International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

Source-free Domain Adaptation via Target Prediction Distribution Searching

Song Tang · An Chang · Fabian Zhang · Xiatian Zhu✉ · Mao Ye✉ · Changshui Zhang

Received: date / Accepted: date

Abstract Existing Source-Free Domain Adaptation (SFDA) methods typically adopt the feature distribution alignment paradigm via mining auxiliary information (*e.g.,* pseudo-labelling, source domain data generation). However, they are largely limited due to that the auxiliary information is usually error-prone whilst lacking effective error-mitigation mechanisms. To overcome this fundamental limitation, in this paper we propose a novel ***Target Prediction Distribution Searching*** (TPDS) paradigm. Theoretically, we prove that in case of sufficient small distribution shift, the domain transfer error could be well bounded. To satisfy this condition, we introduce a flow of *proxy distributions* that facilitates the bridging of typically large distribution shift from the source domain to the target domain. This results in a *progressive searching on the geodesic path* where adjacent proxy distributions are regularized to have small shift so that the overall errors can be minimized. To account for the sequential correlation between proxy distributions, we develop a new pairwise alignment with category consistency (PACon) algorithm for minimizing the adaptation errors. Specifically, a manifold geometry guided cross-distribution neighbour search is designed to detect the data pairs supporting the Wasserstein distance based shift measurement. Mutual information maximization is then adopted over these pairs for shift regularization. Extensive experiments on five challenging SFDA benchmarks show that our TPDS achieves new state-of-the-art performance. The code and datasets are available at https://github.com/tntek/TPDS.

Keywords Domain adaptation · Distribution searching · Geodesic path · Wasserstein distance · Pairwise alignment

1 Introduction

Due to the increasing demand for information security and privacy protection, data sharing across domains becomes less possible. Also, conventional unsupervised domain adaptation (UDA) setting is questioned recently in the term of necessity of access to the source domain (Chidlovskii et al., 2016; Lao et al., 2021; Tanwisuth et al., 2021). In this context, model transfer turns

Song Tang
Institute of Machine Intelligence (IMI), University of Shanghai for Science and Technology, Shanghai, China; Technical Aspects of Multimodal Systems (TAMS) Group, Department of Informatics, Universität Hamburg, Hamburg, Germany.
E-mail: song.tang@uni-hamburg.de

An Chang
Institute of Machine Intelligence (IMI), University of Shanghai for Science and Technology, Shanghai, China.
E-mail: Cchangan666@gmail.com

Fabian Zhang
Department of Information Technology and Electrical Engineering (D-ITET), Eidgenössische Technische Hochschule Zürich, Switzerland.
E-mail: fabzhang@ethz.ch

Xiatian Zhu
Surrey Institute for People-Centred Artificial Intelligence, and Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, Guildford, UK.
E-mail: xiatian.zhu@surrey.ac.uk

Mao Ye
School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China.
E-mail: cvlab.uestc@gmail.com

Changshui Zhang
Institute for Artificial Intelligence, Tsinghua University (THUAI); State Key Lab of Intelligent Technologies and Systems; Beijing National Research Center for Information Science and Technology (BNRist); Department of Automation, Tsinghua University Beijing, China.
E-mail: zcs@mail.tsinghua.edu.cn

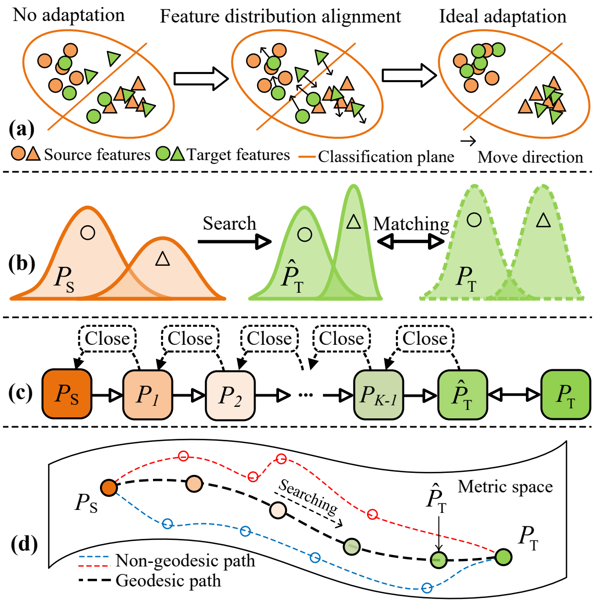

Fig. 1: Paradigm comparison: (a) Conventional feature distribution alignment *vs.* (b) our target prediction distribution ($\hat{P}_\mathrm{T}$) search for matching the ideal target distribution. This searching is driven by a progressive search strategy with error control: (c) we construct a flow of proxy distributions ($\{P_k\}_{k=1}^{K-1}$) with sufficiently small shift (*close*) in-between to connect the source distribution (P_S) and the ideal *unknown* target distribution (P_T). (d) Considering P_S and P_T as two different points in a metric space, in essence our method is searching along the geodesic path between the two points. Critically, it can be proven that by searching along this path subject to minimizing any adjacent proxy distributions in the flow, $\hat{P}_\mathrm{T}$ could closely match P_T well (Theorem 1).

out to be promising (Kim et al., 2021; Li et al., 2020; Liang et al., 2020). This is known as *Source-Free Domain Adaptation* (SFDA), which aims to adapt a pretrained model to a target scenario in an unsupervised manner without access to source domain training data.

Existing SFDA methods rely on mining auxiliary information in order to enable the adoption of well-established feature alignment algorithms (Long et al., 2018; Hoffman et al., 2018) following two strategies. The first one creates a fake source domain by generative models (Li et al., 2020) or by source hypothesis-based target data splitting (Du et al., 2021), and further aligns the pseudo source data and the target data in feature space like UDA. The second one performs self-supervised learning to transfer the source model to the target domain. In practice, the techniques of pseudo-labels (Liang et al., 2020), source prototypes (Tanwisuth et al., 2021), and target geometric information (Tang et al., 2022) are used to guide the self-supervised learning. Essentially, the two strategies perform a feature distribution alignment in an explicit way (the first one) or an implicit way (the second one). In Fig. 1 (a), we illustrate the distribution alignment process. The given source model gives a prediction for the source feature distribution marked in orange. The feature distribution alignment encourages the embedded target data (marked in green) to move/cluster toward the correct class cluster of the source feature distribution. Thus, the frozen classifier in the source model can correctly predict categories for the target data. However, aligning the feature distribution for SFDA is challenging at the absence of source domain training data and target domain labels. First of all, these auxiliary information is error-prone, suffering from further error propagation risk. Furthermore, this limitation would be easily amplified typical to existing SFDA methods due to lacking error mitigation.

To overcome the aforementioned foundational limitation, in this work a novel ***Target Prediction Distribution Searching*** (TPDS) paradigm is introduced. We reformulate the SFDA problem as searching the target prediction distribution, in contrast to conventional feature distribution alignment (Fig. 1 (b)). The target prediction distribution is formed as the model output of all the unlabeled training samples from the target domain. The key challenge is how to mitigate the misleading effect caused by the unknown errors of predicted label distributions. To tackle this obstacle, we search a proxy $\hat{P}_T$ under an *approximated* condition where the source and target domains share the same distribution. That is, the adaption error needs to be minimized. To achieve this, we introduce a progressive search strategy based on *a flow of proxy distributions* with adjacent ones being slightly shifted (Fig. 1 (c)). As a result, a typically large distribution gap from source to target domain can be gradually shifted away in multiple stages. Essentially, understand the P_S and P_T as two distinct points in a metric space, the searching induced by TPDS aims to find an optimal geodesic path in-between with minimal accumulative errors (Fig. 1 (d)). Critically, we prove theoretically that when the distribution shift on this path is sufficiently small, the transfer error across the two domains could be well bounded (Theorem 1).

We further instantiate a TPDS model in deep learning. Concretely, we split the whole training process evenly into multiple stages. Each stage corresponds to a single-step searching driven by aligning two adjacent proxy distributions in this flow. To that end, we design a new algorithm named Pairwise Alignment with Category Consistency (PACon). More specifically, manifold geometry guided credible sampling discovers the potential data pairs (*i.e.,* shift estimation), followed by mutual information maximization based optimization for shift reduction.

The contributions of this work are summarized as follows:

(1) We propose a novel TPDS paradigm for SFDA without high reliance on the accuracy of source domain auxiliary information. Critically, TPDS comes with theoretical guarantee on adaption error mitigation, which is largely lacking in previous feature distribution alignment based alternatives.
(2) To mitigate the cross-domain transfer error, we develop a new PACon method to align any two adjacent distributions in a flow. Unlike the popular shift measures such as MMD or KL-divergence, PACon encourages pairwise alignment with explicit geometric semantics intrinsic to adjacent distributions.
(3) We evaluate the proposed approach on five challenging domain adaptation benchmarks. The extensive experiments show that our TPDS yields new state-of-the-art results.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 details the proposed paradigm, followed by the model instantiation in Section 4. Section 5 presents the experimental results and analyses. Section 6 draws the conclusion in the end.

2 Related Work

2.1 Unsupervised Domain Adaptation

For UDA, the key is to reduce the domain drift. Since the source and target data are accessible during the transfer phase, probability matching becomes the main idea to solve this problem. Based on whether to use a deep learning algorithm, current work in UDA can be divided into two categories: 1) deep-learning-based and 2) non-deep-learning based. In the first category, researchers rely on techniques such as metric learning to reduce domain drift (Long et al., 2015, 2018; Pan et al., 2019). In these methods, an embedding space with unified probability distribution was learnt by minimizing certain statistical measures, *e.g.*, MMD (maximum mean discrepancy) (Tzeng et al., 2014), which were used to evaluate the discrepancy of the domains. In addition, adversarial learning has been another popular framework for its capability of aligning the probabilities of two different distributions (Hoffman et al., 2018; Zhang et al., 2019; Munro & Damen, 2020). The second category reduces the drift in diverse manners. For example, focusing on the energy, an energy distribution-based classifier was developed to detect the confidence target data (Tang et al., 2019). The structural knowledge (Xia et al., 2022), category contrast (Huang et al., 2022) and spectral information (Zhang et al., 2022) were exploited to boost the adaptation. In all the aforementioned methods, the source data is indispensable as labeled samples were used to explicitly formulate domain knowledge (*e.g.*, probability, structural structure, energy or spectral information). When the labeled data in the source domain are not available, these conventional UDA methods fail.

2.2 Source-Free Domain Adaptation

The current mainstream approach to SFDA follows the paradigm of feature distribution alignment. Existing methods can be generally divided into two classes. The *first* class performs explicitly feature alignment by converting SFDA to the conventional UDA problem (Li et al., 2020; Du et al., 2021; Tian et al., 2022). These methods reconstruct a fake source domain under some source hypothesis, and further align the target data to the pseudo source data in the feature space. The *second* class conducts alignment implicitly by adapting the source model to the target domain based on self-supervised learning. In the absence of source domain data, the source model is used to generate an auxiliary factor, such as hard samples (Li et al., 2022) or prototypes (Tanwisuth et al., 2021), to assist feature alignment. On the other hand, alternative methods mine the auxiliary information from the target domain data. Except for widely used pseudo-labels, e.g., clustering-based pseudo-labels generation (Liang et al., 2020), pseudo-labels denoising (Chen et al., 2021; Ahmed et al., 2022), geometry information like the intrinsic neighborhood structure (Yang et al., 2021) and data manifold (Tang et al., 2022) have also been exploited for guiding model adaptation. In contrast to all the previous methods, we introduce a novel target prediction distribution search paradigm conceptually different from feature distribution alignment.

2.3 Gradual Domain Adaptation

In transfer learning, the most relevant work with ours is Gradual Domain Adaptation (GDA) performing knowledge transfer in the time dimension (*e.g.,* years). In this setting, the variety dynamics is given and represented by a series of intermediate unlabeled domains between source domain and target domain. At the high level, Gradual self-training (GST) is the main strategy. There are two main research lines. The *first* line extends the GST framework to address a variety of GDA cases, such as the scenario without some intermediate domains (Abnar et al., 2021) or without the predefined intermediate domain index (Chen & Chao, 2021). The *second* line (Kumar et al., 2020; Wang et al., 2022) focus on understanding GDA with theoretical analysis. Our work differs significantly from GDA due to no intermediate domains, rendering previous GDA methods inapplicable.

2.4 Progressive Transfer in Domain Adaptation

Existing progressive transfer methods for domain adaptation can be split into three groups. The first group is *subspace-based*, assuming that the source and target domains are two points on a manifold. For reducing their domain gap, subspaces along the geodesic path are interpolated to connect the two points (Gopalan et al., 2011; Caseiro et al., 2015; Cui et al., 2014). The second group is *gradual learning-based* (e.g., curriculum learning (Roy et al., 2021), deep clustering (Liang et al., 2021)). In an epoch-wise training fashion, they use the previous epoch model to guide the current training epoch. The third group is *domain generation based*. The core idea is to generate a flow of intermediate smoothly-shifting domains capable of bridging the domain gap between the source and target domains (Gong et al., 2019). Our method belongs to this group. Importantly, we highlight the key novel designs in comparison: (1) We form the intermediate domains with simple yet reliable probability distributions; (2) We uniquely take into account error control to the progressive learning process; (3) Our formulation is tailored for source free domain adaption, without the need for accessing the source data as required in (Gong et al., 2019).

3 Methodology

In this section, we first formulates the SFDA problem, and then formalize target prediction distribution searching. Finally, we present the optimization analysis for a single-step searching in the matching process.

3.1 Source Data-free Domain Adaptation Formulation

Given two different but related domains, *i.e.,* the source domain S and target domain T. Let source $\mathcal{X}_s = \{\boldsymbol{x}_i^s\}_{i=1}^{n_s}$ and $\mathcal{Y}_s = \{y_i^s\}_{i=1}^{n_s}$ be the source samples and the corresponding labels. The target data and their labels are $\mathcal{X}_t = \{\boldsymbol{x}_i\}_{i=1}^{n}$ and $\mathcal{Y}_t = \{y_i\}_{i=1}^{n}$, respectively, in which n is the number of the target data. Both domains remain the same C-way classification task. In the SFDA setting, suppose a source model $\theta_s : \mathcal{X}_s \mapsto \mathcal{Y}_s$ is pre-learned by $(\mathcal{X}_s, \mathcal{Y}_s)$, we intend to learn a target model $\theta_t : \mathcal{X}_t \mapsto \mathcal{Y}_t$ through an adaptation to the target domain. During the transfer process, only the source model θ_s and the unlabeled target data $\mathcal{X}_t$ are available.

3.2 Target Prediction Distribution Searching

Unlike conventional feature distribution alignment, we reformulate the SFDA problem as searching the optimal target prediction distribution. We start with the initial prediction distribution P_{θ_s}, obtained by applying the source model θ_s on $\mathcal{X}_t$. The objective of our TPDS is to identify the ideal prediction distribution P_{T} (unknown), typically different significantly from P_{θ_s} (*i.e.,* large distribution shift/gap). We formulate this as a distribution optimization problem as:

$$\arg\min_{\Theta} \mathrm{D}\left(\hat{P}_{\mathrm{T}}, P_{\mathrm{T}}\right), \hat{P}_{\mathrm{T}} = \mathrm{SE}(P_{\theta_s}) \tag{1}$$

where $\hat{P}_{\mathrm{T}}$ specifies the estimated target prediction distribution, $\mathrm{SE}(\cdot)$ stands for the searching process starting with P_{θ_s}, $\mathrm{D}(\cdot,\cdot)$ measures the discrepancy of two distributions, and Θ refers to the parameters to be learned. Two key challenges for this optimization are that 1) P_{T} is unknown making it hard to optimize, and 2) domain gap between P_{θ_s} and P_{T} are large making one-step searching for $\hat{P}_{\mathrm{T}}$ hard to achieve good result.

To overcome both of the challenges above, inspired by the spirit of gradual adaptation (Liang et al., 2020; Abnar et al., 2021), we design a progressive search strategy. Specifically, from the distribution P_{θ_s} to P_{T}, we construct a proxy prediction distribution flow $P_{\theta_0} \to P_{\theta_1} \cdots \to P_{\theta_k} \to \cdots \to P_{\theta_K}$, where $P_{\theta_0} = P_{\theta_s}$ with $\theta_0 = \theta_s$, $P_{\theta_K} = \hat{P}_{\mathrm{T}}$ with $\theta_K = \theta_t$, θ_k represents the k-th intermediate model estimating the proxy distribution P_{θ_k}. Consider a metric space induced by measure D where the P_{θ_s} and P_{T} can be deemed as two distinct points. In this case, this proxy distribution flow connecting P_{θ_s} and P_{T} specifies the searching route. Clearly, there are a number of possible choices, but which one can lead to the best matching? Intuitively, the geodesic path is optimal due to that its accumulative domain shift is minimal. In fact, we can theoretically prove the rationality of choosing the geodesic path as follows.

First of all, we select a proper measure to quantify the distribution shift. Of note, the P_{θ_k} searching taking $P_{\theta_{k-1}}$ as the start has one key point that P_{θ_k} inherits from $P_{\theta_{k-1}}$, and its shift from $P_{\theta_{k-1}}$ is small enough. Namely, we can regard the geometric shape of P_{θ_k} as derived from $P_{\theta_{k-1}}$ by a slight geometric change. Under this context, this work does not adopt the popular MMD, but selects a Wasserstein distance as the shift measure D for two reasons. First, due to having inherent geometric meaning, in theory, Wasserstein distance is more reasonable than others when the two adjacent distributions have a certain geometric relation (Mueller & Jaakkola, 2015). Second, some work (Shen et al., 2018) have verified that Wasserstein distance is better than MMD in the domain adaptation problem. Considering that SFDA is a classification-orientated problem, we use a Wasserstein-infinity distance-based measure (Kumar et al., 2020), denoted by $D_w(\cdot,\cdot)$. For any adjacent proxy distributions, the measure of their shift for C-way classification is expressed as

$$\begin{aligned} &\mathrm{D}_w\left(P_{\theta_{k-1}}, P_{\theta_k}\right) = \max\{d_0, \ldots, d_c, \ldots d_{C-1}\}, \\ &d_c = W_\infty\left(P_{\theta_{k-1}}\left(X_1 \mid Y_1 = c\right), P_{\theta_k}\left(X_2 \mid Y_2 = c\right)\right), \end{aligned} \tag{2}$$

where $W_\infty(\cdot,\cdot)$ is the Wasserstein-infinity distance, random variables X_1 and X_2 stand for the samples satisfying $P_{\theta_{k-1}}$ and P_{θ_k} respectively, random variables Y_1 and Y_2 denote the category, the conditional distribution $P_{\theta_{k-1}}(X_1 \mid Y_1 = c)$ and $P_{\theta_k}(X_2 \mid Y_2 = c)$ are probability measures on the c-th category by the θ_{k-1} and θ_k respectively.

With the measure presented in Eq. (2) and the theoretical results in (Kumar et al., 2020), we derive the following Theorem for the transfer performance upper bound of our progressive searching. The proof is given in `Appendix A`.

Theorem 1 *Suppose the distributions in the proxy distribution flow $\{P_{\theta_k}\}_{k=0}^{K}$ satisfy no label shift (the C categories are fixed) and the data is bounded (the data is not too large: $\|\boldsymbol{x}_i\|_2^2 \leq \rho$, $\rho > 0$ for $1 \leq i \leq n$). Distribution shifts in this flow are $\mathit{\Pi} = \{\pi_k\}_{k=1}^{K}$ where π_k change gradually from 1 to K, and $\pi_m = \max(\mathit{\Pi})$. If the source model $\theta_s = \theta_0$ has low loss $\alpha_0 \geq \alpha^*$ on the source domain, then*

$$L\left(\theta_K, P_{\theta_K}\right) \leq \left(\frac{2}{1 - \pi_m R}\right)^{K+1} \left(\alpha_0 + O\left(\frac{1}{\sqrt{n}}\right)\right), \tag{3}$$

where $L\left(\theta_K, P_{\theta_K}\right)$ is the objective loss as learning θ_K for P_{θ_K} prediction, R stands for the regularization strength of θ_K, α^* is a given small loss, n is the size of the target dataset.

Since P_{T} in Eq. (1) is unknown, we cannot directly evaluate $\mathrm{D}(\hat{P}_{\mathrm{T}}, P_{\mathrm{T}})$. We instead analyze the objective loss for predicting P_{θ_K} on the target domain, *i.e.,* $L\left(\theta_K, P_{\theta_K}\right)$. In case of $L\left(\theta_K, P_{\theta_K}\right) = 0$, we arrive $\hat{P}_{\mathrm{T}} = P_{\mathrm{T}}$.

Remark: Theorem 1 suggests an insight that reducing the maximal distribution shift π_m of this flow can lower the empirical risk of the resulting distribution P_{θ_k} on the target domain. In practice, π_m is not determined until the end of search. To overcome this challenge, we propose to require all distribution shifts $\{\pi_k\}_{k=1}^{K}$ of this flow to be sufficiently small, so that minimizing the final empirical risk can be approximated. Under the geometry view in the metric space as discussed earlier, our design means that the searching should be along the geodesic path for best matching between $\hat{P}_{\mathrm{T}}$ and P_{T}.

Together with the aforementioned geometry principle, we realize the proposed learning strategy by transforming the original optimization (Eq. (1)) to the following K sub-problems:

$$\arg\min_{\theta_k} \mathrm{D}_w\left(P_{\theta_{k-1}}, P_{\theta_k}\right),\ k = 1, \ldots, K. \tag{4}$$

This defines a search process along the geodesic path, as shown in Fig. 1 (d). Specifically, the k-th sub-problem refers to a *single-step search* that computes the current distribution P_{θ_k}, given the previous distribution $P_{\theta_{k-1}}$ formed by model θ_{k-1}. The entire search process of TPDS yields a proxy distribution flow with sufficient small shift in-between.

3.3 Single-step Searching

As indicated by Eq. (4), a single-step searching is driven by aligning the adjacent distributions, namely minimizing the distance $\mathrm{D}_w\left(P_{\theta_{k-1}}, P_{\theta_k}\right)$. In practice, we do not adopt the original definition in Eq. (2) to estimate this distance. According to Eq. (2), we need to iteratively compute the Wasserstein-infinity distances of all C categories and take the maximal one as the distribution shift. However, in our context no accurate category information is available. To solve this problem, we propose to minimize all the distances $\{d_0, \ldots, d_c, \ldots d_{C-1}\}$, without the need for the category information. This leads to the following reformulation:

$$\begin{aligned} &d_k = W_\infty\left(P_{\theta_{k-1}}\left(X_1 | Y\right), P_{\theta_k}\left(X_2 | Y\right)\right), \\ &\mathit{with}\ Y = Y_1 = Y_2. \end{aligned} \tag{5}$$

This category-agnostic formula allows us to compute the distribution shift in an unsupervised manner, facilitating the subsequent analysis for shift reduction.

For d_k minimization, we have to figure out two issues: **(1)** *what kind of data support the d_k estimation*, and **(2)** *how to reduce d_k based on the found data*. For the first issue, according to Eq. (5), the answer is clear that d_w is associated with the data paired by the same category. Given these paired data, for the second issue, due to that d_k is a kind of Wasserstein distance that builds on a transport in point-to-point way, we can perform a pairwise alignment to mimic this point-to-point process. Obviously, this solution above is not practical for two reasons: (1) we cannot pair the data with the same category accurately due to the absence of real target

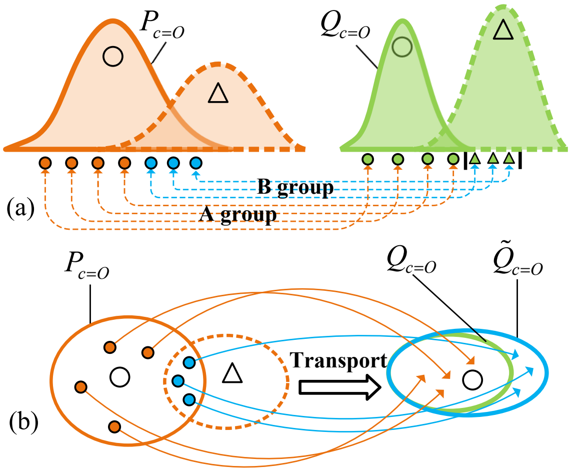

Fig. 2: The alignment from distribution $P_{c=\mathrm{O}}$ to $Q_{c=\mathrm{O}}$ in transport manner using the data pairs with neighbor relation. (a) Due to the neighbor constraint, the cross-distribution data pairs the can be divided into A group with the same category and B group with different but related category. (b) The transport over these data pairs encourage $P_{c=\mathrm{O}}$ align to a wider distribution $\widetilde{Q}_{c=\mathrm{O}}$ covering the real target distribution $Q_{c=\mathrm{O}}$. Clearly, this transport is work: the moving cannot change the structure of the multi-class distribution but only blurs the category boundary to some extent.

labels in the SFDA setting; and (2) how to encourage pairwise alignment is not clear.

To fix the first problem, we select data with neighbor relations in the feature space as the data pairs, denoted by $(\boldsymbol{x}_i, \boldsymbol{x}_i') \in (\mathcal{X}_t \times \mathcal{X}_t)$ for $1 \leq i \leq n$. Despite without the same category constraint, the alignment in transport manner still works. To explain it, we use Fig. 2 to illustrate this situation of aligning $P_{c=\mathrm{O}}$ to $Q_{c=\mathrm{O}}$ based on the data pairs with neighbor relation. Due to the feature closing, $\boldsymbol{x}_i$ and $\boldsymbol{x}_i'$ will have a similar but different distribution over all the categories. Thus, as shown in Fig. 2 (a), the data pairs only have two kinds, the ones sharing the same circle category, termed A group (connected with orange dotted line), and the ones with different but related two categories (their category distributions have an overlap), termed B group (connected with blue dotted line). As shown in Fig. 2 (b), the transport over A group can align $P_{c=\mathrm{O}}$ to $Q_{c=\mathrm{O}}$; the transport over B group broaden the final aligned distribution from $Q_{c=\mathrm{O}}$ (green oval) to $\widetilde{Q}_{c=\mathrm{O}}$ (blue oval). It is clear that this transport cannot change the structure of the multi-class distribution but only blurs the category boundary to some extent.

To fix the second problem, considering that the data with the same category are close to each other in feature space, we encourage the pairwise alignment by introducing a pairwise category consistency constraint on the pairs $\{(\boldsymbol{x}_i, \boldsymbol{x}_i')\}_{i=1}^{n}$. Here, this consistency is only confined to single data-pair, and different pairs may share different categories.

3.4 Overview of Training

In our TPDS paradigm, the proxy distribution flow is supposed to converge to the ideal target distribution P_{T} progressively. To this end, the adaptation training process is sliced into K successive stages $\{E_k\}_{k=1}^{K}$. In E_k, we perform a single-step searching for P_{θ_k} *w.r.t* $P_{\theta_{k-1}}$ via training the model from θ_{k-1} to θ_k.

4 Model Instantiation

As a showcase of our paradigm, we implement a TPDS instantiation in deep learning. Without generality loss, we take the search process of P_{θ_k} in stage E_k as an example in detail. Specifically, at the beginning of this stage, the model is initiated by θ_{k-1}, and a proxy distribution on the target domain is constructed. Next, we search for the optimal θ_k in an unsupervised learning manner, achieved by a ***pairwise alignment with category consistency*** (PACon) algorithm.

4.1 Model structure of θ_k

During the transfer process, all models, including the source model θ_s and all intermediate models $\{\theta_k\}_{k=1}^{K}$ predicting the proxy distributions, have the same composition. Specifically, θ_k consists of a feature extractor ϕ_k and a classifier υ_k with ending softmax operation, thus $\theta_k = \upsilon_k \circ \phi_k$ whilst $\theta_{k-1} = \upsilon_{k-1} \circ \phi_{k-1}$ where the operation $\circ$ means the function composition. In concrete implementation, we use two neural networks as the two modules: 1) a deep architecture is taken as the feature extractor, and 2) a four layer network is used as the classifier. The more details are given in `Implementation Details` of Experiments section.

4.2 Overview of PACon

Corresponding to the insight from the previous section, our PACon algorithm has two successive components: **(I)** *Distribution shift estimation*, and **(II)** *Distribution shift reduction*, as shown in Fig. 3. The first component is based on a credible sampling method for generating data pairs. Specifically, at the beginning of any epoch E_k, all

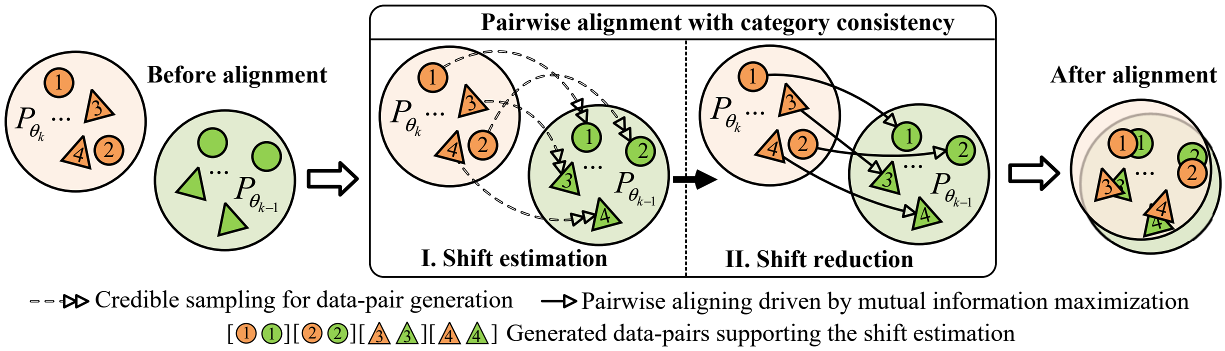
Before alignment
Pairwise alignment with category consistency
I. Shift estimation
II. Shift reduction
After alignment
Credible sampling for data-pair generation
Pairwise aligning driven by mutual information maximization
[① ①][② ②][3 3][4 4] Generated data-pairs supporting the shift estimation

Fig. 3: Overview of pairwise alignment from the distribution P_{θ_k} to distribution $P_{\theta_{k-1}}$. Two steps are involved. (1) **Shift estimation**: Generating sample pairs across the two distributions by chain-like search on the data manifold. (2) **Shift reduction**: Pulling the sample pairs found by maximizing the mutual information in the prediction space.

target samples are first embedded by the previous-epoch model θ_{k-1} to form a feature space for search. During E_k, given an input batch data, we extract their features using the up-to-date model θ_k and identify the paired samples in the search space with the chain-like search process (Fig. 4). The second component then aligns P_{θ_k} to $P_{\theta_{k-1}}$ by maximizing the mutual information entropy of those data pairs.

4.3 Distribution Shift Estimation

In the analysis above, we show that the distribution shift estimation is dependent on the data pairs with a neighbor relation. To account for the fact that the deep features locate on a data manifold, our credible sampling for data pairs detection builds the neighbor relation on the feature manifold by a chain-like searching. Furthermore, since the categories obtained in an unsupervised way are noisy, the sampled data forming the pairs are required to be credible, termed *credible neighbor.*

Suppose the data pairs constructed by the credible sampling are $\{(\boldsymbol{x}_i, \boldsymbol{x}'_i)\}_{i=1}^n \in \mathcal{X}_t$ where $\boldsymbol{x}_i$ (from P_{θ_k}) and $\boldsymbol{x}'_i$ (from $P_{\theta_{k-1}}$) are the input instance and its credible neighbor respectively. By model θ_k and θ_{k-1} respectively, $\boldsymbol{x}_i$ and all target data $\mathcal{X}_t$ are mapped into the feature space. To be concrete, the feature extractor ϕ_k (in θ_k) transforms $\boldsymbol{x}_i$ to feature $\boldsymbol{z}_i^\star$. The feature extractor ϕ_{k-1} (in θ_{k-1}) maps $\mathcal{X}_t$ into features $\mathcal{Z} = \{\boldsymbol{z}_i\}_{i=1}^n$ where $\boldsymbol{z}_i = \phi_{k-1}(\boldsymbol{x}_i)$ forming a data manifold. Then the classifier υ_{k-1} converts $\mathcal{Z}$ to probability vectors $\mathcal{P} = \{\boldsymbol{p}_i\}_{i=1}^n$ where $\boldsymbol{p}_i = \upsilon_{k-1}(\boldsymbol{z}_i)$. The data pair construction can be performed in the following two steps.

Step A: credible group construction. Firstly, we generate a group $\mathcal{G}_e$ using popular entropy-based ranking over $\mathcal{Z}$, like (Liu et al., 2021; Yang et al., 2020). With entropy computation, $\mathcal{P}$ converts to entropy set $\mathcal{H} = \{h_i\}_{i=1}^n$ where $h_i = -\sum \boldsymbol{p}_i \log \boldsymbol{p}_i$. Thus, $\mathcal{G}_e$ can be obtained by

$$\mathcal{G}_e = \{\boldsymbol{z}_i \mid \boldsymbol{z}_i \in \mathcal{Z},\ i \in \mathrm{topk}\left(\mathcal{H}, \sigma_e n\right)\}, \tag{6}$$

where σ_e is also a scaling factor.

We consider this entropy based strategy (*i.e.,* $\mathcal{G}_e$) is limited in the sense that there exists a many-to-one projection problem between the prediction distributions and the entropy values, leading to ambiguous selection. To mitigate this problem, we introduce another selection criterion based on class-aware feature geometrical structure with a particular stress on the most likely class prediction. This provides additional information to the entropy measurement, while being highly correlated and thus redundant. This is because the higher probability the most likely class receives, the lower entropy for the prediction distribution.

Specifically, to enhance the credibility further, we split off another group, $\mathcal{G}_o$, by clustering-based ranking. We obtain C cluster centers by a weighted k-means method formulated by Eq. (7) where $p_{i,c}$ is the c-th element of vector $\boldsymbol{p}_i$.

$$\boldsymbol{o}_c = \frac{\sum_{i=1}^n p_{i,c}\, \boldsymbol{z}_i}{\sum_{i=1}^n p_{i,c}},\ 0 \leq c \leq C-1. \tag{7}$$

Thus, the data credibility can be expressed by the minimum distance of the sample from the C cluster centers. Suppose the distances of $\boldsymbol{z}_i$ from $\{\boldsymbol{o}_c\}_{c=1}^C$ form vector $\boldsymbol{b}_i \in \mathbb{R}^C$. The c-th element of $\boldsymbol{b}_i$, standing for the distance from the c-th cluster center, equals $D_{cos}\left(\boldsymbol{z}_i, \boldsymbol{o}_c\right)$ where D_{cos} means the cosine-distance. Let $a_i = \min(\boldsymbol{b}_i)$ be $\boldsymbol{x}_i$'s minimum distance from the C centers, so that we get a measure set $\mathcal{A} = \{a_i\}_{i=1}^n$ over the target data. Thus, we can obtain $\mathcal{G}_o$ by

$$\mathcal{G}_o = \{\boldsymbol{z}_i \mid \boldsymbol{z}_i \in \mathcal{Z}, i \in \mathrm{topk}\left(\mathcal{A}, \sigma_o n\right)\}, \tag{8}$$

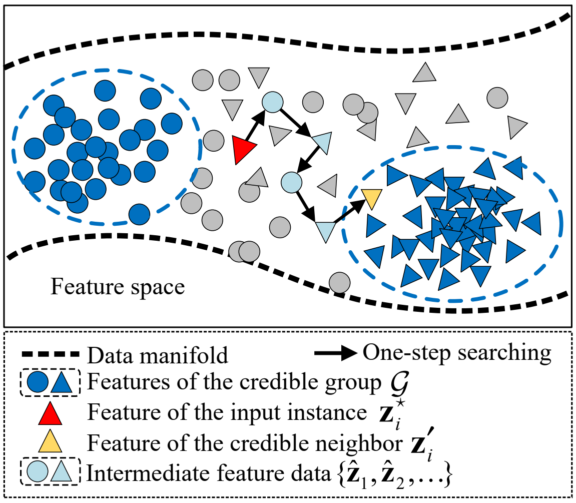
Feature space
Data manifold
One-step searching
Features of the credible group $\mathcal{G}$
Feature of the input instance $\mathbf{z}_i^\star$
Feature of the credible neighbor $\mathbf{z}_i'$
Intermediate feature data $\{\hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2, \ldots\}$

Fig. 4: **Illustration of credible sampling**. The feature vectors of all the target domain data extracted by model θ_{k-1} form a search space. Given the feature vector of a target domain sample extracted by model θ_k (red triangle), the objective is to identify the best match according to the manifold geometry of the search space.

where σ_o is a scaling factor, topk $(\mathcal{X}, k)$ selects the top k lowest elements from set $\mathcal{X}$ and returns their indexes.

Finally, we get the credible data group $\mathcal{G}$ by an intersection operation as $\mathcal{G} = \mathcal{G}_o \cap \mathcal{G}_e$. In Fig. 4, these feature data belonging to $\mathcal{G}$ are marked in blue.

Step B: data pair generation. To construct the data pairs, we find credible neighbor $\boldsymbol{x}_i'$ from $\mathcal{G}$ by a chain-like search, as illustrated in Fig. 4. Starting with the feature of the input instance $\boldsymbol{z}_i^\star$ (the red triangle), we carry out a one-step search to find its nearest neighbor, denoted by $\hat{\boldsymbol{z}}_1$, based on the cosine distance. If $\hat{\boldsymbol{z}}_1$ does not belong to $\mathcal{G}$, we conduct another one-step search for the new nearest neighbor $\hat{\boldsymbol{z}}_2$, taking $\hat{\boldsymbol{z}}_1$ as the start. Repeating this process, we build a search flow $\{\boldsymbol{z}_i^\star, \hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \ldots, \boldsymbol{z}'_i\}$ reaching $\mathcal{G}$. In this flow, the end element $\boldsymbol{z}'_i = \phi_{k-1}(\boldsymbol{x}_i')$ (the yellow triangle) is the feature of the credible neighbor that we are seeking whilst other elements (marked in light blue), except $\boldsymbol{z}_i^\star$, are the intermediate features. Note that to avoid the circle path on the flow, our search forgets its history. For example, when searching for $\hat{\boldsymbol{z}}_k$, the historical elements $\boldsymbol{z}_i^\star$ and $\{\hat{\boldsymbol{z}}_j\}_{j=1}^{k-1}$ are excluded from the similarity comparison.

4.4 Distribution Shift Reduction

In our instantiation, the distribution shift is reduced by pairwisely aligning these detected data pairs under the category consistency constraint. Inspired by the theoretical correlation of mutual information and pair-wise losses (Boudiaf et al., 2020), we use the following objective to reach the goal where $I(\cdot,\cdot)$ is the mutual information function (Paninski, 2003) whose computation is the same as the way in (Ji et al., 2019).

Algorithm 1 Overall training of TPDS

Input: The pre-trained source model θ_s, target data $\mathcal{X}_t$, max epochs K, max iterations in each epoch I_e.
Output: The target model $\theta_t = \theta_K$.

```
Let $\theta_0 = \theta_s$.
for $k = 1$ to $K$ do
    Initialize model $\theta_k$ by the trained $\theta_{k-1}$.
    Generate the credible group $\mathcal{G}$ (Eq. (6) and (7)).
    for $i = 1$ to $I_e$ do
        Sample a mini-batch from $\mathcal{X}_t$.
        Construct data pairs estimating the distribution shift for this mini-batch by Credible Sampling (Section 4.3).
        $\theta_k$ forward propagation (sample feed-forward).
        $\theta_k$ update through the distribution shift reduction (Eq. (11)).
    end for
end for
return: $\theta_K$.
```

$$L_{\mathrm{W}}(\theta_k) = \min_{\theta_k} \left[-I\left(\theta_k(\boldsymbol{x}_i), \theta_{k-1}(\boldsymbol{x}_i')\right) \right]. \tag{9}$$

Of note, L_W is sensitive to the target dataset scale. When the scale is small, since the limited data cannot describe the probability distribution well, the optimization effect based on the single regulator L_W is restricted. To overcome this problem, we introduce the diversity loss encouraging the balance of category prediction. This skill is widely adopted by the unsupervised approaches for multi-way classification (Jabi et al., 2019) to avoid the solution collapse (Ghasedi Dizaji et al., 2017) in which the model predicts all data as some specific categories. Suppose that θ_k transforms $\boldsymbol{x}_i$ to a probability vector $\boldsymbol{q}_i$, this loss is expressed as

$$L_{\mathrm{B}}(\theta_k) = \min_{\theta_k} \sum_{c=1}^{C} \mathrm{KL}\left(\bar{q}_c \| \varrho_c\right), \tag{10}$$

where KL$(\cdot||\cdot)$ means the KL-divergence loss function; $\varrho_{\{c=1,\cdots,C\}} = \frac{1}{C}$ is uniform distribution; $\bar{q}_c = \frac{1}{n}\sum_{i=1}^{n} q_{i,c}$ is empirical label distribution, in which $q_{i,c}$ is the probability of $\boldsymbol{x}_i$ in the c-th category. Combining with Eq. (9), we have the final objective:

$$L_{\mathrm{TPDS}}(\theta_k) = L_{\mathrm{W}}(\theta_k) + \beta_n L_{\mathrm{B}}(\theta_k), \tag{11}$$

where β_n trades off the two regularizations; its value is related to the dataset scale (represented by the target data number n): the smaller the dataset scale, the larger the β_n value (Its rationality is verified in `Experiments`). For clarity, we also summarize the overall training of TPDS to Alg. 1.

5 Experiments & Analyses

5.1 Data Sets

In this paper, we evaluate our method on five widely used benchmarks as follows.

Digits (Hoffman et al., 2018). As a typical dataset in UDA problems, we use the three most frequently used subsets under Digits, *i.e.*, SVHN (S), MNIST (M), and USPS (U). They contain the images of digits from 0 to 9 in different environments. We trained the method on three relatively challenging cross-domain tasks on the digit dataset, *i.e.*, S→M, U→M, and M→U.

Office-31 (Saenko et al., 2010). Office-31 is a small-scale dataset that is widely used in domain adaptation including three domains, *i.e.*, Amazon (A), Webcam (W), and Dslr (D), all of which are taken of real-world objects in various office environments. The dataset has 4,652 images of 31 categories in total. Images in A are online e-commerce pictures. W and D consist of low-resolution and high-resolution pictures, respectively.

Office-Home (Venkateswara et al., 2017). Office-Home is a medium-scale dataset that is mainly used for domain adaptation, all of which contains 15k images belonging to 65 categories from working or family environments. The dataset has four distinct domains, *i.e.*, Artistic images (Ar), Clip Art (Cl), Product images (Pr), and Real-word images (Rw).

VisDA (Peng et al., 2017). VisDA is a challenging large-scale dataset with 12 types of synthetic to real transfer recognition tasks. The source domain contains 152k synthetic images, while the target domain has 55k real object images from Microsoft COCO.

PACS (Li et al., 2017). PACS is an image dataset for domain generalization. It consists of four subdomains with 9.9k images sharing seven categories. The domains are Photo (P), Art Painting (A), Cartoon (C) and Sketch (S).

5.2 Implementation Details

Table 1: Classification accuracies (%) on the Digit dataset. SF means source data-free; the best accuracy for source-need task is highlighted with blue bold type, while we use red bold to emphasize the best result for source-free tasks.

Method	SF	S→M	U→M	M→U	Avg.
Source-model	–	73.3	90.2	78.8	80.8
ADDA (Tzeng et al., 2017)	✗	76.0	90.1	89.4	85.2
ADR (Saito et al., 2018)	✗	95.0	93.1	93.2	93.8
CyCADA (Hoffman et al., 2018)	✗	90.4	96.5	95.6	94.2
CDAN (Long et al., 2018)	✗	89.2	**98.0**	95.6	94.3
CAT (Deng et al., 2019)	✗	98.8	96.0	94.0	96.3
SWD (Lee et al., 2019)	✗	**98.9**	97.1	**98.1**	**98.0**
SHOT (Liang et al., 2020)	✓	**98.9**	97.5	97.9	98.1
TPDS (ours)	✓	**98.9**	**97.8**	**98.4**	**98.4**

Neural network architecture. We design and implement our network architecture based on Pytorch. We can divide the above datasets into two types: Digit recognition and Object recognition. For the digit recognition task, we use a variational LeNet as a feature extraction module, as done in (Liang et al., 2020). For the object recognition task, following the standard practice for fair comparison, we use neural networks including both the feature extractor ϕ_k and the classifier υ_k per model θ_k. The feature extractor ϕ_k contains a heavy-weight deep architecture, a batch-normalization layer and a full-connect layer with a size of 2048x256. As done in (Liang et al., 2020; Yang et al., 2021; Tang et al., 2022) for the deep architecture, we adopt ImageNet pretrained ResNet50 (He et al., 2016) on Office-31, Office-Home and PACS, and ResNet101 (He et al., 2016) on VisDA. For all datasets, the classifier υ_k takes the same structure as initially used in (Liang et al., 2020; Yang et al., 2020, 2021; Tang et al., 2022). The input layer is a fully-connected layer with batch normalization. The output layer is a fully-connected layer with weight normalization.

Source model θ_s training. For all evaluation datasets, θ_s was pretrained in the standard protocol (Liang et al., 2020; Tang et al., 2021; Yang et al., 2021). The adopt objective for training is given in `Appendix` B. We split the labelled source data into two parts of 90%:10% for model pretraining and validation. We set the training epochs on Digit, Office-31, Office-Home, PACS and VisDA to 30, 100, 50, 50 and 10, respectively.

TPDS training. We adopt the stochastic gradient descent (SGD) optimizer with a momentum of 0.9 and weight decay of 0.001. The learning rate is set to 0.01 for Office-31, Office-Home and PACS, 0.001 for VisDA. We train 15 epochs at a batch size of 64 on each target domain. TPDS has three hyperparameters: We set the scaling factors $(\sigma_o, \sigma_e) = (0.6, 0.5)$ for the credible group construction on all target domains, whilst $\beta_n = 1/0.5/0$ for small dataset (Office-31), medium dataset (Office-Home and PACS) and large dataset (VisDA) according to the dataset scale.

Table 2: Classification accuracies (%) on the Office-31 dataset based on ResNet50 backbone. SF means source data-free; the best accuracy for source-need task is highlighted with blue bold type, while we use red bold to emphasize the best result for source-free tasks.

Method	SF	A→D	A→W	D→A	D→W	W→A	W→D	Avg.
Source-model	–	80.5	75.2	60.5	94.3	63.1	98.2	78.6
CDAN (Long et al., 2018)	✗	92.9	94.1	71.0	98.6	69.3	**100.**	87.7
BSP (Chen et al., 2019)	✗	93.0	93.3	73.6	98.2	72.6	**100.**	88.5
TN (Wang et al., 2019)	✗	94.0	95.0	73.4	98.7	74.2	**100.**	89.3
DMRL (Wu et al., 2020)	✗	93.4	90.8	73.0	99.0	71.2	**100.**	87.9
IA (Jiang et al., 2020)	✗	92.1	90.3	75.3	98.7	74.9	99.8	88.8
MCC (Jin et al., 2020)	✗	95.6	95.4	72.6	98.6	73.9	**100.**	89.4
SRDC (Tang et al., 2020)	✗	**95.8**	**95.7**	76.7	**99.2**	**77.1**	**100.**	**90.8**
SUDA (Zhang et al., 2022)	✗	91.2	90.8	72.2	98.7	71.4	**100.**	87.4
CaCo (Huang et al., 2022)	✗	91.7	89.7	73.1	98.4	72.8	**100.**	87.6
MSGD (Xia et al., 2022)	✗	95.6	95.5	**77.3**	**99.2**	77.0	**100.**	**90.8**
SHOT (Liang et al., 2020)	✓	93.9	91.3	74.1	98.2	74.6	**100.**	88.7
BAIT (Yang et al., 2020)	✓	92.0	94.6	74.6	98.1	75.2	**100.**	89.1
3C-GAN (Li et al., 2020)	✓	92.7	93.7	75.3	98.5	**77.8**	99.8	89.6
SFDA (Kim et al., 2021)	✓	92.2	91.1	71.0	98.2	71.2	99.5	87.2
PCT (Tanwisuth et al., 2021)	✓	–	–	–	–	–	–	88.4
GKD (Tang et al., 2021)	✓	94.6	91.6	75.1	98.7	75.1	**100.**	89.2
NRC (Yang et al., 2021)	✓	96.0	90.8	75.3	99.0	75.0	**100.**	89.4
HMI (Lao et al., 2021)	✓	94.4	94.0	73.7	98.9	75.9	99.8	89.5
CPGA (Qiu et al., 2021)	✓	94.4	94.1	76.0	98.4	76.6	98.4	89.9
A2Net (Xia et al., 2021)	✓	94.5	94.0	**76.7**	**99.2**	76.1	**100.**	90.0
U-SFAN+ (Roy et al., 2022)	✓	94.2	92.8	74.6	98.0	74.4	99.0	88.8
VDM (Tian et al., 2022)	✓	94.1	93.2	75.8	98.0	77.1	**100.**	89.7
AAA (Li et al., 2022)	✓	95.6	94.2	75.6	98.1	76.0	99.8	89.9
TPDS (Ours)	✓	**97.1**	**94.5**	75.7	98.7	75.5	99.8	**90.2**

Table 3: Classification accuracies (%) on the Office-Home dataset based on ResNet50 backbone. SF means source data-free; the best accuracy for source-need task is highlighted with blue bold type, while we use red bold to emphasize the best result for source-free tasks.

Method	SF	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg.
Source-model	–	43.7	66.2	73.6	51.4	60.6	64.2	52.7	40.2	73.2	65.8	45.5	78.3	59.6
CDAN (Long et al., 2018)	✗	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
BSP (Chen et al., 2019)	✗	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	59.3	81.9	66.3
SAFN (Xu et al., 2019)	✗	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
TN (Wang et al., 2019)	✗	50.2	71.4	77.4	59.3	72.7	73.1	61.0	53.1	79.5	71.9	59.0	82.9	67.6
IA (Jiang et al., 2020)	✗	56.0	**77.9**	79.2	64.4	73.1	74.4	64.2	54.2	79.9	71.2	58.1	83.1	69.5
SRDC (Tang et al., 2020)	✗	52.3	76.3	**81.0**	69.5	**76.2**	**78.0**	68.7	53.8	81.7	**76.3**	57.1	**85.0**	71.3
TCM (Yue et al., 2021)	✗	58.6	74.4	79.6	64.5	74.0	75.1	64.6	56.2	80.9	74.6	60.7	84.7	70.7
MSGD (Xia et al., 2022)	✗	**58.7**	76.9	78.9	**70.1**	**76.2**	76.6	**69.0**	**57.2**	**82.3**	74.9	**62.7**	84.5	**72.4**
3C-GAN (Li et al., 2020)	✓	57.4	77.5	**82.4**	68.0	77.2	75.1	67.1	55.5	81.9	73.9	59.5	84.2	71.6
BAIT (Yang et al., 2020)	✓	57.4	77.5	**82.4**	68.0	77.2	75.1	67.1	55.5	81.9	73.9	59.5	84.2	71.6
SHOT (Liang et al., 2020)	✓	56.6	78.0	80.6	68.4	78.1	79.4	68.0	54.3	82.2	74.3	58.7	84.5	71.8
SFDA (Kim et al., 2021)	✓	48.4	73.4	76.9	64.3	69.8	71.7	62.7	45.3	76.6	69.8	50.5	79.0	65.7
PCT (Tanwisuth et al., 2021)	✓	–	–	–	–	–	–	–	–	–	–	–	–	71.0
CPGA (Qiu et al., 2021)	✓	**59.3**	78.1	79.8	65.4	75.5	76.4	65.7	**58.0**	81.0	72.0	**64.4**	83.3	71.6
HMI (Lao et al., 2021)	✓	57.8	76.7	81.9	67.1	78.8	78.8	66.6	55.5	82.4	73.6	59.7	84.0	71.9
GKD (Tang et al., 2021)	✓	56.5	78.2	81.8	68.7	78.9	79.1	67.6	54.8	82.6	74.4	58.5	84.8	72.1
PS (Du et al., 2021)	✓	57.8	77.3	81.2	68.4	76.9	78.1	67.8	57.3	82.1	**75.2**	59.1	83.4	72.1
NRC (Yang et al., 2021)	✓	57.7	**80.3**	82.0	68.1	**79.8**	78.6	65.3	56.4	**83.0**	71.0	58.6	**85.6**	72.2
A2Net (Xia et al., 2021)	✓	58.4	79.0	82.4	67.5	79.3	78.9	68.0	56.2	82.9	74.1	60.5	85.0	72.8
VDM (Tian et al., 2022)	✓	**59.3**	75.3	78.3	67.6	76.0	75.9	68.8	57.7	79.6	74.0	61.1	83.6	71.4
AAA (Li et al., 2022)	✓	56.7	78.3	82.1	66.4	78.5	79.4	67.6	53.5	81.6	74.5	58.4	84.1	71.8
U-SFAN+ (Roy et al., 2022)	✓	57.8	77.8	81.6	67.9	77.3	79.2	67.2	54.7	81.2	73.3	60.3	83.9	71.9
TPDS (Ours)	✓	**59.3**	**80.3**	82.1	**70.6**	79.4	**80.9**	**69.8**	56.8	82.1	74.5	61.2	85.3	**73.5**

5.3 Competitors

To verify the effectiveness of our method, we select 35 comparison methods, which can be divided into following two groups according to whether access to the source data during the transfer phase.

(1) 19 state-of-the-art vanilla domain adaptation methods, all requiring source and target data at the same time to solve the domain shift. They are ADDA (Tzeng et al., 2017), ADR (Saito et al., 2018), CDAN (Long et al., 2018), CyCADA (Hoffman et al., 2018), SWD (Lee et al., 2019), CAT (Deng et al., 2019),

Table 4: Classification accuracies (%) on the VisDA dataset based on ResNet101 backbone. SF means source data-free; the best accuracy for source-need task is highlighted with blue bold type, while we use red bold to emphasize the best result for source-free tasks.

Method	SF	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Perclass
Source-model	–	52.1	18.3	33.4	74.5	59.8	4.1	76.7	13.9	65.6	38.8	85.5	5.4	44.0
CDAN (Long et al., 2018)	✗	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
BSP (Chen et al., 2019)	✗	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
SAFN (Xu et al., 2019)	✗	93.6	61.3	84.1	70.6	94.1	79.0	**91.8**	79.6	89.9	55.6	**89.0**	24.4	76.1
DMRL (Wu et al., 2020)	✗	–	–	–	–	–	–	–	–	–	–	–	–	75.5
IA (Jiang et al., 2020)	✗	–	–	–	–	–	–	–	–	–	–	–	–	75.8
MCC (Jin et al., 2020)	✗	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
CGDM (Du et al., 2021)	✗	93.4	82.7	73.2	68.4	92.9	**94.5**	88.7	**82.1**	93.4	82.5	86.8	49.2	82.3
STAR (Lu et al., 2020)	✗	95.0	**84.0**	**84.4**	**73.0**	91.6	91.8	85.9	78.4	94.4	84.7	87.0	42.2	82.7
SUDA (Zhang et al., 2022)	✗	88.3	79.3	66.2	64.7	87.4	80.1	85.9	78.3	86.3	87.5	78.8	**74.5**	79.8
CaCo (Huang et al., 2022)	✗	90.4	80.7	78.8	57.0	88.9	87.0	81.3	79.4	88.7	88.1	86.8	63.9	80.9
MSGD (Xia et al., 2022)	✗	**97.5**	83.4	84.4	69.4	**95.9**	94.1	90.9	75.5	**95.5**	**94.6**	88.1	44.9	**84.6**
3C-GAN (Li et al., 2020)	✓	94.8	73.4	68.8	74.8	93.1	95.4	88.6	84.7	89.1	84.7	83.5	48.1	81.6
HMI (Lao et al., 2021)	✓	–	–	–	–	–	–	–	–	–	–	–	–	82.4
SHOT (Liang et al., 2020)	✓	95.0	87.5	81.0	57.6	93.9	94.1	79.3	80.5	90.9	89.8	85.9	57.4	82.7
BAIT (Yang et al., 2020)	✓	93.7	83.2	84.5	65.0	92.9	95.4	88.1	80.8	90.0	89.0	84.0	45.3	82.7
SFDA (Kim et al., 2021)	✓	86.9	81.7	84.6	63.9	93.1	91.4	86.6	71.9	84.5	58.2	74.5	42.7	76.7
GKD (Tang et al., 2021)	✓	95.3	87.6	81.7	58.1	93.9	94.0	80.0	80.0	91.2	91.0	86.9	56.1	83.0
CPGA (Qiu et al., 2021)	✓	94.8	83.6	79.7	65.1	92.5	94.7	90.1	82.4	88.8	88.0	88.9	60.1	84.1
PS (Du et al., 2021)	✓	95.3	86.2	82.3	61.6	93.3	95.7	86.7	80.4	91.6	90.9	86.0	59.5	84.1
A2Net (Xia et al., 2021)	✓	94.0	87.8	85.6	66.8	93.7	95.1	85.8	81.2	91.6	88.2	86.5	56.0	84.3
NRC (Yang et al., 2021)	✓	96.8	91.3	82.4	62.4	96.2	95.9	86.1	80.6	94.8	**94.1**	90.4	59.7	85.9
U-SFAN+ (Roy et al., 2022)	✓	94.9	87.4	78.0	56.4	93.8	95.1	80.5	79.9	90.1	90.1	85.3	60.4	82.7
AAA (Li et al., 2022)	✓	94.4	85.9	74.9	60.2	96.0	93.5	87.8	80.8	90.2	92.0	86.6	68.3	84.2
NEL (Ahmed et al., 2022)	✓	94.5	60.8	92.3	87.3	87.3	93.2	87.6	**91.1**	56.9	83.4	93.7	**86.6**	84.2
VDM (Tian et al., 2022)	✓	96.9	89.1	79.1	66.5	95.7	**96.8**	85.4	83.3	**96.0**	86.6	89.5	56.3	85.1
TPDS (Ours)	✓	**97.6**	**91.5**	**89.7**	**83.4**	**97.5**	96.3	**92.2**	82.4	**96.0**	**94.1**	**90.9**	40.4	**87.6**

Table 5: Classification accuracies (%) on the PACS dataset based on ResNet50 backbone. SF means source data-free; the best accuracy for source-free task is highlighted with red bold type.

Method	SF	A→C	A→P	A→S	C→A	C→P	C→S	P→A	P→C	P→S	S→A	S→C	S→P	Avg.
Source-model	–	55.0	97.6	44.6	61.4	84.2	46.7	68.0	29.9	31.2	20.8	38.4	33.6	50.9
SHOT (Liang et al., 2020)	✓	86.3	**99.5**	39.7	92.5	98.7	71.1	89.2	80.9	50.3	86.0	90.5	99.2	82.0
GKD (Tang et al., 2021)	✓	87.0	**99.5**	48.1	93.4	98.9	**75.7**	90.7	84.3	55.5	91.2	90.6	99.1	84.4
TPDS (Ours)	✓	**89.5**	99.3	**68.9**	**93.2**	**99.0**	73.1	**93.1**	**84.5**	**70.4**	**94.1**	**93.2**	**99.3**	**88.1**

BSP (Chen et al., 2019), TN (Wang et al., 2019), SAFN (Xu et al., 2019), IA (Jiang et al., 2020), DMRL (Wu et al., 2020), STAR (Lu et al., 2020), MCC (Jin et al., 2020), CGDM (Du et al., 2021), TCM (Yue et al., 2021), SRDC (Tang et al., 2020), SUDA (Zhang et al., 2022), CaCo (Huang et al., 2022) and MSGD (Xia et al., 2022).

(2) 15 current state-of-the-art SFDA models, such as SFDA (Kim et al., 2021), 3C-GAN (Li et al., 2020), SHOT (Liang et al., 2020), BAIT (Yang et al., 2020), HMI (Lao et al., 2021), PCT (Tanwisuth et al., 2021), CPGA (Qiu et al., 2021), AAA (Li et al., 2022), PS (Du et al., 2021), GKD (Tang et al., 2021), A2Net (Xia et al., 2021), NRC (Yang et al., 2021), VDM (Tian et al., 2022), NEL (Ahmed et al., 2022) and U-SFAN+ (Roy et al., 2022). Among them, method 3C-GAN, PS, AAA, A2Net and VDM are based on the pseudo source domain generation or construction, whilst the rest methods are based on the framework of self-supervised learning.

5.4 Comparative Results

Digit recognition. As reported in Tab. 1, TPDS obtains the best results on the all tasks compared with SHOT and has a 0.3% increase in average accuracy. Compared with these UDA work, TPDS achieves the highest performances on 2 out of 3 tasks, except for the transfer task U→M, surpassing the best method SWD by 0.4% in average accuracy.

Object recognition. Tab. 2∼5 present the quantitative results on the four datasets. On Office-31 (Tab. 2), TPDS obtains best results on two transfer tasks among these SFDA methods, A→D and A→W, leading to the 90.2% average accuracy, which increases by 0.2% compared to the second-best method A2Net (90.0%). On Office-Home (Tab. 3), TPDS defeats other methods in 5 out of 12 tasks. Compared with the previous best method A2Net (72.8%), TPDS improves by 0.7% average accuracy and reaches 73.5%. On VisDA (Tab. 4), besides two categories, person and truck, TPDS achieves

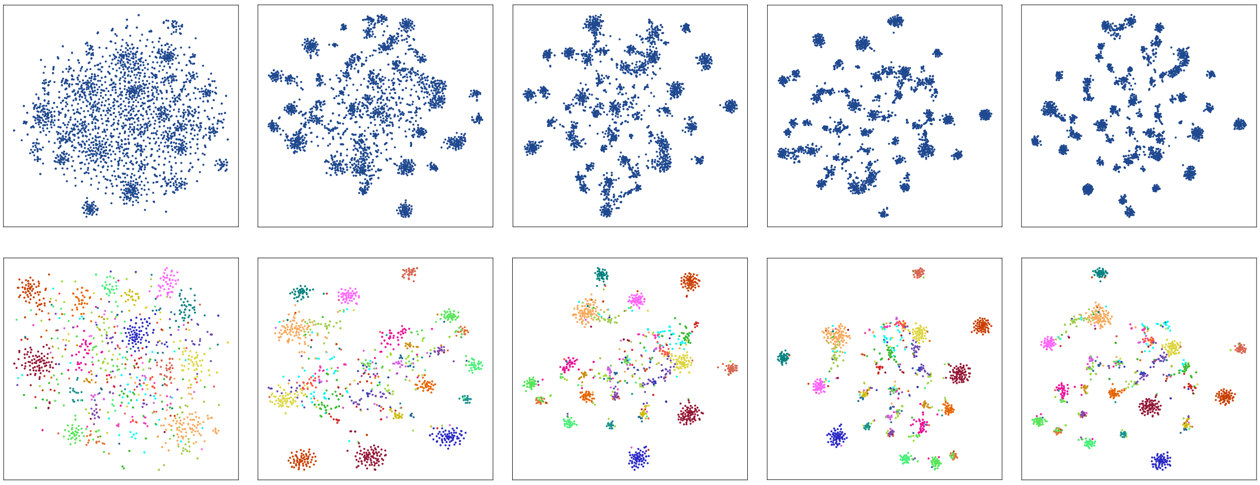

Fig. 5: t-SNE feature visualization on transfer task Cl → Ar in Office-Home. **Top:** The first and end sub-figures presents the feature distribution embedded by the source model θ_s (before adaptation) and the adapted model θ_t (after adaptation), respectively; from left to right the rest three ones sample the feature distribution evaluation in the 2, 5 and 10 epoch. **Bottom:** The aggregation details with category information is shown correspondingly; for clarity, we select the first 30 classes from the 65 categories in total, where different colors stand for different classes.

the best performance. In average accuracy, TPDS obtain 87.6% and surpasses the previous best method NRC by 1.7%. On PACS (Tab. 5), TPDS obtains best results on all except for two transfer tasks with small gap, A→P and C→S. Especially, TPDS improves by 20.8% and 15.1% on transfer task A→S and task P→S respectively compared with the second-best method GKD. As a result, TPDS defeats SHOT by a margin of 3.7% in average accuracy.

Besides, compared with these conventional UDA methods needing the access to the source data, TPDS is also competitive on the three object recognition datasets, as shown in Tab. 2∼4, despite without the facilitating from source data during the adaptation phase. Specifically, on the Office-31 dataset, TPDS has a gap of 0.6% compared with the best UDA method MSGD. However, with the increase of target data, the advantage of TPDS grows further, surpassing the best method MSGD by 1.1% and 3.0% respectively on the Office-Home and VisDA datasets. To sum up, these comparison results on the five datasets mentioned above confirm the state-of-the-art performance of TPDS.

5.5 Further Analyses

Feature visualization. Using the widely used visualization tool t-SNE (Van der Maaten & Hinton, 2008), we conduct a feature visualization experiment based on the 65-way classification results of transfer task Cl→Ar in the Office-Home dataset. Fig. 5 presents the visualization results. As shown in the top, before adaptation, the intertwined features, embedded by the source model θ_s, distribute without apparent aggregation (the first sub-figure); after adaptation, the features aggregate evidently (the ending sub-figure). From left to right, the three sub-figure show that the features gradually cluster during the adaptation phase. For clear observation, we select the first 30 from total of 65 categories to present the clustering details. As shown in the bottom, where different colors stand for different categories, the aggregation is performed with category meaning. The visualization results show that TPDS can predict a probability distribution with category meaning.

Hyperparameter sensitivity. To validate the sensitivity of σ_o and σ_e, we conduct 30 experiments as $\sigma_o \in [0.3, 0.8]$ and $\sigma_o \in [0.3, 0.7]$ based on task Cl→Ar in the Office-Home dataset. As shown on the left of Fig. 6, the accuracy does not change drastically. Thus, TPDS's performance is robust to σ_o, σ_e. As for the sensitivity of β_n, we conduct 6 experiments as β_n varies from 0 to 1.0 on each testing dataset. As shown in the middle of Fig. 6, as the dataset size increases, Office-31 → Office-Home → VisDA, the best result occurs when $\beta_n = 1.0/0.6/0.0$, respectively. This phenomenon is consistent with our expectation. As mentioned above, when the amount of data is not enough to describe the distribution, L_{B} can boost the L_{W}-only-based transfer. Conversely, L_{B} will deteriorate the performance. These results confirm the rationality of our setting β_n to be related to the dataset size.

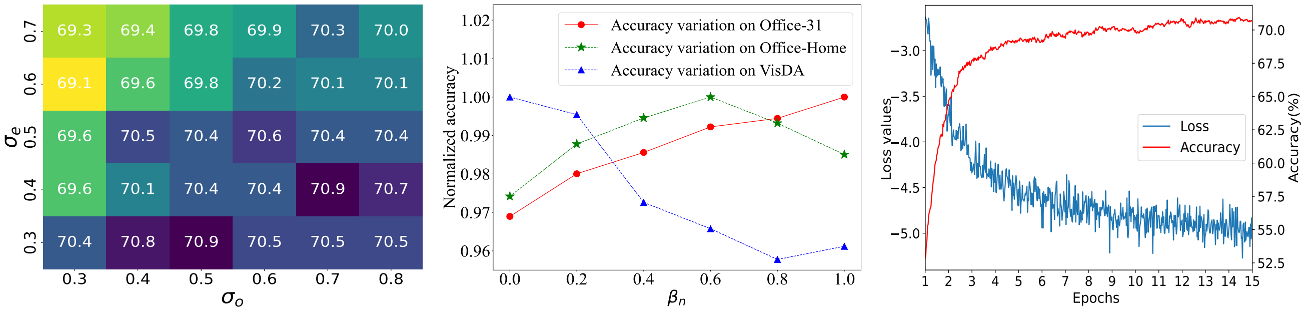

Fig. 6: **Left:** Accuracy variation on task Cl → Ar in the Office-Home dataset as σ_o and σ_e varying. **Middle:** Accuracy variation curves on the three evaluation datasets as β_n varying. For a clear view, all results on each dataset are normalized by the best accuracy on this dataset. **Right:** Accuracy and loss variation curve during training on task Cl → Ar in the Office-Home dataset.

Table 6: Effect of MixMatch data augmentation.

Method	Office-31	Office-Home
Source-model	78.6	59.6
TPDS	90.2	73.5
TPDS + MixMatch	**90.5**	**74.1**

Table 7: Training efficiency evaluation. Metric: Average training time in seconds of each transfer task.

Method	Office-31	Office-Home
SHOT (Liang et al., 2020)	181.1	602.1
TPDS	246.3	822.7

Table 8: Evaluating the chain-like search steps needed during training.

Dataset	Office-31	Office-Home
Average-steps	1.7	1.5

Training stability. On the right of Fig. 6, we present the training stability of TPDS on task Cl→Ar in the Office-Home dataset. As the training goes from epoch 1 to 15, the accuracy rapidly climbs at the early epochs (from epoch 1 to 4) and converges to the maximum through a slow increase with small vibrations (from epoch 4 to 13). The loss value of L_{TPDS} over all epochs gradually decreases, whose trend is consistent with the performance change. The phenomenon indicates that the training of TPDS is stable and reliable.

Extentability. In the spirit of SHOT++ (Liang et al., 2021) that leverages additional training components (e.g., MixMatch for data augmentation (Berthelot et al., 2019)) on top of SHOT, we carry out a test for model extentability where our TPDS is further enhanced by MixMatch, termed TPDS+MixMatch. As reported in Tab. 6, TPDS+MixMatch obtains better results compared with the original version, which confirms the extentability of our TPDS.

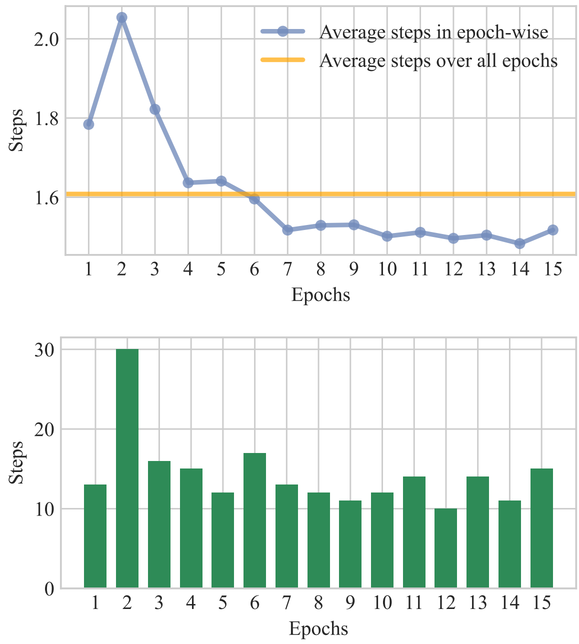

Fig. 7: Chain-like search steps on task Ar→Cl in Office-Home. **Top:** The average steps in epoch view where the red line stands for the average step over all epochs. **Bottom:** The maximal steps of each epoch.

Computational cost. We compare our method with SHOT in terms of the average training time. The results in Tab. 7 show that our model training is some

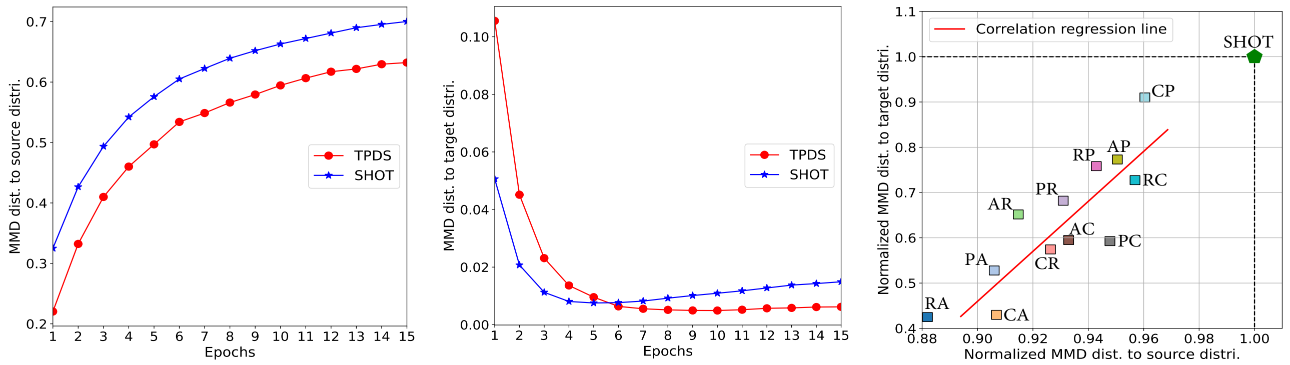

Fig. 8: Distribution shift analysis using MMD distance on the Office-Home dataset. **Left and Middle:** The MMD distance variation to the source distribution and the target distribution, respectively, during the training on transfer task Cl→Ar. **Right:** The further distribution shift analysis on all 12 transfer tasks. A, C, P and R are short for domain Ar, Cl, Pr and Rw, respectively; PR means Pr→Rw. Different from the analysis on Cl→Ar using the intermediate models in adaptation process, these distances are computed by the final trained model.

slower due to the gradual adaptation nature, which is a reasonable cost for better adaptation performance.

Convergence of chain-like search. We evaluate the convergence of our chain-like search. We first track the average steps of the search during the training phase. As shown in Tab. 8, our model takes less than two steps to reach another sample in the credible group. The statistics as reported in Fig. 7 further indicates that our search process converges well across the training epochs.

5.6 Distribution Shift Analysis

TPDS is a probability distribution alignment based scheme. This sub-section gives a distribution shift analysis using the measure of MMD distance (ZongxianLee, 2019) to verify whether TPDS reduces the match error via the progressive alignment. In this experiment, we feedforward the target data $\mathcal{X}_t$ through the source model θ_s and take the outputs as an empirical estimation of the source distribution, *i.e.*, $P_{\mathrm{S}} = \theta_s(\mathcal{X}_t)$. Besides, we train an idea target model $\tilde{\theta}_t$ over $\mathcal{X}_t$ with labels like the source model training (`Appendix` B) and use the outputs to express the ideal target domain, *i.e.*, $P_{\mathrm{T}} = \tilde{\theta}_t(\mathcal{X}_t)$. For comparison, we select SHOT as the baseline owing to the same epoch-wise learning strategy but without errors mitigation. The prediction target distributions obtained by TPDS and SHOT are presented via $\hat{P}_{\mathrm{T:tpds}} = \theta_{tpds}(\mathcal{X}_t)$ and $\hat{P}_{\mathrm{T:shot}} = \theta_{shot}(\mathcal{X}_t)$, respectively.

The first two sub-figures in Fig. 8 display the MMD distance variation to P_{S} and P_{T} during the training, respectively. Both $\hat{P}_{\mathrm{T:tpds}}$ and $\hat{P}_{\mathrm{T:shot}}$ move away from P_{S}, but the distance of $\hat{P}_{\mathrm{T:tpds}}$ to P_{S} keeps always less than $\hat{P}_{\mathrm{T:shot}}$ (the left). This indicates that our progressive alignment is effective for reducing the distance to P_{S}. Thus, the accurate guidance provided by P_{S} are better reserved. Correspondingly, $\hat{P}_{\mathrm{T:tpds}}$ and $\hat{P}_{\mathrm{T:shot}}$ close to P_{T} with an interesting observation (the middle). SHOT's distance decreases rapidly at the early epochs (from 1 to 5), followed by a gradual increase after 6-epoch. In contrast, TPDS's distance presents a declining trend through all the epochs. This phenomenon is understandable. In the late stages, the errors in pseudo-labels propagate further in the adaptation regulated by SHOT. However, TPDS can control this propagation due to that we introduce the adapt error mitigation mechanism.

According to Theorem 1, a progressive alignment on the proxy distribution flow encourages a matching error minimization to the ideal target distribution P_{T}. To verify this conclusion, we perform a further distribution shift analysis on all 12 transfer tasks in the Office-Home dataset. Unlike the analysis mentioned above, we do not use the intermediate models, insteading of the final model finishing 15 epochs training. Besides, the horizontal and vertical coordinates are changed to the distance to P_{S} and P_{T}, respectively, to discover the relation between them. To account for the different cross-domain shift on these tasks, we take a normalization operation on the MMD distance for a clear view. Specifically, the distance of $\hat{P}_{\mathrm{T:tpds}}$ to P_{S} and P_{T} are normalized by the corresponding ones of $\hat{P}_{\mathrm{T:shot}}$, respectively. As shown in the right of Fig. 8, all task points locate in the $[0,1] \times [0,1]$ area and arrange along a red line with a positive slope obtained by linear regression. It is shown that when the distribution shift from the prediction

Table 9: Ablation study results (%) on loss terms with TPDS objective.

Loss item L_{W}	Loss item L_{B}	Office-31	Office-Home
✗	✗	78.6	59.6
✓	✗	87.5	71.7
✓	✓	**90.2**	**73.5**

target distribution ($\hat{P}_{\mathrm{T:tpds}}$) to the source distribution (P_{S}) is controlled to small, the shift from $\hat{P}_{\mathrm{T:tpds}}$ to P_{T}, *i.e.,* the matching error, is suppressed correspondingly. Furthermore, they have a positive correlation. Clearly, the results are consistent with the stated in Theorem 1.

5.7 Ablation Study

Effectiveness of objective components. This part isolates the effect of the loss components in objective L_{TPDS}. Our ablation experiment is conducted in an incremental way, as shown in Tab. 9. When both L_{W} and L_{B} are unavailable (the first row), this is the result obtained by the source model θ_s only. When only L_{W} is used for adaptation, the adapted model improves by 10.0% at least. When L_{B} is added, the performance improves further. The comparisons show that both L_{W} and L_{B} effectively improve the transfer performance.
Effectiveness of progressive searching strategy. In the TPDS praradigm, the usage of progressive searching strategy simplifies the large domain shift, between the source and target domain, into several successive single-step searching tasks with small shift. To verify its effect, we give a variation method, denoted as **TPDS-w/o-Progressive**, without our progressive error control but a fixed one. Specifically, through all the epochs, we encourage the current distribution P_{θ_k} align to the source distribution P_{θ_s}, rather than the previous epoch one $P_{\theta_{k-1}}$. As reported in the first two rows of Tab. 10, TPDS-w/o-Progressive is still superior to SHOT but clearly inferior to our full model. This indicates the efficacy of the proposed progressive error control strategy.

Effectiveness of credible Sampling. In our TPDS instantiation, the credible sampling is the key procedure to estimate the distribution shift between two adjacent proxy distributions. It involves two technical components: (1) In the search space, *i.e.,* the credible group $\mathcal{G}$, (2) the credible neighbors are detected by a chain-like search on a feature manifold. To isolate their effectiveness, we give two TPDS variations as:

(1) **TPDS-w/o-$\mathcal{G}$**: We extend the search space to the overall features of all target data instead of the original credible group $\mathcal{G}$. Due to the absence of $\mathcal{G}$, rendering the chain-like search unavailable, we directly select the nearest data as the credible neighbor based on the cosine distance computation.
(2) **TPDS-w/o-ChainSearch**: We directly project the input feature sample to the feature space formed by $\mathcal{G}$, without detecting the credible neighbor by the chain-like search.

Table 10: Ablation study results (%).

Method	Office-31	Office-Home
SHOT	88.7	71.8
TPDS-w/o-Progressive	89.0	72.8
TPDS-w/o-$\mathcal{G}$	88.6	72.8
TPDS-w/o-ChainSearch	89.0	73.1
TPDS-w-MMD	74.4	52.7
TPDS-w-KL	76.4	45.6
TPDS-w-$\mathcal{G}_e$	89.7	73.3
TPDS-w-$\mathcal{G}_o$	90.1	73.4
TPDS	**90.2**	**73.5**

As reported in Tab. 10, the performances of the three methods rank in descending order as TPDS > TPDS-w/o-Manifold > TPDS-w/o-$\mathcal{G}$ in average accuracy. The results indicate that the introducing of both credible group and the manifold hypothesis with chain-like search can boost the final performance. Also, of note that the implementation difference of TPDS-w/o-Manifold and TPDS-w/o-$\mathcal{G}$ is whether adopt the credible group $\mathcal{G}$ as the search space. The better results of TPDS-w/o-Manifold than TPDS-w/o-$\mathcal{G}$ imply that finding credible data are helpful for our unsupervised learning. It is understandable that we absorb the guidance of accurate category information.
Effectiveness of the cross-distribution pairwise alignment based on mutual information. The cross-distribution pairwise alignment on adjacent proxy distributions is encouraged by the mutual information (MI) maximization on the generated data pairs. To evaluate its effectiveness, we propose two comparisons using conventional measure for the aligning: (1) **TPDS-w-MMD**, and (2) **TPDS-w-KL** where the optimization objective of mutual information is changed to MMD and KL-Divergence, respectively.

From the fifth and sixth rows in Tab. 10, it is seen that both TPDS-w-MMD and TPDS-w-KL have a large gap on all datasets compared with TPDS, along with lower performance than the source-model on Office-31 and Office-Home. This phenomenon confirms the critical role of mutual information maximization in our pairwise alignment. In addition, performance deterioration of the two comparison methods is explainable that the loss of

MMD and KL-Divergence are not pairwise objectives whose computations are based on the entire set.

Effectiveness of credible sample construction. To evaluate the credible sample construction, we compare our design ($\mathcal{G}_e \cap \mathcal{G}_o$) with only using either criterion ($\mathcal{G}_e$ or $\mathcal{G}_o$), termed **TPDS-w-$\mathcal{G}_e$** and **TPDS-w-$\mathcal{G}_e$**. The results in Tab. 10 (the last row group) shows that the two selection criteria are complementary for the performance benefit, validating the efficacy of our design.

6 Conclusion

In this work, we have proposed a new *Target Prediction Distribution Searching* (TPDS) paradigm for SFDA. Unlike the previous methods adopting the conventional feature distribution alignment strategy, TPDS seeks for the target prediction distribution with a principled adaptation error mitigation mechanism. Concretely, we construct a flow of proxy prediction distributions and regulate them to be slightly shifted between adjacent ones. Such that this flow smoothly converge to the target distribution long the geodesic path, on which the overall cumulative errors can be elegantly alleviated. The experiment results on five benchmarks show that TPDS can achieve state-of-the-art performance under the SFDA setting.

Acknowledgements This work is partly funded by the German Research Foundation (DFG) and National Natural Science Foundation of China in project Crossmodal Learning under contract Sonderforschungsbereich Transregio 169, the Hamburg Landesforschungsförderungsprojekt Cross, NSFC (61773083); Horizon2020 RISE project STEP2DYNA (691154); the National Key R&D Program of China (2020YFB1313600); the National Natural Science Foundation of China (62206168, 62276048, U1813202).

Data Availability Statement

The authors confirm that the data supporting the findings of this study are available within the articles: (Hoffman et al., 2018) (**Digits**), (Saenko et al., 2010) (**Office-31**), (Venkateswara et al., 2017) (**Office-Home**), (Peng et al., 2017) (**VisDA**), (Li et al., 2017) (**PACS**).

Appendix

A Proof of Theorem 1

The gradual self-training is the main learning framework for the gradual domain adaptation problem (Zhou et al., 2022) . The article of (Kumar et al., 2020) gives the theoretical results of its single-step transfer performance. Since the conclusion provides the basis for our theoretical analysis, here, we first present a brief introduction as follows.

Suppose any single-step transfer process of gradual self-training can be formulated as

$$\theta_i = \mathrm{SST}(\theta_{i-1}, D_i), \tag{12}$$

where D_i stands for the unlabeled samples from the distribution P_i to be learned, θ_{i-1} is the trained model representing distribution P_{i-1}, the learning result of this step is θ_i representing P_i. We have the following theorem for the adaptation upper bound of the single-step transfer process.

Theorem 2 *(Kumar et al., 2020) Given distribution P, Q with Wasserstein-infinity distance-based measure $\mathrm{D}_w(P,Q) = \pi < 1/R$ (R stands for the regularization strength of models to be learned) and marginals on Y are the same so $P(Y) = Q(Y)$ (no label shift). Suppose P, Q satisfy the bounded data assumption, and we have initial model θ with objective loss $L(\theta, P)$, and n unlabeled samples S from Q, and we set $\theta' = \mathrm{SST}(\theta, S)$ letting objective loss $L(\theta', Q) < \alpha^*$ (α^* is a given small loss), then*

$$L(\theta', Q) \leq \frac{2}{1-\pi R} L(\theta, P) + \alpha^* + O\left(\frac{1}{\sqrt{n}}\right).$$

This theorem applies to our formulation which could be considered as a special case of gradual self-training due to the following properties:

(1) Sharing the same transfer strategy: Leveraging a series of intermediate probability distributions that shift smoothly between the source and target domains.
(2) Sharing the same epoch-wise training method: The whole training is sliced into successive epochs, with each epoch realizing a single-step search of an intermediate distribution under the guidance of the previous intermediate distribution.
(3) Each D_i in the original gradual self-training refers to a different set of samples. In our TPDS, we approximate D_i with a subset of the target samples constructed by the hard samples grouped by the so-far trained model representing the previous adjacent intermediate distribution.

With Theorem 2, we further prove the following theorem about the transfer performance upper bound of our progressive searching.

Restatement of Theorem 1 *Suppose the distributions in the proxy distribution flow $\{P_{\theta_k}\}_{k=0}^K$ satisfy no label shift (the C categories are fixed) and the data is bounded (the data is not too large: $\|\boldsymbol{x}_i\|_2^2 \leq \rho$, $\rho > 0$ for $1 \leq i \leq n$). Distribution shifts in this flow are $\Pi = \{\pi_k\}_{k=1}^K$ where π_k change gradually from 1 to K, and $\pi_m = \max(\Pi)$. If the source model $\theta_s = \theta_0$ has low loss $\alpha_0 \geq \alpha^*$ on the source domain, then*

$$L(\theta_K, P_{\theta_K}) \leq \left(\frac{2}{1-\pi_m R}\right)^{K+1} \left(\alpha_0 + O\left(\frac{1}{\sqrt{n}}\right)\right),$$

where $L(\theta_K, P_{\theta_K})$ is the objective loss as learning θ_K for P_{θ_K} prediction, R stands for the regularization strength of θ_K, α^* is a given small loss, n is the target data number.

Proof of Theorem 1 We start with the source model θ_s with loss α_0. Applying Theorem 2 for each proxy distribution search driven by an adjacent distributions alignment in the flow, considering $\pi_m = \max(\Pi) \geq \pi_k$ for $1 \leq k \leq K$, we have

$$\begin{aligned} L\left(\theta_k, P_{\theta_k}\right) &\leq \frac{2}{1-\pi_k R} L\left(\theta_{k-1}, P_{\theta_{k-1}}\right) + \alpha^* + O\left(\frac{1}{\sqrt{n}}\right), \\ &\leq \frac{2}{1-\pi_m R} L\left(\theta_{k-1}, P_{\theta_{k-1}}\right) + \alpha^* + O\left(\frac{1}{\sqrt{n}}\right). \end{aligned}$$

Expanding, it becomes the sum of a geometric series. Due to $\alpha^* \leq \alpha_0$, according to the formula for the sum of a geometric series, we obtain

$$L\left(\theta_K, P_{\theta_K}\right) \leq \left(\frac{2}{1-\pi_m R}\right)^{K+1} \left(\alpha_0 + O\left(\frac{1}{\sqrt{n}}\right)\right).$$

B Objective for Source Model Training

For all transfer tasks on the three datasets, we train the source model θ_s on the source domain in a supervised manner using the following objective of the classic cross-entropy loss.

$$L_S\left(\mathcal{X}_s, \mathcal{Y}_s; \theta_s\right) = -\frac{1}{n_s} \sum_{i=1}^{n_s} \sum_{c=1}^{C} \tilde{l}_{i,c}^s \log p_{i,c}^s,$$

where n_s is the number of the source data, $p_{i,c}^s$ is the c-th element of $\boldsymbol{p}_i^s = \theta_s(\boldsymbol{x}_i^s)$ that is the category probability vector of input instance $\boldsymbol{x}_i^s$ after θ_s mapping; $\tilde{l}_{i,c}^s$ is the c-th element of the smooth label (Müller et al., 2019) $\tilde{\boldsymbol{l}}_i^s = (1-\sigma)\, \boldsymbol{l}_i^s + \sigma/C$, in which $\boldsymbol{l}_i^s$ is a one-hot encoding of hard label y_i^s and $\sigma = 0.1$.

References

Abnar, S., Berg, R. v. d., Ghiasi, G., Dehghani, M., Kalchbrenner, N., & Sedghi, H. (2021). Gradual domain adaptation in the wild: When intermediate distributions are absent. *arXiv preprint*. Retrieved from arXiv:2106.06080

Ahmed, W., Morerio, P., & Murino, V. (2022). Cleaning noisy labels by negative ensemble learning for source-free unsupervised domain adaptation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 1616–1625).

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., & Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. In *Advances in neural information processing systems* (pp. 5061–5072).

Boudiaf, M., Rony, J., Ziko, I. M., Granger, E., Pedersoli, M., Piantanida, P., & Ayed, I. B. (2020). A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses. In *Eccv 2020* (pp. 548–564).

Caseiro, R., Henriques, J.-F., Martins, P., & Batista, J. (2015). Beyond the shortest path: Unsupervised domain adaptation by sampling subspaces along the spline flow. In *IEEE conference on computer vision and pattern recognition* (pp. 3846–3854).

Chen, H.-Y., & Chao, W.-L. (2021). Gradual domain adaptation without indexed intermediate domains. In *Advances in neural information processing systems* (pp. 8201–8214).

Chen, W., Lin, L., Yang, S., Xie, D., Pu, S., Zhuang, Y., & Ren, W. (2021). Self-supervised noisy label learning for source-free unsupervised domain adaptation. *arXiv preprint*. Retrieved from arXiv:2102.11614

Chen, X., Wang, S., Long, M., & Wang, J. (2019). Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International conference on machine learning* (pp. 1081–1090).

Chidlovskii, B., Clinchant, S., & Csurka, G. (2016). Domain adaptation in the absence of source domain data. In *International conference on knowledge discovery and data mining* (pp. 451–460).

Cui, Z., Li, W., Xu, D., Shan, S., Chen, X., & Li, X. (2014). Flowing on riemannian manifold: Domain adaptation by shifting covariance. *IEEE transactions on cybernetics*, *44*(12), 2264–2273.

Deng, Z., Luo, Y., & Zhu, J. (2019). Cluster alignment with a teacher for unsupervised domain adaptation. In *IEEE international conference on computer vision* (pp. 9943–9952).

Du, Y., Yang, H., Chen, M., Jiang, J., Luo, H., & Wang, C. (2021). Generation, augmentation, and alignment: A pseudo-source domain based method for source-free domain adaptation. *arXiv preprint*. Retrieved from arXiv:2109.04015

Du, Z., Li, J., Su, H., Zhu, L., & Lu, K. (2021). Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *IEEE conference on computer vision and pattern recognition* (pp. 3937–3946).

Ghasedi Dizaji, K., Herandi, A., Deng, C., Cai, W., & Huang, H. (2017). Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *IEEE international conference on computer vision* (pp. 5736–5745).

Gong, R., Li, W., Chen, Y., & Gool, L. V. (2019). Dlow: Domain flow for adaptation and generalization. In *IEEE/CVF conference on computer vision and pattern recognition* (pp. 2477–2486).

Gopalan, R., Li, R., & Chellappa, R. (2011). Domain adaptation for object recognition: An unsupervised approach. In *IEEE international conference on computer vision* (pp. 999–1006).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition* (pp. 1180–1189).

Hoffman, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., ... Darrell, T. (2018). Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning* (pp. 1994–2003).

Huang, J., Guan, D., Xiao, A., Lu, S., & Shao, L. (2022). Category contrast for unsupervised domain adaptation in visual tasks. In *IEEE conference on computer vision and pattern recognition* (pp. 1203–1214).

Jabi, M., Pedersoli, M., Mitiche, A., & Ayed, I. B. (2019). Deep clustering: On the link between discriminative models and k-means. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(6), 1887–1896.

Ji, X., Henriques, J. F., & Vedaldi, A. (2019). Invariant information clustering for unsupervised image classification and segmentation. In *IEEE conference on computer vision and pattern recognition* (pp. 9865–9874).

Jiang, X., Lao, Q., Matwin, S., & Havaei, M. (2020). Implicit class-conditioned domain alignment for unsupervised domain adaptation. In *International conference on machine learning* (pp. 4816–4827).

Jin, Y., Wang, X., Long, M., & Wang, J. (2020). Minimum class confusion for versatile domain adaptation. In *Europeon conference on computer vision* (pp. 464–480).

Kim, Y., Cho, D., Han, K., Panda, P., & Hong, S. (2021). Domain adaptation without source data. *IEEE Transactions on Artificial Intelligence*, *2*(6), 508-518.

Kumar, A., Ma, T., & Liang, P. (2020). Understanding self-training for gradual domain adaptation. In *International conference on machine learning* (pp. 5468–5479).

Lao, Q., Jiang, X., & Havaei, M. (2021). Hypothesis disparity regularized mutual information maximization. In *the AAAI conference on artificial intelligence* (pp. 8243–8251).

Lee, C.-Y., Batra, T., Baig, M. H., & Ulbricht, D. (2019). Sliced wasserstein discrepancy for unsupervised domain adaptation. In *IEEE conference on computer vision and pattern recognition* (pp. 10285–10295).

Li, D., Yang, Y., Song, Y.-Z., & Hospedales, T. M. (2017). Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision* (pp. 5542–5550).

Li, J., Du, Z., Zhu, L., Ding, Z., Lu, K., & Shen, H. T. (2022). Divergence-agnostic unsupervised domain adaptation by adversarial attacks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(11), 8196-8211.

Li, R., Jiao, Q., Cao, W., Wong, H.-S., & Wu, S. (2020). Model adaptation: Unsupervised domain adaptation without source data. In *IEEE conference on computer vision and pattern recognition* (pp. 9638–9647).

Liang, J., Hu, D., & Feng, J. (2020). Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning* (pp. 6028–6039).

Liang, J., Hu, D., Wang, Y., He, R., & Feng, J. (2021). Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Trans. Pattern Anal. Mach. Intell.*. (doi: 10.1109/TPAMI.2021.3103390) doi: 10.1109/TPAMI.2021.3103390

Liu, Y., Zhang, W., & Wang, J. (2021). Source-free domain adaptation for semantic segmentation. In *IEEE conference on computer vision and pattern recognition* (pp. 1215–1224).

Long, M., Cao, Y., Wang, J., & Jordan, M. (2015). Learning transferable features with deep adaptation networks. In *International conference on machine learning* (pp. 97–105).

Long, M., Cao, Z., Wang, J., & Jordan, M. (2018). Conditional adversarial domain adaptation. In *Advances in neural information processing systems* (pp. 1647–1657).

Lu, Z., Yang, Y., Zhu, X., Liu, C., Song, Y.-Z., & Xiang, T. (2020). Stochastic classifiers for unsupervised domain adaptation. In *IEEE/CVF conference on computer vision and pattern recognition* (pp. 9111–9120).

Mueller, J. W., & Jaakkola, T. (2015). Principal differences analysis: Interpretable characterization of differences between distributions. *Advances in Neural Information Processing Systems*, *28*.

Müller, R., Kornblith, S., & Hinton, G. E. (2019). When does label smoothing help? In *Advances in neural information processing systems* (pp. 4696–4705).

Munro, J., & Damen, D. (2020). Multi-modal domain adaptation for fine-grained action recognition. In *IEEE conference on computer vision and pattern recognition* (pp. 119–129).

Pan, Y., Yao, T., Li, Y., Wang, Y., Ngo, C.-W., & Mei, T. (2019). Transferrable prototypical networks for unsupervised domain adaptation. In *IEEE conference on computer vision and pattern recognition* (pp. 2239–2247).

Paninski, L. (2003). Estimation of entropy and mutual information. *Neural computation*, *15*(6), 1191–1253.

Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., & Saenko, K. (2017). Visda: The visual domain adaptation challenge. *arXiv preprint*. Retrieved from arXiv:1710.06924

Qiu, Z., Zhang, Y., Lin, H., Niu, S., Liu, Y., Du, Q., & Tan, M. (2021). Source-free domain adaptation via avatar prototype generation and adaptation. In *International joint conference on artificial intelligence.*

Roy, S., Krivosheev, E., Zhong, Z., Sebe, N., & Ricci, E. (2021). Curriculum graph co-teaching for multi-target domain adaptation. In *IEEE/CVF conference on computer vision and pattern recognition* (pp. 5351–5360).

Roy, S., Trapp, M., Pilzer, A., Kannala, J., Sebe, N., Ricci, E., & Solin, A. (2022). Uncertainty-guided source-free domain adaptation. In *European conference on computer vision* (pp. 537–555).

Saenko, K., Kulis, B., Fritz, M., & Darrell, T. (2010). Adapting visual category models to new domains. In *Europeon conference on computer vision* (pp. 213–226).

Saito, K., Ushiku, Y., Harada, T., & Saenko, K. (2018). Adversarial dropout regularization. In *International conference on learning representations.* OpenReview.net.

Shen, J., Qu, Y., Zhang, W., & Yu, Y. (2018). Wasserstein distance guided representation learning for domain adaptation. In *AAAI conference on artificial intelligence* (Vol. 32).

Tang, H., Chen, K., & Jia, K. (2020). Unsupervised domain adaptation via structurally regularized deep clustering. In *IEEE/CVF conference on computer vision and pattern recognition* (pp. 8725–8735).

Tang, S., Ji, Y., Lyu, J., Mi, J., & Zhang, J. (2019). Visual domain adaptation exploiting confidence-samples. In *Ieee international conference on intelligent robots and systems* (pp. 1173–1179).

Tang, S., Shi, Y., Ma, Z., Li, J., Lyu, J., Li, Q., & Zhang, J. (2021). Model adaptation through hypothesis transfer with gradual knowledge distillation. In *IEEE international conference on intelligent robots and systems* (pp. 5679–5685).

Tang, S., Zou, Y., Song, Z., Lyu, J., Chen, L., Ye, M., … Zhang, J. (2022). Semantic consistency learning on manifold for source data-free unsupervised domain adaptation. *Neural Networks*, *152*, 467-478.

Tanwisuth, K., Fan, X., Zheng, H., Zhang, S., Zhang, H., Chen, B., & Zhou, M. (2021). A prototype-oriented framework for unsupervised domain adaptation..

Tian, J., Zhang, J., Li, W., & Xu, D. (2022). Vdm-da: virtual domain modeling for source data-free domain adaptation. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(6), 3749–3760.

Tzeng, E., Hoffman, J., Saenko, K., & Darrell, T. (2017). Adversarial discriminative domain adaptation. In *IEEE conference on computer vision and pattern recognition* (pp. 2962–2971).

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., & Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. *arXiv preprint*. Retrieved from arXiv:1412.3474

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*(11), 2579-2605.

Venkateswara, H., Eusebio, J., Chakraborty, S., & Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. In *IEEE conference on computer vision and pattern recognition* (pp. 5385–5394).

Wang, H., Li, B., & Zhao, H. (2022). Understanding gradual domain adaptation: Improved analysis, optimal path and beyond. *arXiv preprint*. Retrieved from arXiv:2204.08200

Wang, X., Jin, Y., Long, M., Wang, J., & Jordan, M. (2019). Transferable normalization: Towards improving transferability of deep neural networks. In *Advances in neural information processing systems* (pp. 1951–1961).

Wu, Y., Inkpen, D., & El-Roby, A. (2020). Dual mixup regularized learning for adversarial domain adaptation. In *Europeon conference on computer vision* (pp. 540–555).

Xia, H., Jing, T., & Ding, Z. (2022). Maximum structural generation discrepancy for unsupervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. doi: 10.1109/TPAMI.2022.3174526

Xia, H., Zhao, H., & Ding, Z. (2021). Adaptive adversarial network for source-free domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9010–9019).

Xu, R., Li, G., Yang, J., & Lin, L. (2019). Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *IEEE international conference on computer vision* (pp. 1426–1435).

Yang, S., van de Weijer, J., Herranz, L., Jui, S., et al. (2021). Exploiting the intrinsic neighborhood structure for source-free domain adaptation. In *Advances in neural information processing systems* (pp. 532–542).

Yang, S., Wang, Y., van de Weijer, J., Herranz, L., & Jui, S. (2020). Unsupervised domain adaptation without source data by casting a bait. *arXiv preprint*. Retrieved from arXiv:2010.12427

Yue, Z., Sun, Q., Hua, X.-S., & Zhang, H. (2021). Transporting causal mechanisms for unsupervised domain adaptation. In *IEEE/CVF international conference on computer vision* (pp. 8599–8608).

Zhang, J., Huang, J., Tian, Z., & Lu, S. (2022). Spectral unsupervised domain adaptation for visual recognition. In *IEEE conference on computer vision and pattern recognition* (pp. 9829–9840).

Zhang, Y., Tang, H., Jia, K., & Tan, M. (2019). Domainsymmetric networks for adversarial domain adaptation. In *IEEE conference on computer vision pattern recognition* (pp. 5031–5040).

Zhou, S., Wang, L., Zhang, S., Wang, Z., & Zhu, W. (2022). Active gradual domain adaptation: Dataset and approach. *IEEE Transactions on Multimedia*, *24*, 1210-1220. doi: 10.1109/TMM.2022.3142524

ZongxianLee. (2019). *A pytorch implementation of maximum mean discrepancies (MMD) loss.* https://github.com/ZongxianLee/MMD_Loss.Pytorch.