient of Eqn. (5) to compute $\mathcal{L}_{ot}$ and its by-product $\boldsymbol{w}^s$. After that we compute β^i by Eqn. (15) and update it as same as α^i:

$$\beta^i \leftarrow \lambda_2 \beta^i + (1 - \lambda_2)\beta^{i-1},$$

where $\lambda_1, \lambda_2 \in [0, 1)$ are set to 0.001 in our experiments.

Furthermore, to reduce the possible mistakes that identify a "known" sample as "unknown" sample, we retain only a fraction of $\{w_i^u\}_{i=1}^n$ that have larger values, and set the others as 0. The fraction is set to 25% in all tasks.

5 Experiment

We evaluate our method on UniDA benchmarks. We solve three settings of UniDA, including OPDA, OSDA, and PDA but without using prior knowledge about the mismatch of source and target domain class label sets.

Datasets. **Office-31** (Saenko et al. 2010) includes 4652 images in 31 categories from 3 domains: Amazon (**A**), DSLR (**D**), and Webcam (**W**). **Office-Home** (Venkateswara et al. 2017) consists of 15500 images in 65 categories, and it contains 4 domains: Artistic images (**A**), Clip-Art images (**C**), Product images (**P**), and Real-World images (**R**). **VisDA** (Peng et al. 2017) is a larger dataset which consists of 12 classes, including 150,000 synthetic images (**S**) and 50,000 images from real world (**R**). **DomainNet** (Peng et al. 2019) is one of the most challenging datasets in DA task with about 0.6 million images, which consists of 6 domains sharing 345 categories. We follow (Fu et al. 2020) to use 3 domains: Painting (**P**), Real (**R**), and Sketch (**S**). Following (Saito and Saenko 2021), we show the number of common classes, source private classes, and target private classes in brackets in the header of each result of tables.

Evaluation. In PDA tasks, we compute the accuracy for all target samples. In OSDA and OPDA settings, the target private class samples should be classified as a single category named "unknown". The samples with confidence less than threshold ξ are identified as "unknown", where ξ is set to 0.75 in all experiments. Following (Fu et al. 2020), we report the H-score metric for OSDA and OPDA which is the harmonic mean of the average accuracy on common and private class samples.

Implementation. We implement our method using Pytorch (Paszke et al. 2019) on a single Nvidia RTX A6000 GPU. Following previous works (Saito and Saenko 2021; Chen et al. 2022), we use ResNet50 (He et al. 2016) without last fully-connected layer as our feature extractor. A 256-dimensional bottleneck layer and prediction head h is successively added after the feature extractor. We use MocoV2

Method	Office-31	Office-Home	VisDA	DomainNet (150/50/145)						
	(10/10/11)	(10/5/50)	(6/3/3)	P→R	P→S	R→P	R→S	S→P	S→R	Avg
UAN	63.5	56.6	30.5	41.9	39.1	43.6	38.7	39.0	43.7	41.0
CMU	73.1	61.6	34.6	50.8	45.1	52.2	45.6	44.8	51.0	48.3
DANCE	82.3	63.9	42.8	55.7	47.0	51.1	46.4	47.9	55.7	50.6
DCC	80.2	70.2	43.0	56.9	43.7	50.3	43.3	44.9	56.2	49.2
OVANet	86.5	71.8	53.1	56.0	47.1	51.7	44.9	47.4	57.2	50.7
GATE	87.6	75.6	56.4	57.4	48.7	52.8	47.6	49.5	56.3	52.1
PPOT	**90.4**	**77.1**	**73.8**	**67.8**	**50.2**	**60.1**	**48.9**	**52.8**	**65.4**	**57.5**

Table 1: H-score (%) comparison on Office-31, Office-Home, VisDA and DomainNet for OPDA. Note that we only report the average H-score over all tasks on Office-31 on Office-Home, and the results for different tasks are in Appendix.

Method	Type	Office-Home (25/40/0)	VisDA (6/6/0)
PADA	P	62.1	53.5
IWAN	P	63.6	48.6
ETN	P	70.5	59.8
AR	P	**79.4**	**88.8**
DCC	U	70.9	72.4
GATE	U	73.9	75.6
PPOT	U	**74.3**	**83.0**

Table 2: Comparison of H-score (%) on Office-Home and VisDA for PDA setting. "P" and "U" denote PDA and UniDA methods, respectively with and without assuming the target label set is a subset of the source label set.

Method	Type	Office-Home (25/0/40)	VisDA (6/0/6)
STA	O	61.1	64.1
OSBP	O	64.7	52.3
ROS	O	**66.2**	**66.5**
DCC	U	61.7	59.6
OVANet	U	64.0	66.1
GATE	U	69.1	70.8
PPOT	U	**70.0**	**72.3**

Table 3: Comparison of H-score (%) on Office-Home and VisDA for OSDA setting. "O" and "U" denote OSDA and UniDA methods, respectively with and without assuming the source label set is a subset of the target label set.

(Chen et al. 2020) to contrastive pre-train our feature extractor, the number of epochs in pre-training is 100, batch size is 256, and learning rate is 0.03.

In training phase, we optimize the model using Nesterov momentum SGD with momentum of 0.9 and weight decay of 5×10^{-4}. Following (Ganin and Lempitsky 2015), the learning rate decays with the factor of $(1 + \alpha t)^{-\beta}$, where t linearly changes from 0 to 1 in training, and we set $\alpha = 10, \beta = 0.75$. The batch size is set to 72 in all experiments except in DomainNet tasks where it is changed to 256. We train our model for 5 epochs (1000 iterations per epoch), and update source prototypes and α totally before every epoch. The initial learning rate is set to 1×10^{-4} on Office-31, 5×10^{-4} on Office-Home and VisDA, and 0.01 on DomainNet.

5.1 Results and Comparisons

We compare our method with four PDA methods (PADA (Cao et al. 2018b), IWAN (Zhang et al. 2018), ETN (Cao et al. 2019), AR (Gu et al. 2021)), three OSDA methods (OSBP (Saito et al. 2018), STA (Liu et al. 2019), ROS (Bucci, Loghmani, and Tommasi 2020)) and six UniDA methods (UAN (You et al. 2019), CMU (Fu et al. 2020), DANCE (Saito et al. 2020), DCC (Li et al. 2021), OVANet (Saito and Saenko 2021), GATE (Chen et al. 2022)). All the compared methods use the same backbone as ours.

OPDA setting. Table 1 shows the results of our method. Our method outperforms baselines and achieves state-of-the-art results on all four datasets. On Office-31 and Office-Home datasets, our method surpasses all baselines on average. In larger datasets, VisDA and DomainNet, our method brings more than 17% improvement over previous methods on VisDA, and 5% on DomainNet. In general, these results show that our method is suitable in UniDA tasks, especially on larger and challenging datasets.

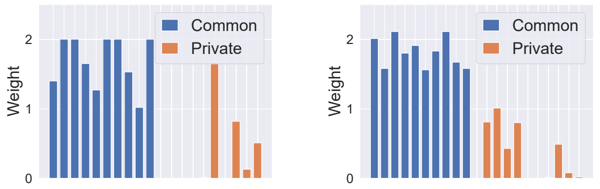
PDA and OSDA settings. Following (Li et al. 2021), we train our model without any prior knowledge of label space mismatch in PDA and OSDA settings. We report the results for PDA setting in Table 2. We can see that our method achieves better results than other UniDA-based methods (denoted as "U") on both datasets. The "P" denotes the PDA methods using prior knowledge that only the source domain has private classes. The results of OSDA setting are shown in Table 3, our method still surpasses all UniDA methods and OSDA methods (denoted as "O") using prior knowledge on label space mismatch on Office-Home and VisDA datasets.

5.2 Model Analysis

Comparison of m-PPOT with m-POT. To compare m-PPOT with m-POT (mini-batch based partial OT without using prototypes) in UniDA, we replace m-PPOT with m-POT in our method and use the average weight of samples in each class to replace the prototype weights $\boldsymbol{w}^s$ in Eqn. (13), and the corresponding method is denoted as "POT". As shown in Table 4, PPOT surpasses POT in all three datasets, confirming that m-PPOT performs better than m-POT in UniDA.

Method	Office-31	VisDA	Office-Home
POT	88.4	66.4	74.2
PPOT (w/o CL)	89.4	58.1	74.3
PPOT (w/o $\mathcal{L}_{pe}$)	88.4	71.1	76.5
PPOT (w/o $\mathcal{L}_{ne}$)	89.6	67.8	74.4
PPOT (w/o reweight)	86.5	69.9	74.7
PPOT	**90.4**	**73.8**	**77.1**

Table 4: Ablation study for OPDA on Office-31, Office-Home and VisDA. "CL" means contrastive pre-training.



(a) Source domain

(b) Target domain

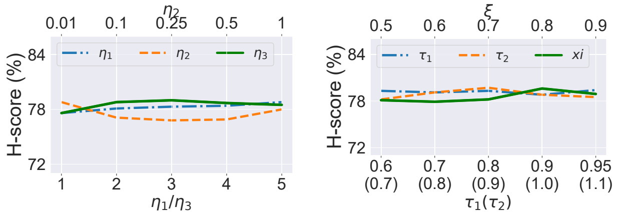
Figure 2: (a) Class weight $\boldsymbol{w}^s$ in Eqn. (13) on the source domain. (b) Average weight $\boldsymbol{w}^t$ in Eqn. (11) for each class on the target domain. Task: W→D on Office-31 for OPDA.

Effect of contrastive pre-training. To evaluate the effect of contrastive pre-training, the contrastive pre-training is removed and the feature extractor is replaced by a ResNet50 pre-trained on ImageNet. The results shown in Table 4 illustrate that performance degenerates in all experiments, especially in more challenging tasks such as VisDA. Note that without contrastive pre-training, our model still surpasses state-of-the-art methods on Office-31 and VisDA and reaches a comparable result on Office-Home.

Effectiveness of reweighted entropy loss. To evaluate this loss, we remove $\mathcal{L}_{pe}$ and $\mathcal{L}_{ne}$ in our model respectively. Table 4 shows that PPOT outperforms PPOT(w/o $\mathcal{L}_{pe}$) by 2% and PPOT(w/o $\mathcal{L}_{ne}$) by 6% on VisDA dataset.

Effectiveness of reweighting strategy. To evaluate the effectiveness of our reweighting strategy in Eqns. (13) and (11), we set $w_i^s = 1$ in Eqn. (13) and $w_j^t = 1$ in Eqn. (11) for any $0 \leqslant i \leqslant m$ and $0 \leqslant j \leqslant n$. The results of Table 4 show that PPOT(w/o reweight) decreases at least 3% more than PPOT in all datasets, which means that our reweighting strategy is important in our method. We further visualize the learned weights of source/target classes in UniDA task W→D on Office-31 datasets, as shown in Fig. 2. In both domains, most common classes have higher weights than private classes, which implies that our model can separate common and private classes effectively.

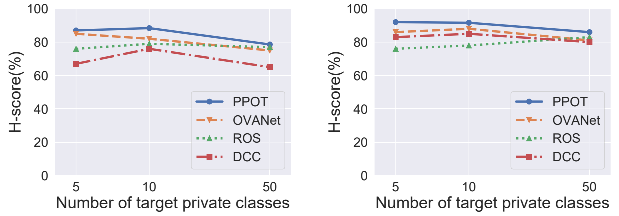
Sensitivity to hyper-parameters. Figure 3 evaluates the sensitivity of our model to hyper-parameters τ_1, τ_2, η_1, η_2, η_3, and ξ. Results show that our model is relatively stable to τ_1 and τ_2 at the range of $[0.6, 0.95]$ and $[0.7, 1.1]$ respectively, as shown in Fig. 3(b). In Fig. 3(b), we can also see that the setting of threshold ξ does not impact the performance much on our model in range $[0.5, 0.9]$. Furthermore, Fig. 3(a) shows that our model is relatively stable to varying values of η_1, η_2, and η_3.



(a)

(b)

Figure 3: Sensitivity to hyper-parameters (a) η_1, η_2 and η_3 in Eqn. (14), (b) τ_1, τ_2 in Eqn. (15) and threshold ξ. All results are for the OPDA setting in task C→A.



(a) A→P

(b) P→R

Figure 4: H-score curves of different methods with varying number of target private classes for OPDA tasks A→P and P→R.

H-score with varying number of target private classes. We evaluate our method with different numbers of target private classes. Results in A→P and P→R tasks are shown in Fig. 4, our method outperforms other baselines in all cases. It shows that our method is effective for OPDA with respect to different numbers of target domain private classes, and the performance marginally decreases with the increase of the number of target domain private classes.

6 Conclusion

In this paper, we propose to formulate the universal domain adaption (UniDA) as a partial optimal transport problem in deep learning framework. We propose a novel mini-batch based prototypical partial OT (m-PPOT) model for UniDA task, which is based on minimizing mini-batch prototypical partial optimal transport between two domain samples. We also introduce reweighting strategy based on the transport plan in UniDA. Experiments on four benchmarks show the effectiveness of our method for UniDA tasks including OPDA, PDA, OSDA settings. In the future, we plan to further theoretically analyze the mini-batch based m-PPOT, and apply it to more applications requiring partial alignments in deep learning framework.

Acknowledgments

This work was supported by National Key R&D Program 2021YFA1003002, NSFC (12125104, U20B2075, 11971373, 61721002, U1811461), and the Fundamental Research Funds for the Central Universities.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *ICML*.

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A theory of learning from different domains. *ML*, 79(1): 151–175.

Ben-David, S.; Blitzer, J.; Crammer, K.; and Pereira, F. 2006. Analysis of representations for domain adaptation. In *NeurIPS*.

Benamou, J.-D.; Carlier, G.; Cuturi, M.; Nenna, L.; and Peyré, G. 2015. Iterative Bregman projections for regularized transportation problems. *SISC*, 37(2): A1111–A1138.

Bucci, S.; Loghmani, M. R.; and Tommasi, T. 2020. On the effectiveness of image rotation for open set domain adaptation. In *ECCV*.

Caffarelli, L. A.; and McCann, R. J. 2010. Free boundaries in optimal transport and Monge-Ampere obstacle problems. *Annals of Mathematics*, 673–730.

Cao, Z.; Long, M.; Wang, J.; and Jordan, M. I. 2018a. Partial transfer learning with selective adversarial networks. In *CVPR*.

Cao, Z.; Ma, L.; Long, M.; and Wang, J. 2018b. Partial adversarial domain adaptation. In *ECCV*.

Cao, Z.; You, K.; Long, M.; Wang, J.; and Yang, Q. 2019. Learning to transfer examples for partial domain adaptation. In *CVPR*.

Chen, L.; Lou, Y.; He, J.; Bai, T.; and Deng, M. 2022. Geometric Anchor Correspondence Mining With Uncertainty Modeling for Universal Domain Adaptation. In *CVPR*.

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Courty, N.; Flamary, R.; Habrard, A.; and Rakotomamonjy, A. 2017. Joint distribution optimal transportation for domain adaptation. In *NeurIPS*.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*.

Damodaran, B. B.; Kellenberger, B.; Flamary, R.; Tuia, D.; and Courty, N. 2018. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *ECCV*.

Fatras, K.; Séjourné, T.; Flamary, R.; and Courty, N. 2021. Unbalanced minibatch optimal transport; applications to domain adaptation. In *ICML*.

Figalli, A. 2010. The optimal partial transport problem. *Archive for Rational Mechanics and Analysis*, 195(2): 533–560.

Flamary, R.; Courty, N.; Tuia, D.; and Rakotomamonjy, A. 2016. Optimal transport for domain adaptation. *TPAMI*, 1.

Fu, B.; Cao, Z.; Long, M.; and Wang, J. 2020. Learning to detect open classes for universal domain adaptation. In *ECCV*.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*.

Ge, Z.; Liu, S.; Li, Z.; Yoshie, O.; and Sun, J. 2021. Ota: Optimal transport assignment for object detection. In *CVPR*.

Gu, X.; Yu, X.; Sun, J.; Xu, Z.; et al. 2021. Adversarial Reweighting for Partial Domain Adaptation. In *NeurIPS*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Ho, N.; Nguyen, X.; Yurochkin, M.; Bui, H. H.; Huynh, V.; and Phung, D. 2017. Multilevel clustering via Wasserstein means. In *ICML*.

Kantorovitch, L. 1958. On the translocation of masses. *Management Science*, 5(1): 1–4.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NeurIPS*.

Li, G.; Kang, G.; Zhu, Y.; Wei, Y.; and Yang, Y. 2021. Domain consensus clustering for universal domain adaptation. In *CVPR*.

Liu, H.; Cao, Z.; Long, M.; Wang, J.; and Yang, Q. 2019. Separate to adapt: Open set domain adaptation via progressive separation. In *CVPR*.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. 2015. Learning transferable features with deep adaptation networks. In *ICML*.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018. Conditional adversarial domain adaptation. In *NeurIPS*.

Nguyen, K.; Nguyen, D.; Pham, T.; Ho, N.; et al. 2022. Improving mini-batch optimal transport via partial transportation. In *ICML*.

Pan, Y.; Yao, T.; Li, Y.; Wang, Y.; Ngo, C.-W.; and Mei, T. 2019. Transferrable prototypical networks for unsupervised domain adaptation. In *CVPR*.

Panareda Busto, P.; and Gall, J. 2017. Open set domain adaptation. In *ICCV*.

Papyan, V.; Han, X.; and Donoho, D. L. 2020. Prevalence of neural collapse during the terminal phase of deep learning training. *PNAS*, 117(40): 24652–24663.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment matching for multi-source domain adaptation. In *ICCV*.

Peng, X.; Usman, B.; Kaushik, N.; Hoffman, J.; Wang, D.; and Saenko, K. 2017. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*.

Peyré, G.; Cuturi, M.; et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6): 355–607.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *ECCV*.

Saito, K.; Kim, D.; Sclaroff, S.; and Saenko, K. 2020. Universal domain adaptation through self supervision. In *NeurIPS*.

Saito, K.; and Saenko, K. 2021. Ovanet: One-vs-all network for universal domain adaptation. In *ICCV*.

Saito, K.; Yamamoto, S.; Ushiku, Y.; and Harada, T. 2018. Open set domain adaptation by backpropagation. In *ECCV*.

Shen, K.; Jones, R. M.; Kumar, A.; Xie, S. M.; HaoChen, J. Z.; Ma, T.; and Liang, P. 2022. Connect, not collapse: Explaining contrastive learning for unsupervised domain adaptation. In *ICML*.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sun, B.; and Saenko, K. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep hashing network for unsupervised domain adaptation. In *CVPR*.

Villani, C. 2009. *Optimal transport: old and new*, volume 338. Springer.

Xie, S.; Zheng, Z.; Chen, L.; and Chen, C. 2018. Learning semantic representations for unsupervised domain adaptation. In *ICML*.

Xu, R.; Liu, P.; Zhang, Y.; Cai, F.; Wang, J.; Liang, S.; Ying, H.; and Yin, J. 2021. Joint partial optimal transport for open set domain adaptation. In *IJCAI*.

You, K.; Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2019. Universal domain adaptation. In *CVPR*.

Zhang, J.; Ding, Z.; Li, W.; and Ogunbona, P. 2018. Importance weighted adversarial nets for partial domain adaptation. In *CVPR*.

Zhang, Y.; Liu, T.; Long, M.; and Jordan, M. 2019. Bridging theory and algorithm for domain adaptation. In *ICML*.