Applied Intelligence (2022) 52:14070–14084
https://doi.org/10.1007/s10489-021-03112-9

Decomposed-distance weighted optimal transport for unsupervised domain adaptation

Bilin Wang[1,2] · **Shengsheng Wang**[1,2] · **Zhe Zhang**[1,2] · **Xin Zhao**[1,2] · **Zihao Fu**[1,2]

Accepted: 12 December 2021 / Published online: 2 March 2022

Abstract

Unsupervised Domain Adaptation (UDA) aims to transfer knowledge from a label-rich source domain to an unlabeled target domain with a different but related distribution. Optimal Transport (OT) based Wasserstein distance has recently been used to measure and reduce the domain discrepancy in virtue of its robustness in distance measurement. However, the inaccurate estimation of the transport cost between samples is harmful to the fine-grained domain alignment. This paper proposes Decomposed-Distance Weighted Optimal Transport (DDW-OT) method for better adaptation. Technically, according to the clustering-based prototype generation (CPG), DDW-OT constructs a decomposed-distance reweighing matrix to revise the original inaccurate transport distance on sample-level, which conjoins the category uncertainty of the target samples and the correlation degree of category between domains. Besides, the dual-OT solver takes neural netw11 orks to parameterize the dual variables and alleviate the computation cost. DDW-OT also allocated an explicit class-conditional alignment strategy to enhance transfer performance. Extensive experiments on benchmarks demonstrate the effectiveness of the proposed method.

Keywords Optimal transport · Unsupervised domain adaptation · Deep clustering · Wasserstein distance

1 Introduction

Currently, machine learning has been used in multiple applications and industries [20, 27, 33]. In recent years, the rapid improvement of computing power has promoted the development of deep learning algorithms. Deep Neural Network (DNN) has the ability to model complex relationships, and the large-scale labeled datasets make it learn specific representations across a variety of learning tasks [31, 50, 54]. However, the well-trained deep learning models cannot perform well on unlabeled new datasets (or domains) due to their differences in probability distributions. Unsupervised Domain Adaptation (UDA) comes up as an appealing way to solve this domain shift problem, which takes into account samples not only from the labeled source domain but also the unlabeled target domain. UDA provides plenty of methods by learning transferable knowledge to generalize a target model [46, 48] and has been extended to various applications [21, 45].

The main idea in UDA is to generate domain-invariant features and minimize the domain discrepancy. It is common to use the Maximum Mean Discrepancy [30, 31] or the series of the H-divergence [5, 6, 41, 55] to measure the distance between domains. To preserve the topology of the data, the Wasserstein metric has been used in DA with several theoretical guarantees [10, 36]. The OT-based domain divergence, *i.e.*, the total cost of transporting, accumulates the cost in moving the mass between distributions. In UDA, the moving cost is often computed as the square Euclidean distance between samples in the feature space. While the OT distance has the strong ability to retain the spatial geometry information of distributions, the OT-based methods are still impracticable in the measurement of the intra-class domain discrepancy.

This work is supported by the Innovation Capacity Construction Project of Jilin Province Development and Reform Commission (2021FGWCXNLJSSZ10), the National Key Research and Development Program of China (No. 2020YFA0714103) and the Science & Technology Development Project of Jilin Province, China (20190302117GX) and the Graduate Innovation Fund of Jilin University under Grant 101832020CX179.

✉ Shengsheng Wang
wss@jlu.edu.cn

[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China

[2] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

The inaccurate transport cost matrix always causes class-level misalignment.

To solve the above problems, we propose decomposed-distance weighted optimal transport (DDW-OT), an end-to-end UDA model, which incorporates the spatial structure of the data distributions for more accurate fine-grained alignment. Specifically, there are mainly three objectives in DDW-OT. At first, DDW-OT estimates the prototypical representations of both domains through clustering and then constructs a reweighing matrix, which conjoins the category uncertainty of the target samples and the correlation degree of category between domains to revise the original coarse transport cost matrix. For computation efficiency, inspired by the stochastic OT [43], we employ fully connected neural networks to parameterize the dual variables and estimates the domain discrepancy by solving the regularized-OT problem. Moreover, DDW-OT also allocated an explicit class-conditional alignment and discriminate strategy to enhance transfer performance. The contributions of this paper can be summarized as follows:

(1) We utilize the clustering-based method to generate relatively reliable prototypical representations for the target domain instead of using the pseudo classification probability predicted by the source model, reducing the negative effect caused by the sensitivity of inaccurate prediction.
(2) We devise the decomposed-distance based on the spatial information of samples and the above prototypical representations, which precisely characterized the association between domains in sample-level, and alleviate intra-class discrepancy implicitly.
(3) The dual variables in the regularized-OT problem are parameterized by two shallow neural networks and optimized inside the overall training process. Experimental results show that DDW-OT achieves competitive performance on several benchmark datasets compared with the latest UDA methods.

2 Related work

UDA has attached increasing attention, and has been divided mainly into two directions: (1)utilizing a distance metric to measure and minimize the domain divergence [8, 12, 16, 46, 48] and (2)learning domain invariant feature representations through adversarial-based methods [15, 42, 44]. Here we summarize the work most relevant to our proposed method.

Discrepancy-based methods Typical discrepancy-based methods are usually set out from several aspects [22]. The prevailing approach is the feature-based distribution alignment, which utilizes a distance metric [31, 46, 48] or adopt adversarial learning [1, 23, 29, 47] to minimize the domain discrepancy. The classifier-based adaptation turns the domain divergence into the disparity measure between the scores provided by two independent scoring functions [5, 41, 55, 56]. To obtain fine-grained class-level alignment, [9, 13, 35] generate prototypes for each category in source and target domains, and explicitly minimize the distance between prototypes.

OT for DA Optimal Transport (OT) [37] recently shown to be an up-and-coming tool to perform DA tasks. OT consists of mapping two source and target probability measures with a minimal transportation cost associated with the so-called Wasserstein distance. In [38], the authors provide the theoretical guarantee that the divergence between domains measured by Wasserstein distance can converge to the generalization bounds. [10] first learns a transportation plan matching both domains and then computes a transformation of source samples through barycentric mapping. Damodaran et al. [12] provides an end-to-end method that minimizes the divergence between domains and learning a classifier simultaneously.

In selecting the distance function, it is common to use the Euclidean metric as distance measurement to computing the coupling matrix for OT. However, RWOT [52] points out that the direct use of pure square Euclidean distance cannot precisely measure the transmission cost between samples, for the coarse match probably leads to negative transfer. Furthermore, RWOT exploited prototypical spatial information and proposed a weighted optimal transport strategy to achieve the precise pair-wise transport procedure. Similarly, ETD [26] computes a re-weighted distance matrix based on the attention mechanism to adjust the current batch to the real distribution. Unlike the above cost matrix weighting algorithm, [25] utilizes the Mahalanobis distance instead of the Euclidean distance to aligns the subspace generated by PCA across domains.

OT solver According to some ground cost, OT distances compute the minimal effort for moving the probability mass of one distribution to the other, which could be seen as a linear program in discrete distribution. The prevalent way to compute discrete OT distance is by solving the so-called Kantorovitch problem [24]. However, the computation of the transport plan has an enormous computational cost. A commonly used approach is to add entropy penalization to the primal Kantorovitch problem. The Sinkhorn algorithm [11] can solve the entropy-regularized OT efficiently, meanwhile differentiable w.r.t. their inputs, enabling used as a loss function in a machine learning pipeline [12, 52]. Besides, to handle continuous probability measures, Genevay et al. [4] optimized a "semi-dual" objective

function through stochastic gradient methods. Arjovsky et al. [2] first used the dual-objective function to measure the discrepancy between distributions, with the constraints that the dual function needs to be constrained by a 1-Lipschitz function. Seguya et al. [43] proposed a two-step method. They used a neural network to approximate the optimal map between the underlying continuous measures and prove the convergence of the regularized optimal plan.

In this paper, we concentrate on the objective function derived by regularized-OT distances rather than the optimal transport plan. We neglect the sparsity that the regularization terms bring and solve the regularized dual OT problem by parameterizing the dual variables with neural networks.

3 Methodology

Firstly, we introduce the formulation of the problem. Suppose we are given a set of labeled source data $X^s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, including n_s samples associated with class labels $y_i^s \in \{1, 2, ..., K\}$, and a target domain set $X^t = \{x_j^t\}_{j=1}^{n_t}$ of n_t samples with unknown labels. Notably, it is supposed that the source and target samples have the same dimension $x^{s(t)} \in R^d$ and contain the same classes K but are drawn from different probability distributions. The discrepancy between the two probability distributions makes the classifier learned on the source domain cannot be directly adapted to the target domain with robustness. Deep learning methods have been introduced to learn a transferable model and finding domain-invariant representations to overcome the domain shift, which is a particularly challenging aspect of the UDA tasks.

Our method originates from the discrepancy-based methods, which simultaneously optimizing the classifier and minimizing the distance between the marginal distributions of the two domains. To be specific, we use the OT distance to depict the discrepancy loss. The architecture of Decomposed-Distance Weighted Optimal Transport (DDW-OT) is illustrated in Fig. 1. Apart from the primary pipeline, our proposed DDW-OT method including the Clustering-based Prototype Generation (CPG) module and compute the DDW matrix to reweigh the original distance C. Meanwhile, the Dual-OT solver is used to measure the discrepancy between domains.

3.1 Optimal transport revisit

Optimal transport (OT) is a powerful computational tool to measure the difference of probability distributions. The original formulation of OT was first proposed by Monge, which searched for a map to minimize the total cost between distributions. Under the domain adaptation scene, the source and target domains are two distinct joint probability

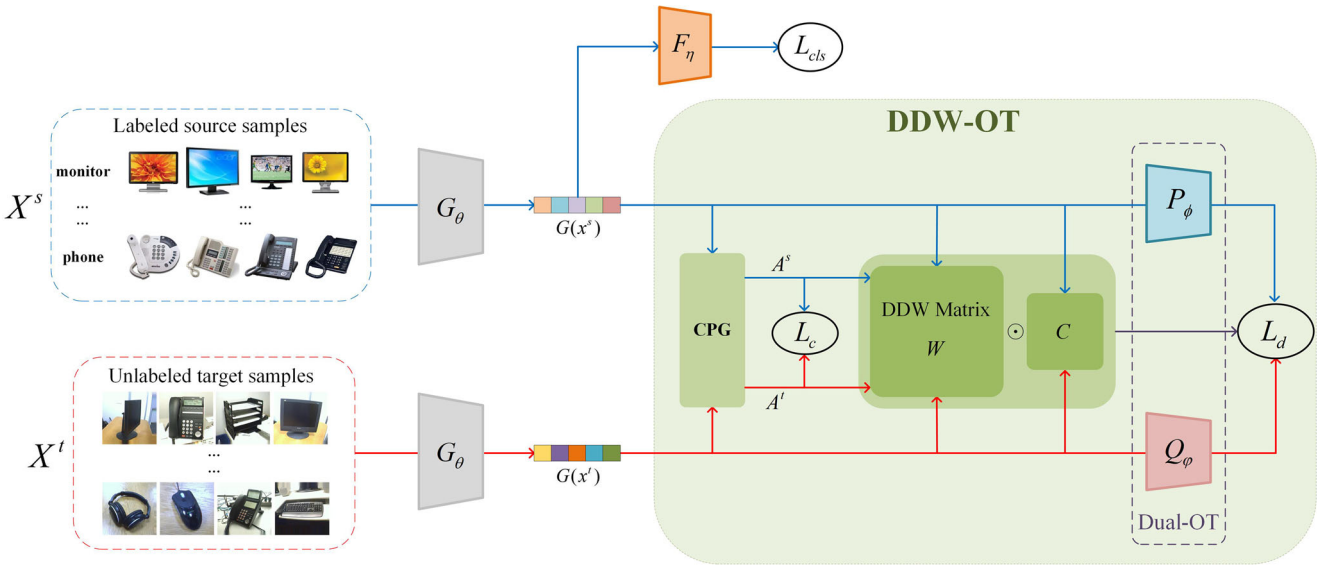

Fig. 1 The framework of the proposed DDW-OT. The shared feature extractor G_θ maps the domains into a common latent space. The CPG module generates the prototypical representations A^s, A^t and further used to compute a DDW matrix W to reweigh the original OT distance C. Finally, the dual-OT solver takes the features of both domains together with the reweigh-OT distance to measure the domain discrepancy

distributions, noted as Ω_s and Ω_t, where Ω is a measure space. We can estimate the empirical distributions μ_s and μ_t through finite discrete samples.

Kantorovich [24] formulation of OT is a convex relaxation of the Monge problem, which is cast as a linear program:

$$OT(\mu_s, \mu_t) := min_{T \in U(\mu_s, \mu_t)} < T, C > \quad (1)$$

where U is the transportation couplings between distributions:

$$U(\mu_s, \mu_t) := \{T \in \mathbb{R}_+^{n_s \times n_t} : T\mathbf{1}_{n_t} = \mu_s, T^\top \mathbf{1}_{n_s} = \mu_t\} \quad (2)$$

and $C \in R^{n_s \times n_t}$ is a cost matrix. whenever the cost C is a norm raised to the power p, it is referred to as the Wasserstein distance:

$$\begin{aligned} OT(\mu_s, \mu_t)^{\frac{1}{p}} : &= \left(\inf_{T \in U} \int_{\Omega_s \times \Omega_t} d(x_s, x_t)^p dT(x_s, x_t) \right)^{\frac{1}{p}} \\ &= \inf_{x_s \in \mu_s, x_t \in \mu_t} \left\{ \left(\mathbb{E}\ d(x_s, x_t)^p \right)^{\frac{1}{p}} \right\} \end{aligned} \quad (3)$$

where $c(x_s, x_t) = d(x_s, x_t)^p$. Experimentally, to compute optimal transportation, the best result is usually obtained when choosing the square Euclidean distance as the distance function between two locations [10], *i.e.* $p = 2$.

3.2 Decomposed-distance weighted OT

In this section, we begin with the clustering-based prototype generation method, which accurately estimates the prototypical representations of the target domain without the error accumulation brought by explicit pseudo-label predictions. We then propose the decomposed-distance of samples between source and target domain based on the above prototypical information. Finally, we combine the two-parts distance and construct a weighting matrix to refine the primal Euclidean distance and derive the authentic OT distance between domains.

3.2.1 Clustering-based prototype generation

In UDA, it is prevailing to align the marginal distributions to learn domain-invariant features between domains. However, the neglect of category information leads to the misalignment of samples in different categories. Prototype-based class-conditioned domain alignment [9, 35] is proposed to address this problem. The most common class-level alignment method is achieved by narrowing the distance of prototypical representations of the same class. The prototypical information is derived by the mean embedding of samples within the same category. The main limitation of this method is that the unlabeled target domain's prototypical representations rely on the predicted pseudo-labels, which is inaccurately caused by error accumulation.

Recently, the clustering method combined with the neural network has been widely used in many scenes [3, 53]. The clustering-based DA approaches [13, 14] grouping the unlabeled target samples likely belong to the same class, under the hypothesis that the embedding vectors generated by an embedding function could be seen as multiple discriminative clusters in a high-dimensional feature space. We utilize the spherical K-means [17] clustering method to generate relatively reliable prototypical representations for the target domain without relying on the pseudo-labels.

The learning of prototypical representations alternates with the parameter optimization of the classification network. To be specific, in each training epoch, we first derive all the embedding vectors of training data from both domains through the feature extractor, and then we perform the clustering-based methods to generate prototypical representations. In the first iteration, the initial centers $A^t = \{a_i^t\}_{i=1}^K$ are randomly selected from the embedding vector $G(X^t)$ of target samples $X^t = \{x_j^t\}_{j=1}^{n_t}$. The number of clusters is set as the number of classes K in the label space. To obtain stable clusters, we aim to minimize the objective function :

$$J(\mathbb{I}, a^t) = \sum_{i=1}^{n^t} \sum_{j=1}^{K} \mathbb{I}_{ij} dist(x_i^t, a_j^t) \quad (4)$$

where $\mathbb{I}_{ij} = \begin{cases} 1, & j = \text{argmin}_j ||x_i^t - a_j^t|| \\ 0, & \text{otherwise} \end{cases}$ to indicate whether the target sample x_i^t is assigned to the cluster a_j^t. During clustering, we take the commonly used cosine dissimilarity as the distance measurement function between samples, i.e. $dist(x_i^t, a_j^t) = \frac{1}{2}(1 - \frac{<x_i^t, a_j^t>}{||x_i^t|| ||a_j^t||})$. The overall clustering process is iteratively into two steps and repeat until convergence: (1) Assignment step: assign samples to the nearest cluster. (2) Update step: set the cluster center as the mean value of the current samples embedding which belonging to this cluster.

After clustering, we derive K clusters on target samples, and the label of each cluster could be assigned based on the distance to each source prototype. The source prototypical representation is the mean value of the source samples embedding of each class:$A^s = \{a_k^s\}_{k=1}^K = \frac{1}{|S_k^s|} \sum_{x_i^s \in S_k^s} G(x_i^s; \theta)$, where S_k^s denotes the sets of source samples from class k. Therefore, assigning labels to target clusters based on the distance between source and target prototypes is equivalent to solving the minimum weight matching problem in bipartite graphs [34]. The process of clustering-based prototype generation is graphically illustrated in Fig. 2.

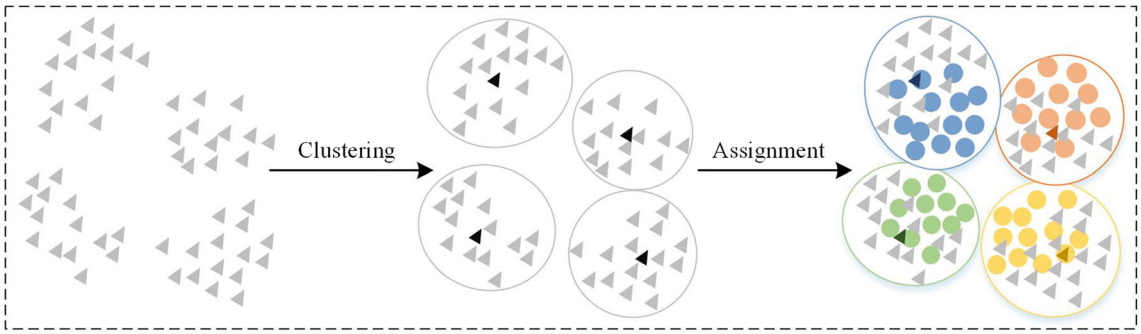

Fig. 2 The process of Clustering-based Prototype Generation (CPG). Firstly, according to the representations of target samples, K clusters are obtained and the representations of the cluster centers are derived. Then, assign the category labels to each target cluster under the condition that the total distance of the prototypes between domains is minimum

3.2.2 Two parts of decomposed-distance

When using OT-based Earth-mover (EM) distance to depict the divergence between domains, it is critical to accurately estimate the transport cost matrix to capture the geometric characteristics of the discrete samples' feature distribution. The traditional method commonly use Euclidean distance as the transport cost, shown in Fig. 3(a). However, samples at the edge of distribution are most likely closer to the samples of distinct classes in this setting. The inaccurate measurement will establish a large proportion of transport between different class distributions among domains, resulting in class-level domain misalignment. To overcome this issue, we propose the decomposed-distance, which illustrated in Fig. 3(c), and further use it to refine the original cost matrix. See Section 3.2.3 for details.

Intuitively, the decomposed-distance is a compromise between the target domain's prototypical distribution and the intra-domain structure. The first part of the decomposed-distance estimates the category uncertainty of the target samples. Specifically, according to the target prototypical representations $A^t \in \mathbb{R}^{K\times d}$ obtained in Section 3.2.1, where d is the dimension of representations. We derive an uncertainty matrix $M_{unc} \in \mathbb{R}^{b\times K}$ as:

$$M_{unc}(x_j^t, a_i^t) = \frac{\exp\left\{-\left\|G(x_j^t) - a_i^t\right\|_2^2\right\}}{\sum_{c=1}^{K}\exp\left\{-\left\|G(x_j^t) - a_c^t\right\|_2^2\right\}} \tag{5}$$

where b represents the batch size in the training step.

The second part of the decomposed-distance depicts the intra-domain structure based on prototypical information of each domain. We assume that each target sample is assigned to the nearest cluster, *i.e.*, each x_i^t in the current batch can be represented by its corresponding prototypical representation a_i^t. Similarly, each source sample x_j^s can be represented by its class prototype a_j^s. The intra-domain correlation matrix $M_{cor} \in \mathbb{R}^{n\times K}$ is defined as:

$$M_{cor}(a_i^t, a_j^s) = \frac{\exp\left\{-\left\|a_i^t - a_j^s\right\|_2^2\right\}}{\sum_{c=1}^{K}\exp\left\{-\left\|a_i^t - a_c^s\right\|_2^2\right\}} \tag{6}$$

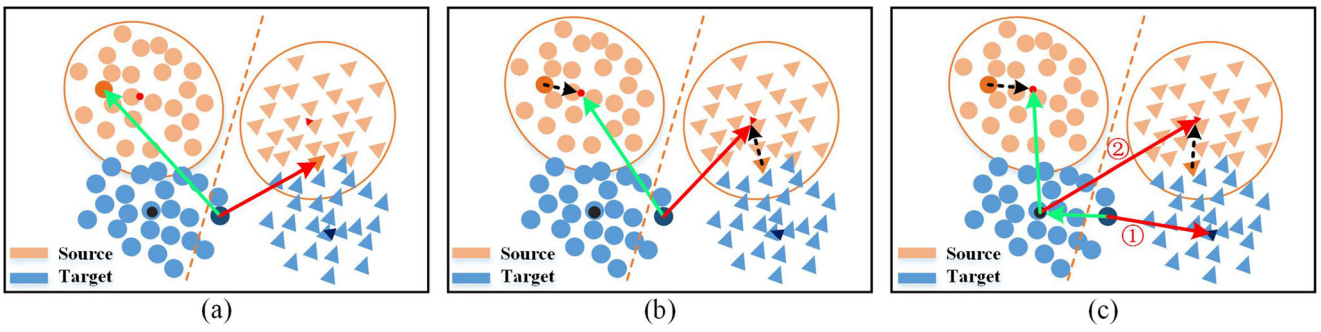

Fig. 3 Illustration of the proposed Decomposed-distance. Green arrow: True distance. Red arrow: Wrong distance. Black dash arrow: corresponding prototype. (a) Squared Euclidean distance between samples. The drawback is that the target samples at the edge of the distribution (dark dot) may be less distant to the source samples from other categories. (b) Replace the source samples with the corresponding prototype, but still len (green)>len(red). (c) The proposed decomposed-distance. The combination of the two-parts distance achieves reasonable estimation

In the next section, we construct a weight refined matrix based on the decomposed-distance mentioned above to estimate the actual OT distance between domains.

3.2.3 Decomposed-distance weight matrix

First of all, based on the uncertainty matrix M_{unc} and the intra-domain correlation matrix M_{cor} introduced in the last section, we combine the two parts of decomposed matrix as $M \in \mathbb{R}^{n\times K}$. This matrix contains the distance between each target sample in the current batch and each prototypical representation of the source domain:

$$M(x_j^t, a^s) = \lambda M_{unc} + (1-\lambda) M_{cor} \tag{7}$$

where λ is a trade-off parameter of two components. The decomposed-distance formed by uncertainty matrix M_{unc} and intra-domain correlation matrix M_{cor} prevent the single relationship between the target sample and the prototypical representations of the source domain, which is shown in Fig. 3(b).

Then, in order to derive the reweigh matrix $W \in \mathbb{R}^{n\times n}$ adapted to the current training batch, the `select`(.) function is used to pick out the columns in $M(x_j^t, a^s)$ with the set B containing the category index of the source domain that appears in the current batch. In addition, the distance between samples is negatively correlated with the correlation degree. The reweigh matrix W can be defined as:

$$W(x_i^s, x_j^t) = 1 - \texttt{select}_B(M(x_j^t, a^s))^{\mathrm{T}} \tag{8}$$

Here, for the reweighing matrix W, to avoid the over smoothing brought by the softmax, we design an average temperature softmax which originated from [19]. The set T contains several temperature hyper-parameters to control the sparsity of W. In experiments, $T = \{T_1, T_2, T_3\} = \{5, 10, 25\}$ is recommended.

$$W_{ij} = \frac{1}{M}\sum_{m=1}^{M} \frac{\exp(w_{ij}\cdot(-T_m))}{\sum_{j=1}^{n}\exp(w_{ij}\cdot(-T_m))} \tag{9}$$

Furthermore, the reweigh matrix W is used to refine the original transport cost matrix C in (1), and construct an accurate OT-based domain discrepancy in consideration of intra-domain structure:

$$OT(\mu_s, \mu_t) := min_{T\in U(\mu_s,\mu_t)} < T, C\cdot W > \tag{10}$$

3.3 Dual OT algorithm

Previous methods used to find a push-forward nonlinear transformation T [10] or a coupling matrix γ [12, 52] and construct an objective function based on them so as to reduce the domain discrepancy. However, the generation of a specific transport plan is redundant if the OT-based distance can be calculated directly. Fortunately, according to Fenchel-Rockafellar's duality theorem, the goal of reducing the domain discrepancy is equivalent to optimizing the dual regularized OT [43] as:

$$\sup \mathbb{E}_{(X^s,X^t)\sim\mu\times\nu}[u(X^s) + v(X^t) + F_\varepsilon(u(X^s), v(X^t))] \tag{11}$$

where u and v are dual variables, F_ε is a penalty term. In this paper, the L_2 regularization is used instead of entropy regularization by virtue of its stability in convergence without exponential terms. Moreover, it has a smaller approximation error than the entropy reg. when in the same ε [7]:

$$F_\varepsilon(u(X^s), v(X^t)) = -\frac{1}{4\varepsilon}(u(X^s) + v(X^t) - C(X^s, X^t))^2 \tag{12}$$

The regularized dual OT problem can be solved by stochastic OT computation [43] by optimizing the dual variables u and v. The dual variables needs to be parameterized so as to carry out the optimization. In this paper, we utilize two shallow fully connected neural networks P_ϕ and Q_φ to approximate the dual variables u and v, respectively, with the computational complexity of $\mathcal{O}(b^2)$. The optimization is shown in Algorithm 1. Formally, the domain discrepancy based on the regularized dual OT can be represented as:

$$\begin{aligned}\mathcal{L}_d(\phi,\varphi) = \sup \mathbb{E}[&P(G(X^s);\phi) + Q(G(X^t);\varphi) \\ &+ F_\varepsilon(P(G(X^s);\phi), Q(G(X^t);\varphi))]\end{aligned} \tag{13}$$

Algorithm 1 Optimization of Dual-OT.

Require: Batch $H_s = \{(x_i^s, y_i^s)\}_{i=1}^b$, $H_t = \{x_j^t\}_{j=1}^b$ sampled from X^s, X^t; DDW-reweigh matrix W; Original cost matrix C; Learning rate lr_2 .

1: Reweigh C. *i.e.*, $C = C \cdot W$;
2: **while** not converged **do**
3: Calculate F_ε using (11);
4: Update $\phi \leftarrow \phi + lr_2 \sum \nabla\phi\,(x^s) + \partial_\phi F_\varepsilon\left(\phi\,(x^s), \varphi\left(x^t\right)\right)\nabla\phi\,(x^s)$;
5: Update $\varphi \leftarrow \varphi + lr_2 \sum \nabla\varphi\left(x^t\right) + \partial_\varphi F_\varepsilon\left(\phi\,(x^s), \varphi\left(x^t\right)\right)\nabla\varphi\left(x^t\right)$;
6: **end while**

3.4 Optimization

In this section, we present the overall algorithm flow of DDW-OT. We first define the classification loss of the source domain:

$$\mathcal{L}_{cls} = \frac{1}{n_s}\sum_{i=1}^{n_s} l(F(G(x_i^s;\theta);\eta), y_i^s) \tag{14}$$

where l is a cross-entropy loss.

According to the class centers of both domains, we introduce an explicit class alignment loss $\mathcal{L}_c$ as:

$$\mathcal{L}_c(a_i^s, \hat{a}_j^t) = \frac{1}{|K|^2}\sum_{i=1}^{K}\sum_{j=1}^{K}[\delta_{ij}\|a_i^s - \hat{a}_j^t\|_2^2 + (1-\delta_{ij})max(0, m - \|a_i^s - \hat{a}_j^t\|_2^2))] \quad (15)$$

where $\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$. Here, $\hat{a}_j^t$ is defined by taking the average of all embedding vectors with the pseudo label j, which predicted by the classifier F_η, so as to encourages the centers from the same class to concentrate together and pushes the centers from different classes far away from each other with a distance m at least.

The total objective function of DDW-OT is described as:

$$\min \mathcal{L} = \mathcal{L}_{cls} + \alpha\mathcal{L}_d + \beta\mathcal{L}_c \quad (16)$$

where α, β are trade-off parameters. The optimization process is shown in Algorithm 2. The basic framework adopts the process in ETD [26], which optimizes the parameters in the dual-OT network and the classification network alternatively.

Algorithm 2 Optimization process of DDW-OT.

Require: $X^s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, $X^t = \{x_j^t\}_{j=1}^{n_t}$; Learning rate lr_1, lr_2.
Ensure: θ, ϕ, φ.
1: Pretrain classification network using $X^s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$;
2: Pretrain Dual-OT network using $X^s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, $X^t = \{x_j^t\}_{j=1}^{n_t}$;
3: **while** not converged **do**
4: Compute A_s, A_t based on CPG using X_s, X_t;
5: Sample minibatch H_s, H_t from X_s, X_t;
6: Calculate DDW reweigh matrix W via (8);
7: Reweigh C = C · W;
8: **while** not converged **do**
9: Algorithm1 (step3 → step5);
10: **end while**
11: Update parameters G_θ and F_η via (15);
12: **end while**

4 Experiments

4.1 Setups

Datasets *Office-31* [40] is a real-world dataset which is used widely in domain adaptation task. It contains 4110 images from 31 categories composed of three distinct domains: Amazon (A), DSLR (D), Webcam (W), which collected images download from amazon.com, taken by digital SLR camera and web camera, respectively. We analyze all six transfer tasks across domains: A→W, A→D, D→W, D→A, W→D and W→A.

OfficeHome [49] is a challenging adaptation dataset, which consists of images from four domains: Art (Ar), Clipart (Cl), Product (Pr), and Real World (Rw), containing 15500 images in 65 categories and overall 12 transfer tasks.

ImageCLEF-DA consists of three dissimilar domains: ImageNet ILSVRC 2012 (I), Pascal VOC 2012 (P), and Caltech-256 (C). The number of images is balance across domains, including 600 images of 12 categories, including six transfer tasks: I→P, P→I , I→C, C→I, C→P and P→C.

VisDA-2017 is a challenging large-scale synthetic-to-real dataset, including more than 200k images across 12 categories in the training, validation, and testing domains. In this paper, we take the training images as the source domain and the validation images as the target domain.

Implementation details All experiments are implemented by the Pytorch framework. The Resnet-50 [18] pretrained on ImageNet [39] is used as our backbone networks, which also equipped with domain-specific batch normalization layers. The mini-batch size is set as 16/30/30/30 per domain for Office-31/OfficeHome/Image-CLEF-DA/VisDA-2017. We utilize stochastic gradient descent (SGD) for the training of feature extractor layers G_θ and FC layers F_η with a momentum of 0.9 with the learning rate lr_1 adjusted following $lr_1 = \frac{lr_0}{(1+mp)^n}$, where p linearly increases from 0 to 1. The initial learning rate lr_0 is set as 0.0005, m=10, and n=0.75, but for VisDA-2017, n=2.25. For the optimization of dual-OT network P_ϕ and Q_φ, we use Adam optimizer with the initial learning rate lr_2 = 0.003. For the trade-off parameters, we set λ=0.5, α=1 and β=0.1, regularization value ε=1, and the constrain margin m=20.

Compared methods To empirically evaluate the advantage of DDW-OT, our approach is compared with several series of methods. We cite the performance of these methods reported in their corresponding papers for a fair comparison. (1)MMD-based models: DAN [28], JAN [31], DWL [51]. (2)Adversarial-based models: DANN [1], ADDA [47], CDAN [29]. (3)OT-based models: Deep-JDOT [12], ETD [26]. Approaches mentioned above are all proposed for learning domain-invariant features for UDA.

4.2 Results and comparison

Table 1 exhibits the results on six tasks from Office-31. As can be seen, all UDA methods outperform ResNet-50, which is only trained by source samples. Overall, our proposed DDW-OT achieves the best performance among these baselines and is better than all comparison methods on half of the transfer tasks. Notably, DDW-OT improves

Table 1 Results (accuracy %) on Office-31 for UDA. The best method is emphasized in bold

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet-50 [18]	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DANN [1]	84.5	96.8	99.4	77.5	66.2	64.8	81.6
ADDA [47]	86.2	96.8	99.1	78.8	69.5	68.5	83.2
JAN [31]	82.0	96.9	99.1	79.7	68.2	67.4	82.2
CDAN [29]	**93.1**	98.2	100.0	**89.8**	70.1	68.0	86.6
Deep-JDOT [12]	88.9	98.5	99.6	88.2	72.1	**70.1**	86.2
ETD [26]	92.1	100.0	100.0	88.0	71.0	67.8	86.2
DWL [51]	89.2	99.2	100.0	91.2	73.1	69.8	87.1
DDW-OT	92.1	**100.0**	**100.0**	90.8	**73.9**	68.5	**87.6**

the accuracy on D→A from 72.1% to 73.9%, under the condition that source domain D only has a relatively small amount of samples compared with the target domain A. These results suggest that our method is able to capture the spatial location information and perform adaptation effectively on the target domain.

Table 2 shows the detailed results on 12 transfer tasks of the OfficeHome dataset. We can see that DDW-OT outperforms other compared approaches in 9 out of 12 transfer tasks. In particular, DDW-OT can improve difficult tasks, such as 57.6% to 65.3% on Cl→Ar, and boosts accuracy on easier tasks 5.6% for Ar→Pr and 4.3% for Cl→Pr.

For the ImageCLEF-DA task, the experiment accuracy is shown in Table 3. It is worth noting that in P→I, DDW-OT shows less transfer efficiency. We assume that it may be caused by the over-dependence on the initial transport cost estimation due to the inaccurate prototypical representations. On average, DDW-OT achieves 90.7%, which is competitive with the latest discrepancy-cased UDA method DWL [51] and outperforms the state-of-the-art OT-based method ETD [26].

We further perform an adaptation of DDW-OT on the VisDA-2017. Table 4 shows the classification accuracy of 12 categories. Our model achieves 79.3% on average, which is higher than 77.1% of the DWL. Notably, DWL adjusts the weight of discriminability loss to control the degree of discriminability. Compared with DWL, our DDW-OT pays more attention to the computational process of inter-domain discrepancy. Besides, VisDA-2017 is a large-scale dataset with a large domain discrepancy, which shows the effectiveness of our DDW-OT method on large datasets.

4.3 Ablation study

The proposed DDW-OT mainly contains two components: decomposed-distance weighting strategy on original OT distance and an explicit class center alignment by operating prototypical representations. We conduct ablation study to separate contributions and verify the effectiveness of each component. The results on OfficeHome are shown in Table 5. We can observe that compared with the source-only method, *i.e.* $\mathcal{L}_{cls}$, the minimization of the domain divergence measured by the Euclidean distance-based original OT can achieve 3.9% improvement on average. Furthermore, by equipping with the proposed decomposed-distance weighting strategy ($\mathcal{L}_{cls}+\alpha\mathcal{L}_d$), the model has a further improvement of accuracy, which demonstrates the

Table 2 Results (accuracy %) on OfficeHome for UDA. The best method is emphasized in bold

Method	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
ResNet-50 [18]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [28]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN [1]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN [31]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN [29]	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
Deep-JDOT [12]	39.7	50.4	62.5	39.5	54.3	53.2	36.7	39.2	63.6	52.3	45.4	70.5	50.7
ETD [26]	51.3	71.9	**85.7**	57.6	69.2	73.7	57.8	**51.2**	79.3	70.2	**57.5**	82.1	67.3
DDW-OT	**53.2**	**77.5**	82.7	**65.3**	**73.5**	**75.2**	**60.5**	49.8	**79.8**	**72.4**	56.3	**83.5**	**69.2**

Table 3 Results (accuracy %) on ImageCLEF-DA for UDA. The best method is emphasized in bold

Method	I→P	P→I	I→C	C→I	C→P	P→C	Avg
ResNet [18]	74.8	83.9	91.5	78.0	65.5	91.27	80.7
DAN [28]	74.8	83.9	91.5	78.0	65.5	91.27	80.7
DANN [1]	75.0	86.0	96.2	87.0	74.3	91.5	85.0
JAN [31]	76.8	88.4	94.8	89.5	74.2	91.7	85.8
CDAN [29]	76.7	90.6	97.0	90.5	74.5	93.5	87.1
ETD [26]	81.0	91.7	97.9	93.3	79.5	95.0	89.7
DWL [51]	82.3	**94.8**	98.1	92.8	77.9	**97.2**	90.5
DDW-OT	**82.6**	92.8	**98.5**	**93.9**	**79.9**	96.7	**90.7**

effectiveness of the transport cost reconstruction between source and target samples, deriving more accurate inter-domain transfer. By adding the explicit class center alignment loss ($\mathcal{L}_{cls}+\alpha\mathcal{L}_d+\beta\mathcal{L}_c$), the prototypes which from the same categories are gathered, and the prototypes from different categories are separated, making the intra-class compactness and inter-class separability, and achieving another improvement of the accuracy.

Effect of clustering-based prototype generation To evaluate the effect of the proposed clustering-based prototype generation (CPG) module, the experiments conducted from two aspects, mainly make a comparison with the pseudo-label-based prototype generation (PPG).

During training, the real prototype for each category P_r of the target domain can be obtained by computing the mean value of the embedding features, which is guided by the ground-truth label of the target domain. Similarly, the clustering-based prototypes P_c and the pseudo-label-based prototypes P_p can also obtained through clustering centers and the pseudo labels generated by the classifier.

We take the real prototypes P_r as the baseline, and analyze the distance between the estimation to the real. The distance $d(P_c, P_r)$obtained by CPG and $d(P_p, P_r)$ obtained by PPG are calculated as follows:

$$\begin{aligned} d(P_c, P_r) &= \frac{1}{K}\sum_{i=1}^{K} dist(P_c^i, P_r^i); \quad d(P_p, P_r) \\ &= \frac{1}{K}\sum_{i=1}^{K} dist(P_p^i, P_r^i) \end{aligned} \tag{17}$$

The visualization of the analysis is shown in Fig. 4. It can be seen that, the relative distance $d(P_c, P_r)$ of CPG is more stable than the distance $d(P_p, P_r)$ of PPG. That means, the inaccurate predicted pseudo label brings more fluctuation to the prototypes, leading to unreliable estimations.

Furthermore, Table 6 shows the performance of DDW-OT equipped with CPG or PPG. The comparisons of this case verify the effectiveness of CPG. The reliable prototypes bring more stability to the follow-up decomposed-distance and further boost the transferability of our DDW-OT method.

Effect of neural network dual variable parameterization To evaluate the effect of the neural network parameterization,

Table 4 Results (accuracy %) on VisDA-2017 for UDA. The best method is emphasized in bold

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg
ResNet-101 [18]	55.0	53.2	61.8	59.2	80.7	17.8	79.6	31.1	81.1	26.4	73.6	8.6	52.3
DANN [1]	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
DAN [28]	87.1	63.0	76.5	42.0	90.3	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
Deep-JDOT [12]	85.4	73.4	77.3	87.3	84.1	64.7	**91.5**	79.3	**91.9**	44.4	88.5	**61.8**	77.4
DWL [51]	**90.7**	**80.2**	**86.1**	67.6	**92.4**	**81.5**	86.8	78.0	90.6	57.1	85.6	28.7	77.1
DDW-OT	89.2	77.6	81.7	**89.0**	87.5	68.4	91.2	**81.4**	89.3	**62.1**	**89.2**	45.3	**79.3**

Table 5 Results (accuracy %) of ablation study on OfficeHome. The best method is emphasized in bold

Method	Pr→Ar	Cl→Pr	Cl→Rw	Ar→Cl	Avg
$\mathcal{L}_{cls}$	52.4	63.8	66.4	45.6	61.3
$\mathcal{L}_{cls}$ + orig.OT	55.7	69.8	72.9	49.4	65.2
$\mathcal{L}_{cls} + \alpha\mathcal{L}_d$	59.8	72.1	74.8	51.3	67.8
$\mathcal{L}_{cls} + \alpha\mathcal{L}_d + \beta\mathcal{L}_c$	**60.5**	**73.5**	**75.2**	**53.2**	**69.2**

we optimized dual variables using an n-dimensional vector instead of a shallow neural network, where n is the number of samples in the corresponding domain. When using vectors, the calculation of inter-domain distance actually does not consider the high-dimensional embedding representations of the samples, but only uses the index of samples in the dataset. We alternately use the vector parameterization method for dual variables u and v, and also evaluate the performance of DDW-OT when u and v are all represented by vectors. Table 7 shows the effect of the neural network and n-dimensional vector parameterization. As we can see, the vector parameterization of a single dual variable is harmful to the performance. However, when both dual variables are vector parameterized, the performance gains some improvement.

4.4 Sensitivity analysis

Sensitive of regularized value ε In this experiment, we evaluate the sensitivity of the ε. Here, ε is a non-negative real number which weighting the regularization term increases. Previous experiments show that when ε is too small, the training of the Dual-OT process cannot converge. We choose ε from the set {0.05, 0.1, 0.2, 0.5, 1, 2}, and conduct on transfer tasks in OfficeHome and ImageCLEF-DA. The performance is shown in Fig. 5(a). As we can see, the adaptation performance is stable under the different value of ε, and achieve slightly better results when $\varepsilon = 1$.

Sensitive of decomposed-distance λ Here we evaluate the sensitivity of λ in our experiments. λ is a trade-off parameter across the target domain spatial structure M_{unc} and the intra-domain category association M_{cor}. Here we choose λ from {0, 0.1, 0.3, 0.5, 0.7, 0.9, 1}. Figure 5(b)(c) shows the model's performance under different values. For the transfer task A→W in Office-31, the increasing weight for M_{unc} is beneficial to the optimization, while the opposite result is obtained in task W→A. We assume the reason that M_{cor} is relatively more stable and less likely to be affected by a single sample's inaccurate representation compared with M_{unc}. Therefore, in difficult tasks, given more concentrate on the inaccurate M_{unc} distance caused by the clustering error may deteriorate the performance.

Sensitive of trade-off parameters α, β The parameters α and β make a trade-off between the domain discrepancy loss and the class alignment loss. We fix one of the values $\alpha(\beta)$ and perform model under changing another value $\beta(\alpha)$. The experiment is performed on several transfer tasks, P→I, C→P from ImageCLEF-DA, Cl→Pr from OfficeHome, and D→A from Office-31. The performance is shown in

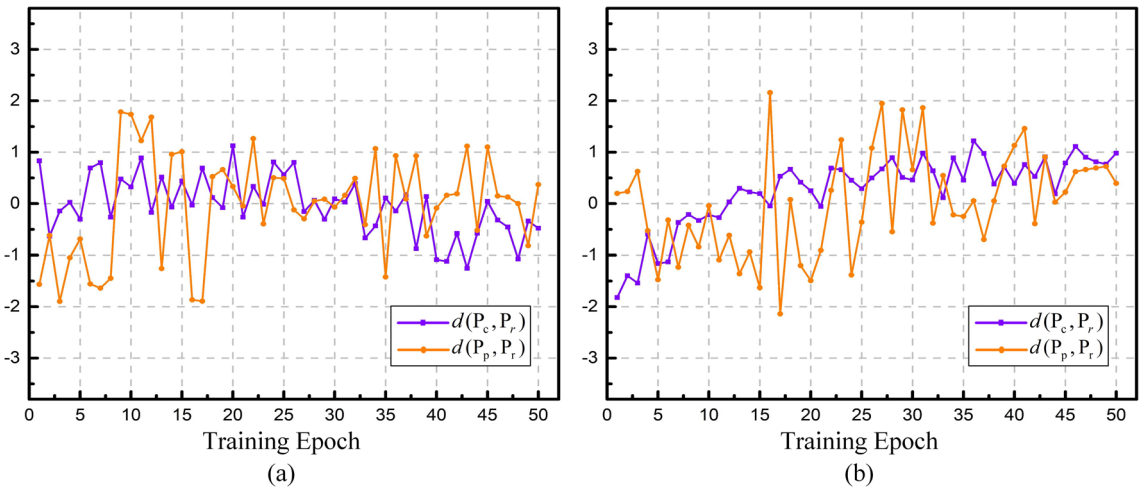

Fig. 4 The distance between estimate to the real. Experiment on two tasks, W→A from Office-31 and Ar→Cl from OfficeHome, respectively

Table 6 The effect of clustering-based prototype generation (CPG) compared with the traditional pseudo label-based prototype generation (PPG)

Method	A→W	D→W	A→D	W→A	Avg
CPG	92.1	100	90.8	68.5	87.6
PPG	90.5	98.7	88.1	67.8	85.2

Results (accuracy %) on Office-31 are reported

Fig. 6(b)(c). As we can see, the accuracy decreases when given extremely high value to α (β), and relatively satisfying performance is achieved when α=0.5/1 and β=0.1/0.5. For consistency, we set α=1, β=0.1 for all transfer tasks.

The effect of the network depth for dual variables As mentioned earlier, we use fully-connected layers ($d \rightarrow 1024(n) \rightarrow 1$) to parameterize the dual variables u and v. Figure 6(a) presents the results under different settings. As can be seen, DDW-OT achieves relatively stable results when the dual network depth changes, and the adaptation ability can not be continuous improved with the increase of depth. At the same time, a deeper network also brings more parameters and higher computational complexity. Therefore, in all other experiments of this paper, we set the network depth n=3 for the dual variables.

4.5 Visualization and training convergence

Training convergence We evaluate the training stability of DDW-OT on two tasks, i.e., A→W from Office-31 and Pr→Rw from OfficeHome. The results are presented in Fig. 7, including the training loss, the classification accuracy of both domains, and the clustering-based accuracy on the target domain within each training epoch. The first ten epochs include the pre-training stage of the classification network and the dual-OT network (before the orange dash line). As we can see, during the pre-training stage, the training loss drops rapidly. The classification accuracy of both domains has already reached a relatively high level, gradually improved and tends to be stable in the following fine-tuning stage. For the clustering-based classification accuracy on the target samples, it is all about fluctuating in the whole training process, but because DDW-OT construct decomposed-distance on this basis instead of relying solely on the classification accuracy of this part, it still plays a positive role in the training process.

Feature visualization We visualize the deep feature of the last hidden layer by utilizing t-SNE [32] to illustrate the feature transferability of our method. We perform t-SNE on task P-C from ImageCLEF-DA and task Pr-Rw from OfficeHome. The features in Fig. 8(a)(c) are derived from the source-only model. We can see that the features in the source domain have a relatively obvious cluster structure. In contrast, the features of the target domain do not form clear category boundary and can not be discriminative well. As we can see, the adaptation process of DDW-OT can make the target domain features much more compact and well separated, shown in Fig. 8(b)(d). The above observations suggest that our method is able to learn the domain invariant features and reduce the intra-class variations.

5 Conclusion

In this paper, we propose a decomposed-distance weighted optimal transport method to perform sample-level alignment for UDA. To achieve better distance measurement between domains, we design a new reweighing matrix. The combination of the two parts of distance considers the spatial information of the target domain and analyzes the correlation degree of category between domains. The extensive experiments on several benchmark datasets illustrated the effectiveness of our method. Although DDW-OT makes the alignment of both domains at sample level, the accuracy of the transport plan is still affected by the inconsistent sample categories, which are caused by the class distribution shift in both domains during the batch-wise training. To further remove the above restriction, the optimal partial transport theory or the specific sample selection strategy

Table 7 The effect of the neural network (NN) and the n-dimensional vector (Vec.) parameterization for dual variables

u		v		P→I	C→P	P→C	I→P	Avg
NN	Vec.	NN	Vec.					
	✓		✓	92.1	75	94.7	78	87.6
✓			✓	91.8	74.1	93.6	77.8	86.8
	✓	✓		91.2	73.3	94.4	77.9	87.1
✓		✓		92.5	80.2	96.7	82.6	90.7

Results (accuracy %) on ImageCLEF-DA are reported

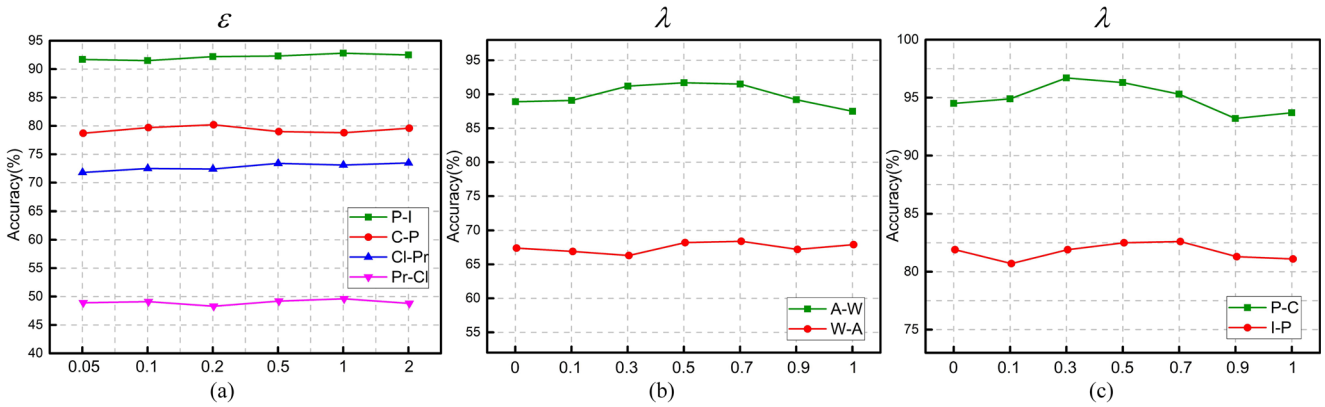

Fig. 5 (a)Sensitivity of the regularized value ε. (b)(c)Sensitivity of the trade-off parameter λ. Validated on several tasks under different values

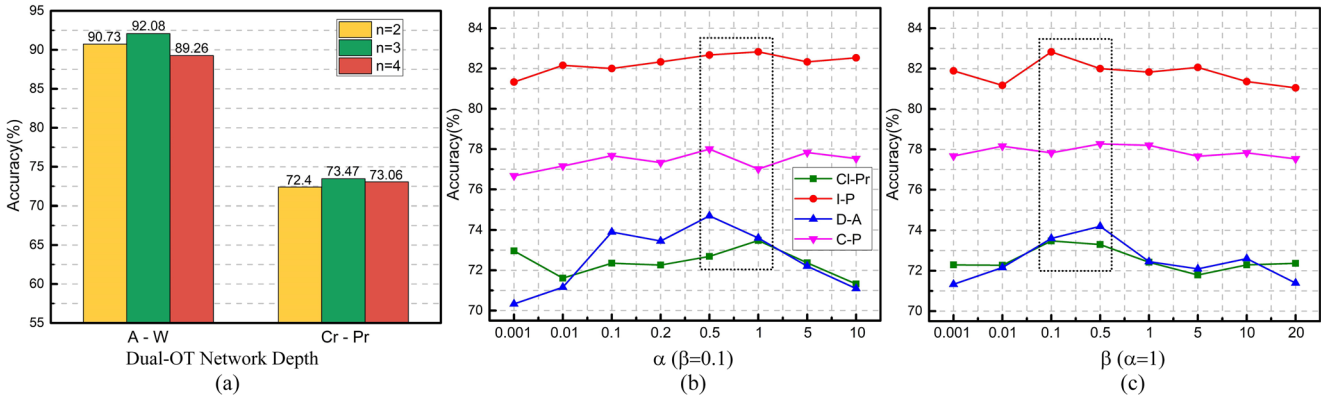

Fig. 6 The effect of the network depth for Dual-OT network and trade-off parameters α, β experimented on several tasks

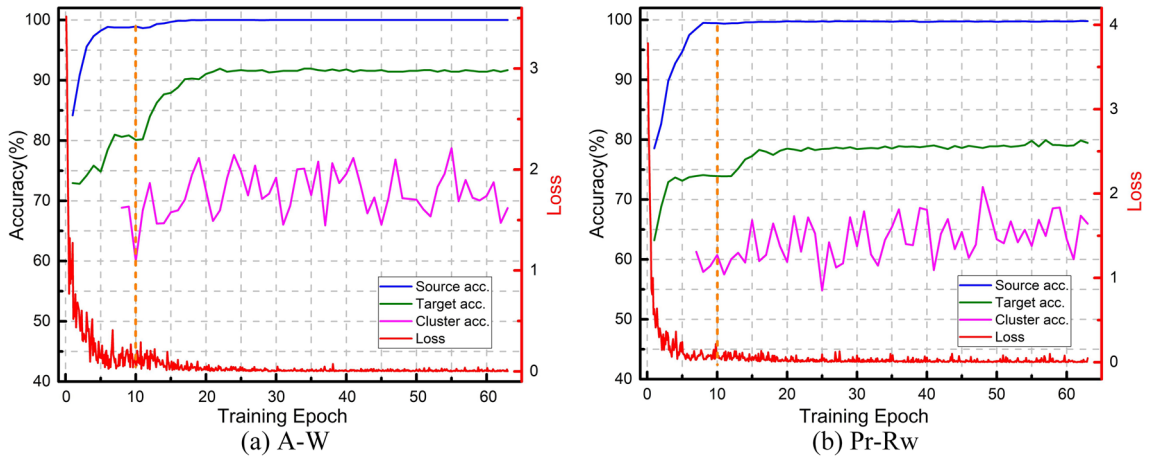

Fig. 7 The evaluation of the training stability, experiments on two tasks. Best viewed in color

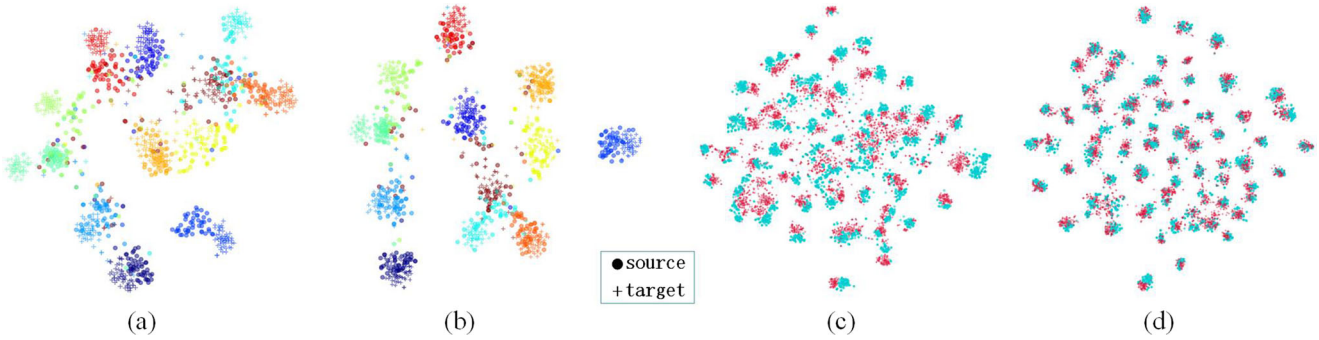

Fig. 8 The t-SNE visualization. (a)(b) are generated from class information of P→C in ImageCLEF-DA. Each color reflects a category. (c)(d) are generated from domain information of Pr→Rw in OfficeHome. Blue and red shapes represent samples from source and target domain, respectively

may be considered, so as to obtain a more reliable OT matrix in an implicit way.

Declarations

Ethics Statement This article does not contain any studies with human participants or animal performed by any of the authors.

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M (2014) Domain-adversarial neural networks. arXiv:1412.4446
2. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein gan
3. Asad M, Jiang H, Yang J, Tu E, Malik AA (2021) Multi-stream 3d latent feature clustering for abnormality detection in videos. Appl Intell pp 1–18
4. Aude G, Cuturi M, Peyré G, Bach F (2016) Stochastic optimization for large-scale optimal transport. arXiv:1605.08527
5. Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW (2010) A theory of learning from different domains. Machine Learning 79(1):151–175
6. Ben-David S, Blitzer J, Crammer K, Pereira F et al (2007) Analysis of representations for domain adaptation. Advances in Neural Information Processing Systems 19:137
7. Blondel M, Seguy V, Rolet A (2018) Smooth and sparse optimal transport. In: International conference on artificial intelligence and statistics. PMLR, pp 880–889
8. Chen C, Chen Z, Jiang B, Jin X (2019) Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 3296–3303
9. Chen C, Xie W, Huang W, Rong Y, Ding X, Huang Y, Xu T, Huang J (2019) Progressive feature alignment for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 627–636
10. Courty N, Flamary R, Tuia D, Rakotomamonjy A (2017) Optimal transport for domain adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(9):1853–1865. https://doi.org/10.1109/TPAMI.2016.2615921
11. Cuturi M (2013) Sinkhorn distances: lightspeed computation of optimal transport. In: NIPS, vol 2, p 4
12. Damodaran BB, Kellenberger B, Flamary R, Tuia D, Courty N (2018) Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In: Proceedings of the european conference on computer vision (ECCV), pp 447–463
13. Deng Z, Luo Y, Zhu J (2019) Cluster alignment with a teacher for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9944–9953
14. Gao B, Yang Y, Gouk H, Hospedales TM (2020) Deep clustering for domain adaptation. In: ICASSP 2020-2020 IEEE International conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4247–4251
15. Ghifary M, Kleijn WB, Zhang M, Balduzzi D, Li W (2016) Deep reconstruction-classification networks for unsupervised domain adaptation. In: European conference on computer vision. Springer, pp 597–613
16. Gholenji E, Tahmoresnezhad J (2020) Joint discriminative subspace and distribution adaptation for unsupervised domain adaptation. Appl Intell 50(7):2050–2066
17. Hartigan JA, Wong MA (1979) A k-means clustering algorithm. Appl Stat 28(1)
18. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
19. Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. Comput Sci 14(7):38–39
20. Hu C, He S, Wang Y (2021) A classification method to detect faults in a rotating machinery based on kernelled support tensor machine and multilinear principal component analysis. Appl Intell 51(4):2609–2621
21. Hu C, Wang Y, Gu J (2020) Cross-domain intelligent fault classification of bearings based on tensor-aligned invariant subspace learning and two-dimensional convolutional neural networks. Knowl-Based Syst 209:106214
22. Jiang X, Lao Q, Matwin S, Havaei M (2020) Implicit class-conditioned domain alignment for unsupervised domain adaptation. In: International conference on machine learning. PMLR, pp 4816–4827
23. Kang G, Jiang L, Yang Y, Hauptmann AG (2019) Contrastive adaptation network for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4893–4902
24. Kantorovich L (1942) On the transfer of masses (in russian). In: Doklady akademii nauk, vol 37, pp 227–229
25. Kerdoncuff T, Emonet R, Sebban M (2020) Metric learning in optimal transport for domain adaptation
26. Li M, Zhai YM, Luo YW, Ge PF, Ren CX (2020) Enhanced transport distance for unsupervised domain adaptation. In:

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13936–13944
27. Liakos KG, Busato P, Moshou D, Pearson S, Bochtis D (2018) Machine learning in agriculture: a review. Sensors 18(8):2674
28. Long M, Cao Y, Wang J, Jordan M (2015) Learning transferable features with deep adaptation networks. In: International conference on machine learning. PMLR, pp 97–105
29. Long M, Cao Z, Wang J, Jordan MI (2017) Conditional adversarial domain adaptation. arXiv:1705.10667
30. Long M, Wang J, Ding G, Sun J, Yu PS (2014) Transfer joint matching for unsupervised domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1410–1417
31. Long M, Zhu H, Wang J, Jordan MI (2017) Deep transfer learning with joint adaptation networks. In: International conference on machine learning. PMLR, pp 2208–2217
32. Van der Maaten L, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Rese 9(11)
33. Maxwell AE, Warner TA, Fang F (2018) Implementation of machine-learning classification in remote sensing: an applied review. Int J Remote Sens 39(9):2784–2817
34. Munkres J (1962) Algorithms for the assignment and transportation problems. SIAM J, 10
35. Pan Y, Yao T, Li Y, Wang Y, Ngo CW, Mei T (2019) Transferrable prototypical networks for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2239–2247
36. Perrot M, Courty N, Flamary R, Habrard A (2016) Mapping estimation for discrete optimal transport. In: Proceedings of the 30th international conference on neural information processing systems, pp 4204–4212
37. Peyré G., Cuturi M et al (2019) Computational optimal transport: With applications to data science. Foundations and Trends® in Machine Learning 11(5-6):355–607
38. Redko I, Habrard A, Sebban M (2017) Theoretical analysis of domain adaptation with optimal transport. In: Joint european conference on machine learning and knowledge discovery in databases. Springer, pp 737–753
39. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
40. Saenko K, Kulis B, Fritz M, Darrell T (2010) Adapting visual category models to new domains. In: European conference on computer vision. Springer, pp 213–226
41. Saito K, Watanabe K, Ushiku Y, Harada T (2018) Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3723–3732
42. Sankaranarayanan S, Balaji Y, Castillo CD, Chellappa R (2018) Generate to adapt: Aligning domains using generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8503–8512
43. Seguy V, Damodaran BB, Flamary R, Courty N, Rolet A, Blondel M (2018) Large-scale optimal transport and mapping estimation. In: ICLR 2018-International conference on learning representations, pp 1–15
44. Shen J, Qu Y, Zhang W, Yu Y (2018) Wasserstein distance guided representation learning for domain adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32
45. Song L, Wang C, Zhang L, Du B, Zhang Q, Huang C, Wang X (2020) Unsupervised domain adaptive re-identification: theory and practice. Pattern Recogn 102:107173
46. Sun B, Saenko K (2016) Deep coral: Correlation alignment for deep domain adaptation. In: European conference on computer vision. Springer, pp 443–450
47. Tzeng E, Hoffman J, Saenko K, Darrell T (2017) Adversarial discriminative domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7167–7176
48. Tzeng E, Hoffman J, Zhang N, Saenko K, Darrell T (2014) Deep domain confusion: Maximizing for domain invariance. arXiv:1412.3474
49. Venkateswara H, Eusebio J, Chakraborty S, Panchanathan S (2017) Deep hashing network for unsupervised domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5018–5027
50. Wang Y, Ye H, Cao F (2021) A novel multi-discriminator deep network for image segmentation. Appl Intell (12)
51. Xiao N, Zhang L (2021) Dynamic weighted learning for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 15242–15251
52. Xu R, Liu P, Wang L, Chen C, Wang J (2020) Reliable weighted optimal transport for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4394–4403
53. Zhang B, Qian J (2020) Autoencoder-based unsupervised clustering and hashing. Appl Intell (8)
54. Zhang T, Wang H, Du W, Li M (2021) Deep cnn-based local dimming technology. Appl Intell (1)
55. Zhang Y, Deng B, Tang H, Zhang L, Jia K (2020) Unsupervised multi-class domain adaptation: theory, algorithms, and practice. IEEE Trans Pattern Anal Mach Intell
56. Zhang Y, Liu T, Long M, Jordan M (2019) Bridging theory and algorithm for domain adaptation. In: International conference on machine learning. PMLR, pp 7404–7413

Bilin Wang received the B.S. degree from the College of Computer Science and Technology, Jilin University, in 2017, where she is currently pursuing the Ph.D. degree. Her main research interests include the application of optimal transport and domain adaptation.

Shengsheng Wang received the B.S., M.S., and Ph.D. degrees in Computer Science from Jilin University, in 1997, 2000, and 2003, respectively. He is currently a Professor with the College of Computer Science and Technology, Jilin University. His current research interests include the areas of computer vision, deep learning, and data mining.

Zhe Zhang received the M.S. degree from the College of Computer Science and Technology from Jilin University, Jilin, China, in 2018, where he is currently pursuing the Ph.D. degree. He is also a member of the Key Laboratory for Symbol Computation and Knowledge Engineering of National Education Ministry, China. His main research interests include computer vision, image processing and deep learning.

Xin Zhao received the B.S. degree from the College of Computer Science and Technology, Jilin University, in 2016, where he is currently pursuing the Ph.D. degree. His current research interests include deep learning, transfer learning, and image processing.

Zihao Fu received the M.S. degree from the College of Computer and Information Engineering, Henan Normal University, in 2020. He is currently pursuing the Ph.D. degree in the College of Computer Science and Technology, Jilin University. His current research interests include casual reasoning and domain adaptation.