Applied Intelligence (2023) 53:7862–7875
https://doi.org/10.1007/s10489-022-03805-9

Open-set domain adaptation by deconfounding domain gaps

Xin Zhao[1,2] · **Shengsheng Wang**[1,2] · **Qianru Sun**[3]

Accepted: 23 May 2022 / Published online: 27 July 2022

Abstract

Open-Set Domain Adaptation (OSDA) aims to adapt the model trained on a source domain to the recognition tasks in a target domain while shielding any distractions caused by open-set classes, i.e., the classes "unknown" to the source model. Compared to standard DA, the key of OSDA lies in the separation between known and unknown classes. Existing OSDA methods often fail the separation because of overlooking the confounders (i.e., the domain gaps), which means their recognition of "unknown classes" is not because of class semantics but domain difference (e.g., styles and contexts). We address this issue by explicitly deconfounding domain gaps (DDP) during class separation and domain adaptation in OSDA. The mechanism of DDP is to transfer domain-related styles and contexts from the target domain to the source domain. It enables the model to recognize a class as known (or unknown) because of the class semantics rather than the confusion caused by spurious styles or contexts. In addition, we propose a module of ensembling multiple transformations (EMT) to produce calibrated recognition scores, i.e., reliable normality scores, for the samples in the target domain. Extensive experiments on two standard benchmarks verify that our proposed method outperforms a wide range of OSDA methods, because of its advanced ability of correctly recognizing unknown classes.

Keywords Open-set domain adaptation · Image classification

1 Introduction

Deep learning has made a remarkable success in a wide range of computer vision tasks [1–3], given a large amount of annotated training data. However, deep models can not generalize well to novel domains due to the domain shift [4]. To adapt these models, people always have to collect and annotate a large volume of training samples in the target domain as well, which is costly.

Unsupervised Domain adaptation (UDA) [5] tackles this issue by transferring knowledge from a source domain to a related but different domain (target domain) through using only unlabeled data. Most of UDA algorithms assume that the source and target datasets cover identical categories, known as Closed-Set Domain Adaptation (CSDA), as shown in Fig. 1a. While this assumption does not stand in real applications, as it is not possible to guarantee two domains sharing the same label space if no labels are available in one domain (the target domain). Therefore, researchers come up with a more reasonable and realistic setting called Open-Set Domain Adaptation (OSDA) [6–8]. The mainstream setting was introduced by Saito et al. [7], where the classes in the source domain are fully known and some of the classes in the target domain are unknown to the model trained in the source domain, as shown in Fig. 1b. The methods for OSDA specifically aim to classify the target domain samples correctly either into the label space of the source domain or as a special class called "unknown".

The key in OSDA lies in how to effectively recognize and isolate the unknown samples, compared to the DA in closed-set scenarios. Existing methods usually define

✉ Qianru Sun
qianrusun@smu.edu.sg

Xin Zhao
focusxin@outlook.com

Shengsheng Wang
wss@jlu.edu.cn

[1] College of Computer Science and Technology, Jilin University, 2699 Qianjin Street, Changchun, 130012, Jilin, China

[2] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, 2699 Qianjin Street, Changchun, 130012, Jilin, China

[3] School of Computing and Information Systems, Singapore Management University178903, Singapore, Singapore

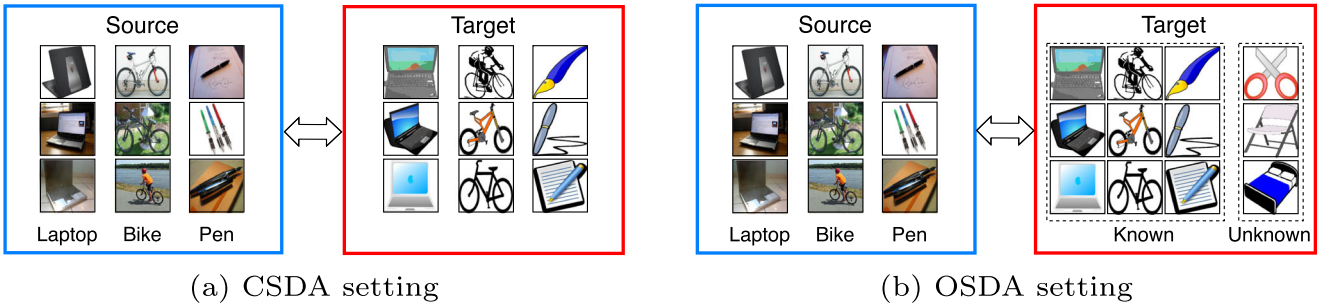


Fig. 1 A comparison between the CSDA setting and the OSDA setting. (a) The CSDA setting assumes that the label space of two domains is identical. (b) The OSDA setting assumes that the target domain includes both known (shared) classes and unknown classes

the normality score whose value shows to be lower for unknown sample than known sample. There are two typical issues. First, the model producing such scores is trained solely on source datasets, overlooking the confounder, i.e., the domain gap between source and target datasets. The essential reason behind is that when the model learns the semantic information of shared (known) classes, it is misled by spurious image styles or contexts. For example, if there are many samples of "dog" on grass in source domain and "sheep" on grass in target domain (while there are quite fewer "dog" on grass in the target domain), the model gets misled by the context "on grass" and thus makes the wrong prediction on "sheep" samples as "dog" [9]. The second issue is that the model usually produces a single uncalibrated prediction on each input data, making the recognition of unknown samples unstable or unreliable.

In this paper, we solve the above issues in the two-stage framework presented in Fig. 2. In the first stage, we improve the ability and stability of the model to separate known and unknown samples. 1) We propose an explicit module of deconfounding domain gaps (DDP), which transfers image styles and contexts from the target domain to the source domain. We then fine-tune the model on the source samples with transferred styles and contexts to enable it to recognize target samples as known (or unknown) because of their class semantics (rather than the confusion caused by spurious styles or contexts). 2) We propose a module of ensembling multiple transformations (EMT), which

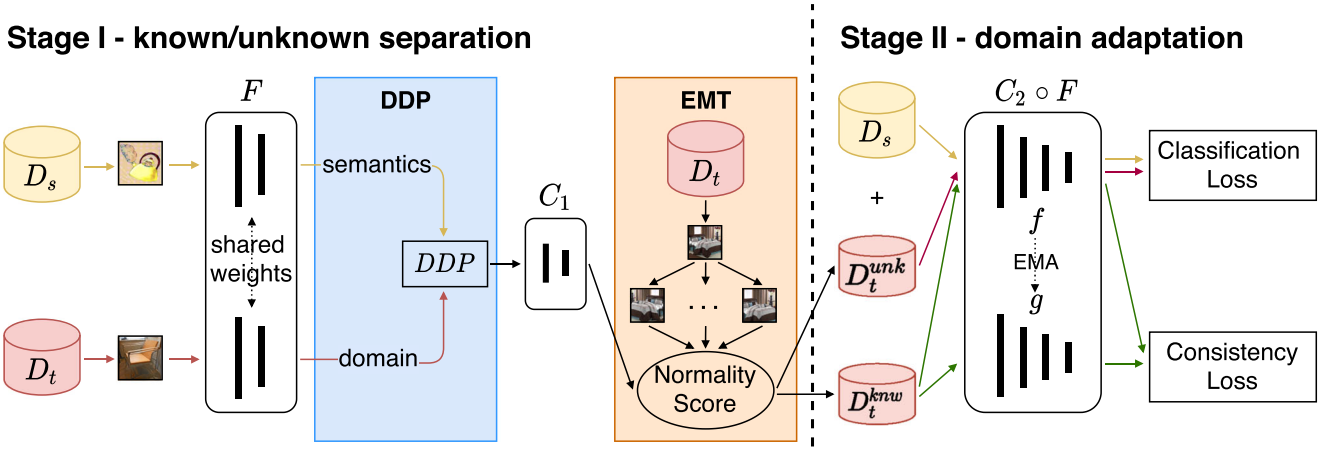


Fig. 2 An illustration of the proposed method. Stage I: we propose an explicit module of deconfounding domain gaps (DDP), which transfers domain information (i.e., image styles and contexts) from the target domain to the source domain. We then train the encoder F and semantic network C_1 on the source samples with transferred styles and contexts to enable the model to recognize a class as known (or unknown) because of the class semantics rather than the confusion caused by spurious styles or contexts. After convergence, we propose a module of ensembling multiple transformations (EMT), which calibrates the model predictions by ensembling predictions from multiple transformations of each target sample. Based on the calibrated predictions, we can compute reliable normality scores used to divide the target datasets into a known target dataset $\mathcal{D}_t^{knw}$ and an unknown target dataset $\mathcal{D}_t^{unk}$. Stage II: two networks with the same architecture are used: a student network $f = C_2 \circ F$, and a teacher network g with its weights being automatically set as an exponential moving average (EMA) of weights of the student network f. We minimize the consistency loss to mitigate the domain discrepancy between the source dataset $\mathcal{D}_s$ and the target known dataset $\mathcal{D}_t^{knw}$. In addition, we can classify the target known samples and reject the target unknown samples by minimizing the classification loss on source dataset $\mathcal{D}_s$ and unknown target dataset $\mathcal{D}_t^{unk}$. We also utilize the proposed DDP to further deconfound domain gaps, which is not shown in the right side of the figure for simplicity

calibrates the model predictions by ensembling predictions from multiple transformations of each target sample. In the second stage, we leverage both the self-ensembling method [10] and the proposed DDP to deconfound domain gaps. Finally, we get the model that can recognize each target sample either as one of the known classes or as the special class "unknown". We conduct extensive experiments on two OSDA benchmarks and show that both modules contribute to the performance improvement of the trained models. Therefore, our main contributions are in three folds.

1. We point out that separating known and unknown classes remains a challenging problem in OSDA due to the confusion caused by domain gaps. We propose a novel OSDA method that can perform effective separation.
2. We introduce an explicit module of deconfounding domain gaps (DDP), which transfers image styles and contexts from the target domain to the source domain and enables the model to correctly recognize unknown samples without being confounded by domain gaps. In addition, we propose a module of ensembling multiple transformations (EMT) to calibrate the recognition of the model and get more reliable normality scores.
3. We conduct experiments on two standard OSDA benchmarks. Our results demonstrate that our method outperforms the state-of-the-art. Our in-depth analyses verify that it gets better results because of its advanced ability of separating between known and unknown samples in the target domain.

2 Related works

In this section, we briefly review methods for domain adaptation and anomaly detection.

2.1 Domain adaptation

Closed-Set Domain Adaptation (CSDA) works with the assumption that two domains have identical categories. CSDA approaches focus on mitigating the domain discrepancy between domains and can be grouped into several categories based on the adopted strategy. *Discrepancy-based* methods measure the divergence between domains in the feature space with a discrepancy metric, such as Maximum Mean Discrepancy (MMD) [11], Higher-order moment matching (HoMM) [12], and Wasserstein distance [13]. The domain shift will be reduced by minimizing the metric during training. *Adversarial* methods [14] leverage a domain classifier to distinguish source and target features, while training the feature encoder to device the domain classifier in order to extract domain-agnostic feature representations. Adversarial methods are the most popular ones and have obtained promising performance, which is further enhanced by recent works [15–18] with novel network designs. *Generative* methods [19, 20] leverage generative models to translate source samples to the target dataset and then reduce the domain discrepancy in both feature and pixel levels. *Self-supervised* methods [21, 22] design auxiliary self-supervised tasks for unlabeled target data to learn robust cross-domain representations. Consistency-enforcing methods [10, 23] force the model to make similar predictions for unannotated target samples even after they have been augmented. *Feature disentanglement* methods [24–26] decouple the feature representations into domain-invariant and domain-specific parts, and only the former is used to predict the target labels. Our proposed DDP shares a similar spirit with this line of methods, as we also try to separate the domain-related representation (style/context) from the semantic representation. However, we intend to transfer the styles from the target domain to the source domain instead of only using the semantic representation. In addition, feature disentanglement methods for CSDA cannot be used in the OSDA scenario, as they need to exploit adversarial training to enhance the domain-invariant part, which may incur negative transfer. In comparison, our proposed DDP explicitly exploits feature statistics as the style representation, so we can also transfer the styles from unknown target samples without being interfered with by their semantic information.

Open-Set Domain Adaptation (OSDA) assumes target label set contains source label set. There are two different settings in the OSDA literature. Busto et al. [6] assumed each domain includes unknown categories besides the shared classes. And they proposed an algorithm called Assign-and-Transform-Iteratively (AIT), which maps target data to source domain and then utilizes SVMs for final prediction. Saito et al. [7] eased the setting by requiring no unknown data from the source dataset, so target dataset contains all the source classes (known) and additional private classes that do not belong to the source (unknown). They also proposed a method, called Open Set Back-Propagation (OSBP), which adversarially trains a classifier with an extra 'unknown' class to achieve common-private separation. Later OSDA methods all follow this more challenging and realistic setting. Separate To Adapt (STA) [8] aims to conduct known and unknown separation through a coarse-to-fine filtering process which includes two stages. First, multiple binary classifiers will be trained to compute the similarity score between source and target data. Second, target samples with very low and high scores will be selected to train a final binary recognizer to distinguish known and unknown target samples. Attract or Distract (AoD) [27] leverage metric learning to match target samples

with the corresponding neighborhood or distract away from the known classes. Rotation-based Open Set (ROS) [28] adopts rotation classification, a self-supervised method, to distinguish the known and unknown target samples and then adapt source knowledge to the target known data.

Universal domain adaptation (UniDA), as a more general scenario, makes no assumption about the relationship of label sets between two domains. Universal Adaptation Network (UAN) [29] designs a measurement to evaluate sample-level transferability based on domain similarity and prediction uncertainty. Then samples with high transferability will be used with higher weight to promote common-class adaptation. However, as pointed by [30], this criterion is not discriminative and robust enough. Fu et al. [30] designed a better measurement that combines confidence, entropy, and consistency using multiple auxiliary classifiers to measure sample-wise uncertainty. Similarly, a class-level weighting strategy is applied for subsequent adversarial adaptation.

2.2 Anomaly detection

Anomaly detection targets at detecting out-of-distribution (anomalous) samples by learning from normal samples. The approaches in this direction can be grouped into three categories. *Distribution-based* approaches [31, 32] leverage the normal samples to model the distribution function so that anomalous samples with lower likelihood can be filtered out. *Reconstruction-based* approaches [33, 34] leverage the encoder-decoder network to reconstruct the normal training samples. Then anomalous samples can be recognized as they have larger reconstruction error compared with normal samples. *Discriminative* approaches [35, 36] train a classifier on the normal samples and directly recognize anomalous samples based on the model prediction.

3 Method

In this section, we first formally introduce the preliminaries, then we present an overview of the proposed method and describe it in detail.

3.1 Preliminaries

We denote the annotated source domain drawn from distribution p_s as $\mathcal{D}_s = \left\{\left(x_j^s, y_j^s\right)\right\}_{j=1}^{N_s} \sim p_s$ and the unannotated target domain drawn from distribution p_t as $\mathcal{D}_t = \left\{x_j^t\right\}_{j=1}^{N_t} \sim p_t$. In OSDA, target label set $\mathcal{C}_t$ contains source label set $\mathcal{C}_s$, i.e., $\mathcal{C}_s \subset \mathcal{C}_t$. We refer to classes from $\mathcal{C}_s$ as the known classes and classes from $\mathcal{C}_t \backslash \mathcal{C}_s$ as the unknown classes. In OSDA, we both have $p_s \neq p_t$ and $p_s \neq p_t^{\mathcal{C}_s}$, where $p_t^{\mathcal{C}_s}$ represents the distribution of the target known data. Thus, we encounter both domain shift $\left(p_s \neq p_t^{\mathcal{C}_s}\right)$ and class shift $(\mathcal{C}_s \neq \mathcal{C}_t)$ problems in OSDA. The goal of OSDA methods is to classify target known data correctly and reject target unknown data.

OSDA introduces two challenges: negative transfer and known/unknown separation. (1) Enforcing to match the whole distribution of two domains as done in closed-set scenario will incur negative transfer, as the unknown target samples will also align mistakenly with source data. To solve this problem, we need to apply adaptation only to the shared $\mathcal{C}_s$ categories, mitigating the domain shift between p_s and $p_t^{\mathcal{C}_s}$. (2) Thus, we encounter the second challenge: known/unknown separation. All target samples should be recognized from target private categories $\mathcal{C}_t \backslash \mathcal{C}_s$ (unknown) or the shared categories $\mathcal{C}_s$ (known).

3.2 Overview

To handle the aforementioned two challenges, we propose a novel OSDA method with a two-stage structure (Fig. 2): (i) we divide target datasets into known and unknown; (ii) we apply adaptation to source samples and target samples predicted as known. If we consider the unknown samples as anomalies, the first stage can be seen as an anomaly detection issue. And we can also treat the second stage as a CSDA issue between target known and source distributions. Specifically, in the first stage, we propose an explicit module of deconfounding domain gaps (DDP), which transfers image styles and contexts from the target domain to the source domain, eliminating the confounding effect caused by domain gaps. In addition, we propose a module of ensembling multiple transformations (EMT) to calibrate the model predictions. Thus, we can obtain more reliable normality scores based on the calibrated predictions. In the second stage, on the one hand, we leverage both the self-ensembling method [10] and the proposed DDP to reduce the domain discrepancy between the source data and target known data. On the other hand, we train the network to classify target known samples and reject target unknown samples by minimizing the classification loss on source data and unknown target data.

3.3 Deconfounding Domain Gaps (DDP)

Recent domain generalization works observe that image styles and contexts are closely related to visual domains [37, 38]. Inspired by this observation, we propose an explicit module of deconfounding domain gaps (DDP) that transfers image styles and contexts from the target domain to the source domain. It enables the model to recognize a class as

known (or unknown) because of the class semantics rather than the confusion caused by spurious styles or contexts. Following the common practice [39, 40], we use the feature statistics that preserved at the lower layers of the CNN as domain-related representation (i.e., styles and contexts) and their spatial configuration as semantic representation.

For an input sample x, we first obtain its feature maps $z \in \mathbb{R}^{C\times H\times W}$ from the feature encoder, where C indicates the number of channels, H and W represent spatial dimensions. Then we compute the channel-wise mean and standard deviation $\mu(z), \sigma(z) \in \mathbb{R}^C$ as style/context representation:

$$\mu(z) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} z_{hw}, \tag{1}$$

$$\sigma(z) = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(z_{hw}-\mu(z))^2 + \epsilon}. \tag{2}$$

Given intermediate feature maps $z^s, z^t \in \mathbb{R}^{C\times H\times W}$ corresponding to a source sample x^s and a target sample x^t, we replace the style/context of x^s with the style/context of x^t through adaptive instance normalization (AdaIN) [39]:

$$\mathrm{DDP}\left(z^s, z^t\right) = \sigma(z^t)\cdot\left(\frac{z^s-\mu\left(z^s\right)}{\sigma\left(z^s\right)}\right) + \mu(z^t). \tag{3}$$

After training, the learned model can reject unknown samples only based on the semantic content without the influence of the confounder (i.e., the domain gaps).

3.4 Ensembling Multiple Transformations (EMT)

Previous methods usually utilize confidence (i.e., the largest probability of all classes) [7, 8] and uncertainty (i.e., entropy) [28, 29] as the normality score to separate known (normal) and unknown (anomalous) samples, with an assumption that known samples have high confidence and low uncertainty, and vice versa. However, the confidence and the uncertainty used before are based on uncalibrated prediction, meaning they cannot represent the real confidence and uncertainty of the sample. Therefore, all the previous normality scores are not reliable and thus unable to separate known and unknown samples accurately.

To obtain more reliable normality score for separation, we propose a module of **ensembling multiple transformations (EMT)**, which ensembles predictions from multiple transformations of each target sample to calibrate the confidence and the entropy. Specifically, given a target image x^t, we apply random transformations (i.e., random crop and horizontal flip) to it to obtain m augmented samples $\{\tilde{x}_i^t\}_{i=1}^m$. $\hat{y}_i^t = C_1(F(\tilde{x}_i^t)), (i = 1, ..., m)$ is the corresponding prediction of each augmented sample $\tilde{x}_i^t$, where F and C_1 are the feature encoder and the semantic network. We compute the confidence w_{conf} and the entropy w_{ent} as follows:

$$w_{\mathrm{conf}}\left(\hat{y}_i^t\big|_{i=1}^m\right) = \frac{1}{m}\sum_{i=1}^{m}\max\left(\hat{y}_i^t\right), \tag{4}$$

$$w_{\mathrm{ent}}\left(\hat{y}_i^t\big|_{i=1}^m\right) = \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{k=1}^{|\mathcal{C}^s|} -\hat{y}_{ik}^t\log\left(\hat{y}_{ik}^t\right)\right), \tag{5}$$

where $\hat{y}_{ik}^t$ indicates the probability of k-th class and max get the maximum entry in $\hat{y}_i^t$. We unify the w_{conf} and w_{ent} within [0, 1] by the minmax normalization. The formulation of the normality score is:

$$\mathcal{N}\left(x^t\right) = \max\left\{w_{\mathrm{conf}}, 1-w_{\mathrm{ent}}\right\}. \tag{6}$$

We maximize over these two terms to obtain the most reliable measurement.

3.5 Training procedure

Stage I: known/unknown separation. To separate the known and unknown samples of $\mathcal{D}_t$, a CNN is trained on the source samples with transferred styles and contexts. To boost the discriminability of the model and facilitate the following known/unknown separation, we also exploit the label smoothing (LS) as it pushes samples to distribute in tight evenly separated clusters [41]. The network consists of a feature encoder F and a semantic network C_1. We train network by minimizing the following cross-entropy objective:

$$L_{\mathrm{cls}} = -\mathbb{E}_{(x^s,y^s)\in\mathcal{D}_s, x^t\in\mathcal{D}_t}\left[y^{ls}\log C_1\left(\mathrm{DDP}\left(F(x^s), F(x^t)\right)\right)\right], \tag{7}$$

where $y^{ls} = (1-\alpha)y^s + \alpha/|\mathcal{C}_s|$ indicates the smoothed label and α represents the smoothing parameter. After training, we compute the normality score for each target sample using F and C_1 as (6). Known samples have large values of $\mathcal{N}$, and vice versa. The target dataset can be divided into an unknown target dataset $\mathcal{D}_t^{unk}$ and a known target dataset $\mathcal{D}_t^{knw}$ using the normality score. We use the average of the normality score over all target samples $\overline{\mathcal{N}} = \frac{1}{N_t}\sum_{j=1}^{N_t}\mathcal{N}_j$ as the threshold, without the need to introduce any further parameter:

$$\begin{cases} x^t \in \mathcal{D}_t^{knw} & \text{if} \quad \mathcal{N}\left(x^t\right) > \overline{\mathcal{N}} \\ x^t \in \mathcal{D}_t^{unk} & \text{if} \quad \mathcal{N}\left(x^t\right) < \overline{\mathcal{N}}. \end{cases} \tag{8}$$

The detailed process about the computation of $\mathcal{N}$ and the generation of $\mathcal{D}_t^{knw}$ and $\mathcal{D}_t^{unk}$ is described in Algorithm 1.

Algorithm 1 Compute normality score and generate $\mathcal{D}_t^{knw}$ &$\mathcal{D}_t^{unk}$.

Input:
Trained networks F and C_1
Target dataset $\mathcal{D}_t = \left\{x_j^t\right\}_{j=1}^{N_t}$
Output:
Known target dataset $\mathcal{D}_t^{knw} = \left\{x_j^{t,knw}\right\}_{j=1}^{N_{t,knw}}$
Unknown target dataset $\mathcal{D}_t^{unk} = \left\{x_j^{t,unk}\right\}_{j=1}^{N_{t,unk}}$
procedure GETENTROPYSCORE(y)
  return $\sum_{k=1}^{|\mathcal{C}^s|} -y_k \log(y_k)$
end procedure
procedure GETNORMALITYSCORE($F, C_1, \mathcal{D}_t$)
  for each x_j^t **in** $\mathcal{D}_t$ **do**
    Initialize: conf = {}, ent = {}
    for each i **in** {1, ..., m} **do**
      $\tilde{x}_j^t$ = Transform(x_j^t) # Apply transformation to x_j^t
      $\hat{y}_j^t = C_1\left(F\left(\tilde{x}_j^t\right)\right)$
      conf $\leftarrow \max(\hat{y}_j^t)$
      ent $\leftarrow$ getEntropyScore($\hat{y}_j^t$)
    end for
    w_{conf} = mean(conf)
    w_{ent} = mean(ent)
    w_{conf} = normalize(w_{conf}) # Apply the minmax normalization
    w_{ent} = normalize(w_{ent})
    $\mathcal{N} \leftarrow \eta_j = \max\{w_{\text{conf}}, 1 - w_{\text{ent}}\}$
  end for
  return $\mathcal{N}$
end procedure
procedure MAIN( )
  Initialize: $\mathcal{D}_t^{knw} = \{\}, \mathcal{D}_t^{unk} = \{\}$
  $\mathcal{N}$ = getNormalityScore $(F, C_1, \mathcal{D}_t)$
  for each (x_j, η_j) **in** $(\mathcal{D}_t, \mathcal{N})$ **do**
    if $\eta_j \geq$ mean($\mathcal{N}$) **then**
      $\mathcal{D}_t^{knw} \leftarrow \mathbf{x}_j$
    else
      $\mathcal{D}_t^{unk} \leftarrow \mathbf{x}_j$
    end if
  end for
end procedure

Stage II: domain adaptation. The problem is simplified to a CSDA problem after the target unknown data have been filtered out. Without the distraction of $\mathcal{D}_t^{unk}$, we can exploit $\mathcal{D}_t^{knw}$ to decrease the domain discrepancy directly. In addition, $\mathcal{D}_t^{unk}$ can be used to train the classifier to recognize the unknown samples. The network has a similar architecture to that of Stage I, consisting of a feature extractor F and a semantic network C_2. The semantic network C_2 is the same as C_1 except for the last layer: the output dimension of C_1 is $|\mathcal{C}_s|$, while the output dimension of C_2 is $(|\mathcal{C}_s| + 1)$ because of the additional unknown class. We utilize the self-ensembling method [10] to close the domain gap. Two networks with the same architecture are used: a student network $f(x) = C_2(F(x))$, and a teacher network $g(x)$ with its weights being automatically set as an exponential moving average (EMA) of weights of the student network. The student network is trained to minimize the classification loss on source and target unknown samples, while maintaining consistent predictions with the teacher network for target known samples. The loss function of consistency can be formulated as:

$$L_{\text{con}} = \mathbb{E}_{x^t \in \mathcal{D}_t^{knw}}\left[\left(f\left(x^t\right) - g\left(x^t\right)\right)^2\right]. \tag{9}$$

The classification losses for samples from source and target unknown datasets are:

$$L_{\text{cls}}^s = -\mathbb{E}_{(x^s, y^s) \in \mathcal{D}_s, x^t \in \mathcal{D}_t^{knw}}\left[y^s \log C_2\left(\text{DDP}\left(F\left(x^s\right), F\left(x^t\right)\right)\right)\right], \tag{10}$$

$$L_{\text{cls}}^{unk} = -\mathbb{E}_{(x^t, y^t) \in \mathcal{D}_t^{unk}}\left[y^t \log f\left(x^t\right)\right]. \tag{11}$$

It is worth noting that we also exploit the proposed DDP to transfer styles and contexts from the known target datasets to the source datasets, aiming to further deconfound domain gaps. We train the network to minimize the following overall objective:

$$L = (L_{\text{cls}}^s + L_{\text{cls}}^{unk}) + \lambda L_{\text{con}}, \tag{12}$$

where λ is the weight that trades off between classification loss and consistency loss. Once the training is complete, we predict the labels for all target samples using F and C_2.

4 Experiments

In this section, we first introduce the experimental settings including datasets, compared approaches, evaluation metrics, and implementation details. Then, we present classification results on two standard datasets. Finally, we conduct further analysis to verify the effectiveness of the proposed method.

4.1 Experimental settings

Datasets. Office-31 [42] contains images within 31 classes collected from three visually different domains: *Webcam* (**W**) with 795 low-quality images obtained by web camera, *DSLR* (**D**) with 534 high-quality images taken by digital SLR camera, and *Amazon* (**A**) with 2820 images obtained from *amazon.com*. Following the protocol in [7], we set the first 10 categories (1-10) as known and the last 11 categories (21-31) as unknown (in alphabetic order). We

show some example images from Office-31 dataset in Fig. 3a. **Office-Home** [43] contains 15,500 images within 65 classes collected from four different domains, Artistic images (**Ar**), Product images (**Pr**), Clip-Art images (**Cl**), and Real-World images (**Rw**). Following the protocol in [8], we set the first 25 categories (1-25) in alphabetical order as known classes and the remaining 40 categories (26-65) as unknown. Office-Home is much more challenging than Office-31 due to the numerous categories and the large domain discrepancy. Some example images from Office-Home dataset are shown in Fig. 3b.

Compared Approaches. We compare our method with: (1) Source Only model: ResNet-50 [1]; (2) CSDA method: DANN [14]; (3) OSDA methods: STA [8], OSBP [7], and ROS [28]; (4) UniDA method: UAN [29]. For ResNet-50 and DANN, we leverage a confidence threshold to separate known and unknown samples. All the results reported are the average over three random runs.

Evaluation Metrics. OS^* and UNK are two usual metrics used to evaluate OSDA. OS^* denotes the average accuracy on known classes, and UNK denotes the accuracy on the unknown class. They can be combined in $OS = \frac{|\mathcal{C}_s|}{|\mathcal{C}_s|+1} \times OS^* + \frac{1}{|\mathcal{C}_s|+1} \times UNK$ to evaluate the overall performance. However, OS is not an appropriate metric as it assumes the accuracy of each known class has the same importance as the whole "unknown" class. Considering the trade-off between the accuracy of known and unknown classes is important in evaluating OSDA methods, we exploit a metric: $HOS = 2\frac{OS^* \times UNK}{OS^* + UNK}$ [28], which is the harmonic mean of OS^* and UNK. Unlike OS, HOS gives a high score only if the method achieves high performance both for known and unknown data.

Implementation Details. We utilize ResNet-50 [1] pretrained on ImageNet [44] as the backbone network. The feature encoder F consists of the first few blocks of the ResNet architecture (the second residual block and all layers before it), while the remaining part combines the semantic network C_1. Our proposed DDP module is inserted between these two networks. We use the same hyperparameters for each dataset. Following DANN [14], we adjust the learning rate with $lr_p = \frac{lr_0}{(1+\omega p)^{\phi}}$, where p changes from 0 to 1 during the training process, lr_0 equal to 0.01 and 0.003 for Stage I and Stage II respectively, $\omega = 10$, and $\phi = 0.75$. The batch size is 32 for both two stages. For all the pretrained layers, the learning rate is 10 times lower than the layers learned from scratch. We adopt SGD to optimize the network, setting the momentum as 0.9 and the weight decay as 0.0005. In Stage I, we set the smoothing parameter to 0.1 and the number of multi-transformations (m) to 5. In Stage II, the trade-off parameter for consistency loss is $\lambda = 3$. We use the network learned in Stage I as the start for Stage II. The learning rate of the new unknown class is set to two times of the known classes.

4.2 Classification results

To evaluate the performance of the OSDA methods, we focus on the HOS as it can balance the importance between the accuracy of known (OS*) and unknown classes (UNK), as discussed in Section 4.1. For a fair comparison, all results

(a) Office-31 (b) Office-Home

Fig. 3 Example images in Office-31 and Office-Home

of the compared methods are either taken from [28] or obtained by running the code of [45].

Table 1 reports the classification results of Office-31. Our method outperforms all comparison approaches on most tasks except $W \rightarrow D$. Specifically, our method significantly outperforms OSBP by 3.7%. Our method also boosts the HOS of state-of-the-art method ROS by 1.5%. In addition, we observe that DANN, STA, and UAN perform even worse than the ResNet-50 backbone since they suffer from negative transfer caused by the mismatching between the source samples and target unknown samples. The failure of these methods are mainly due to their poor ability to target known and unknown separation.

We also compare our method with previous works on the challenging Office-Home dataset. From Table 2, we can find that our method outperforms all compared methods on a total of 9 out of 12 transfer scenarios, demonstrating that our method works well with large domain gaps. On average, our method achieves the highest performance, 1.8% higher than the second-best method ROS. In addition, Our method outperforms STA and OSBP by a large margin, 6.9% and 3.3% respectively. The encouraging results indicate that our method is very effective for the OSDA setting.

From Tables 1 and 2, we can get one key observation that the advantage of our method is mainly due to its capability in distinguishing known and unknown samples. We can observe that while the average OS* of the compared methods is close to ours, the UNK of our method is much higher, e.g., 2.8% and 3.4% higher than ROS on Office-31 and Office-Home respectively. This observation proves that our method is very significant for separating target known and unknown samples.

4.3 Analysis

Ablation Study. To investigate how our method benefits known/unknown separation, we compare the performance of our Stage I with Stage I of ROS and STA. Both ROS and STA include two stages: they use a multi-rotation classifier and a multi-binary classifier to distinguish known and unknown target samples, respectively. We compute the *area under receiver operating characteristic curve* (AUC-ROC) over the normality scores $\mathcal{N}$ on Office-31 to evaluate the performance. As shown in Table 3, the AUC-ROC of our method (93.0) is higher than that of the multi-rotation used by ROS (91.5) and the multi-binary used by STA (79.9). Table 3 also reports the performance of Stage I when alternatively removing the module of deconfounding domain gaps (No DDP), the module of ensembling multiple transformations (No EMT), and the label smoothing (No LS). The performance of all above cases drops significantly compared to our complete method, verifying each component's importance: (1) the DDP

Table 1 Accuracy (%) of all methods on Office-31 dataset. The best method is emphasized in bold

	Office-31																					
	A → W			A → D			D → W			W → D			D → A			W → A			Avg.		
	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS
ResNet-50 [1] (2016)	67.7	65.9	66.8	78.5	62.8	69.8	93.6	73.0	82.0	98.6	79.3	87.9	58.1	81.0	67.7	56.9	80.6	66.7	75.6	73.8	73.5±1.2
DANN [14] (2016)	93.8	62.9	75.3	88.5	63.3	73.8	98.1	42.7	59.5	99.8	47.9	64.7	68.4	51.5	58.7	71.9	56.7	63.4	86.8	54.2	65.9±1.4
STA [8] (2019)	86.7	67.6	75.9	91.0	63.9	75.0	94.1	55.5	69.8	84.9	67.8	75.2	83.1	65.9	73.2	66.2	68.0	66.1	84.3	64.8	72.5±0.8
OSBP [7] (2018)	86.8	79.2	82.7	90.5	75.5	82.4	97.7	96.7	97.2	99.1	84.2	91.1	76.1	72.3	75.1	73.0	74.4	73.7	87.2	80.4	83.7±0.4
UAN [29] (2019)	95.5	31.0	46.8	95.6	24.4	38.9	99.8	52.5	68.8	81.5	41.4	53.0	93.5	53.4	68.0	94.1	38.8	54.9	93.4	40.3	55.1±1.4
ROS [28] (2020)	88.4	76.7	82.1	87.5	77.8	82.4	99.3	93.0	96.0	100.0	99.4	**99.7**	74.8	81.2	77.9	69.7	86.6	77.2	86.6	85.8	85.9±0.2
Ours	89.5	79.0	**83.9**	86.4	82.7	**84.5**	99.8	96.3	**98.0**	100.0	98.8	99.4	75.2	84.3	**79.5**	70.1	90.2	**78.9**	86.8	88.6	**87.4±0.4**

Table 2 Accuracy (%) of all methods on Office-Home dataset. The best method is emphasized in bold

	Office-Home																				
	Pr → Rw			Pr → Cl			Pr → Ar			Ar → Pr			Ar → Rw			Ar → Cl				
	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS		
ResNet-50 [1] (2016)	70.2	61.0	65.3	32.8	67.1	44.1	45.7	70.5	55.5	64.4	64.0	64.2	76.9	59.7	67.2	44.8	68.7	54.2		
DANN [14] (2016)	77.4	48.4	59.5	50.5	49.9	50.2	61.6	54.5	57.8	71.0	38.9	50.2	75.0	50.1	60.1	54.6	48.8	51.6		
STA [8] (2019)	76.2	64.3	69.5	44.2	67.1	53.2	54.2	72.4	61.9	68.0	48.4	54.0	78.6	60.4	68.3	46.0	72.3	55.8		
OSBP [7] (2018)	76.2	71.7	73.9	44.5	66.3	53.2	59.1	68.1	**63.2**	71.8	59.8	65.2	79.3	67.5	72.9	50.2	61.1	55.1		
UAN [29] (2019)	84.0	0.1	0.2	59.1	0.0	0.0	73.7	0.0	0.0	81.1	0.0	0.0	88.2	0.1	0.2	62.4	0.0	0.0		
ROS [28] (2020)	70.8	78.4	**74.4**	46.5	71.2	56.3	57.3	64.3	60.6	68.4	70.3	69.3	75.8	77.2	76.5	50.6	74.1	60.1		
Ours	69.3	76.9	72.9	48.6	75.6	**59.2**	56.3	68.3	61.7	65.5	79.4	**71.8**	76.4	78.2	**77.3**	50.1	83.9	**62.7**		
	Rw → Ar			Rw → Pr			Rw → Cl			Cl → Rw			Cl → Ar			Cl → Pr			Avg.		
	OS*	UNK	HOS	OS*	UNK	**HOS**	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS
ResNet-50	61.7	63.5	62.5	74.4	58.4	65.4	40.7	54.6	46.7	59.5	68.8	63.8	40.1	76.1	52.5	51.6	67.8	58.6	55.2	65.0	58.3±0.5
DANN	67.3	51.9	58.6	80.8	46.6	59.1	59.5	49.3	53.9	73.5	55.2	63.1	57.6	56.7	57.1	66.2	45.0	53.6	66.3	49.6	56.2±0.6
STA	67.5	66.7	67.1	77.1	55.4	64.5	49.9	61.1	54.5	67.0	66.7	66.8	51.4	65.0	57.4	61.8	59.1	60.4	61.8	63.3	61.1±0.3
OSBP	66.1	67.3	66.7	76.3	68.6	72.3	48.0	63.0	54.5	72.0	69.2	70.6	59.4	70.3	**64.3**	67.0	62.7	64.7	64.1	66.3	64.7±0.2
UAN	77.5	0.1	0.2	85.0	0.1	0.1	66.2	0.0	0.0	80.6	0.1	0.2	70.5	0.0	0.0	74.0	0.1	0.2	75.2	0.0	0.1±0.0
ROS	67.0	70.8	68.8	72.0	80.0	75.7	51.5	73.0	60.4	65.3	72.2	68.6	53.6	65.5	58.9	59.8	71.6	65.2	61.6	72.4	66.2±0.3
Ours	66.8	71.8	**69.2**	72.5	80.1	**76.1**	54.5	74.2	**62.9**	69.4	73.3	**71.3**	54.7	72.1	62.2	63.7	75.3	**69.0**	62.3	75.8	**68.0±0.4**

Table 3 Ablation analysis. The best method is emphasized in bold

STAGE I (AUC-ROC)	A → W	A → D	D → W	W → D	D → A	W → A	Avg.
Ours	91.2	91.1	99.6	99.7	89.9	86.2	93.0
Multi-Rotation (from ROS [28])	90.1	88.1	99.4	99.9	87.5	83.8	91.5
Multi-Binary (from STA [8])	83.2	84.1	86.8	72.0	75.7	78.3	79.9
Ours - No DDP	84.6	83.9	90.8	80.4	81.3	83.5	84.1
Ours - No EMT	88.4	87.9	99.0	99.6	84.7	83.9	90.6
Ours - No LS	89.8	89.1	98.4	99.7	87.5	86.9	91.9
STAGE II (HOS)	A → W	A → D	D → W	W → D	D → A	W → A	Avg.
Ours	83.9	84.5	98.0	99.4	79.5	78.9	**87.4**
Ours Stage I - GRL [14] Stage II	84.6	84.0	98.4	99.3	79.1	76.4	87.0
Ours Stage I - No DDP in Stage II	83.3	83.9	98.2	99.4	78.6	77.8	86.9

module can shield the distractions caused by confounding styles and contexts from source domain during separation; (2) the EMT module can produce reliable normality scores by the calibration from the ensemble; (3) label smoothing is helpful to suppress the overconfident predictions. To verify the efficiency of the self-ensembling method in OSDA, we also compare our method with the wildly adopted GRL [14] based on our Stage I. Table 3 shows that self-ensembling outperforms GRL by 0.4% in average. Furthermore, we also evaluate the role of the DDP in Stage II. As shown in Table 3, our full method outperforms the case when removing the DDP module (No DDP in stage II), which verifies the proposed DDP is also helpful for domain adaptation.

Where to apply DDP? We conduct experiments on Office-31 using ResNet-50 to examine the effect of the position where DDP is applied. Given that the ResNet architecture consists of four residual blocks, we apply DDP to different layers to train different models. For notation, `block1` means DDP is applied after the first residual block, `block2` means DDP is applied after the second residual block, and so on. The results are shown in Fig. 4. We have the following observations: (1) we can get the best performance when DDP is applied after `block2`; (2) DDP is less helpful when applied after too low-level layers (i.e., `block1`); (3) The performance drops significantly when applying DDP after too high-level layers (i.e., `block4`), even worse than when we do not use DDP in Stage-I and Stage-II. This makes sense because `block4` is the closest to the classification layer and is inclined to capture semantic (i.e., label-related) information rather than style/context. As a result, transferring the statistics at `block4` will introduce unwanted semantic information.

Feature Visualization. To intuitively showcase the effectiveness of our method, we visualize features of target samples from the ResNet-50, ROS [28] and our method on the $A \rightarrow D$ task by t-SNE [46]. The features obtained by ResNet-50 can be served as the initial state without adaptation. As shown in Fig. 5a, the features of unknown classes and several known classes mix together, demonstrating that ResNet-50 cannot separate known and unknown classes. In Fig. 5c, our method is capable of separating known and unknown features and discriminating different known classes. Compared with our method, the known and unknown features obtained from ROS (Fig. 5b) appear more confused.

Distribution Discrepancy. As discussed in [47], distribution discrepancy can be measured by the $\mathcal{A}$-distance. It is defined as $d_{\mathcal{A}} = 2(1 - 2\epsilon)$, where ϵ indicates the generalization error of a domain classifier. A larger distribution discrepancy corresponds with a larger $d_{\mathcal{A}}$ and vice versa. We compute $d_{\mathcal{A}}$ for both *source-known* and *known-unknown*: *source-known* represents the distribution discrepancy between source samples and target known samples, and *known-unknown* represents the distribution discrepancy between target known samples and target unknown samples. We compare our method with ResNet-50 and ROS using a kernel SVM as the classifier on two tasks $A \rightarrow W$ and

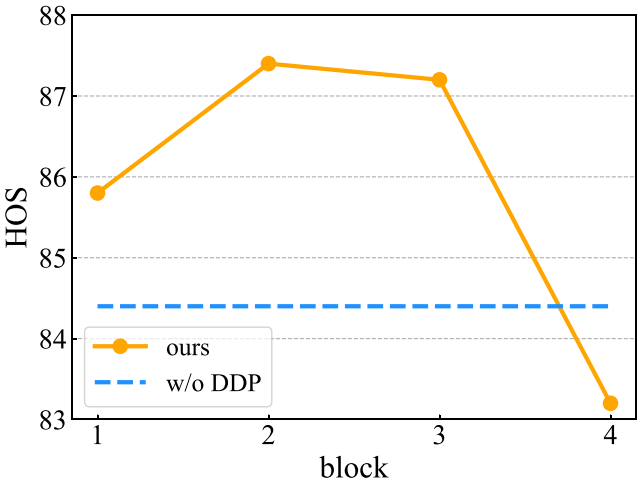


Fig. 4 HOS on Office-31 with applying DDP to different layers

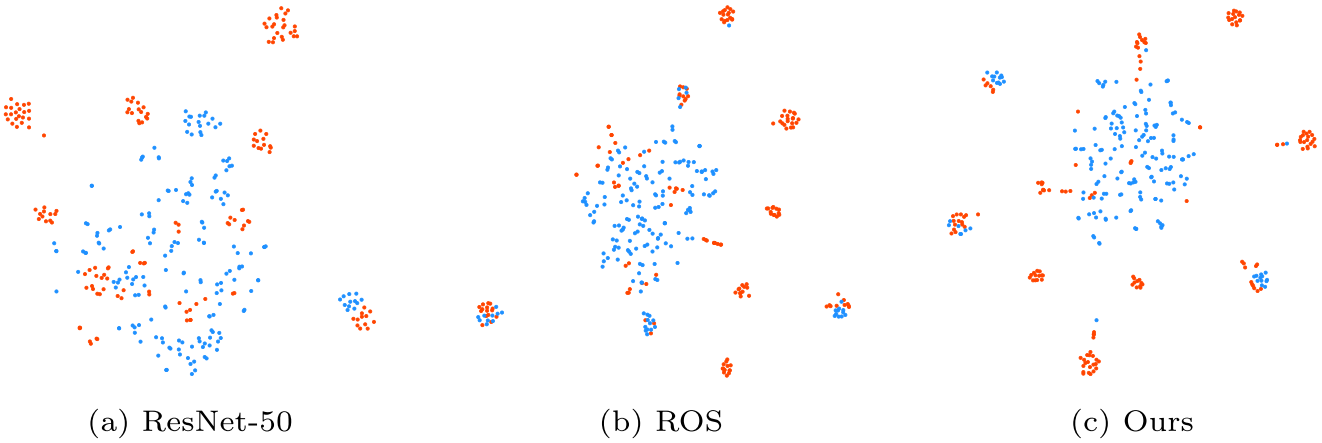


Fig. 5 Visualization of obtained target features for $A \rightarrow D$ using t-SNE. Known and unknown samples are denoted as red and blue points respectively

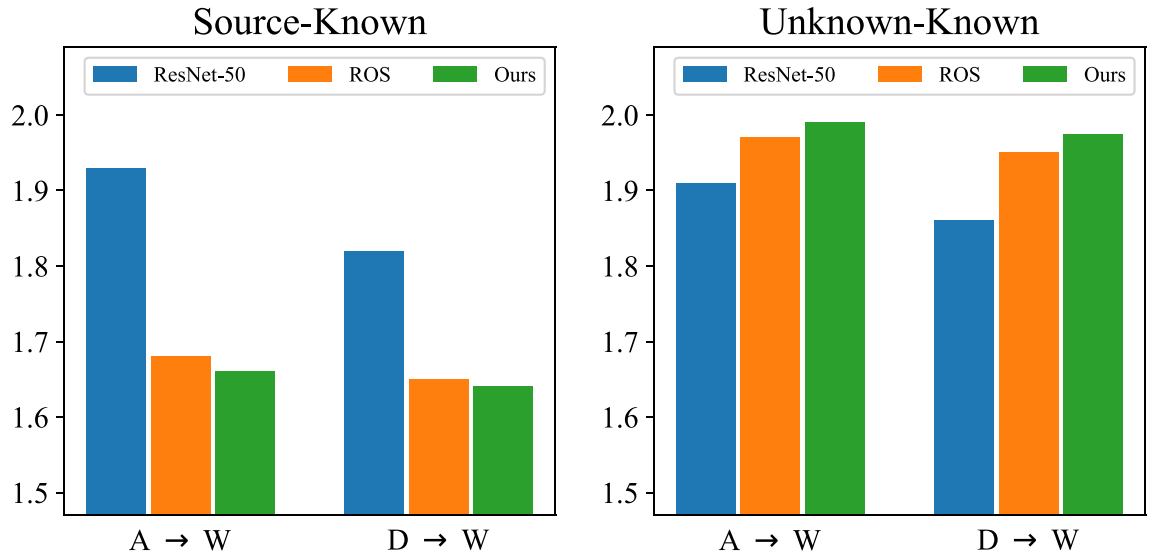


Fig. 6 The value of $\mathcal{A}$-distance for source-known (smaller is better) and unknown-known (larger is better)

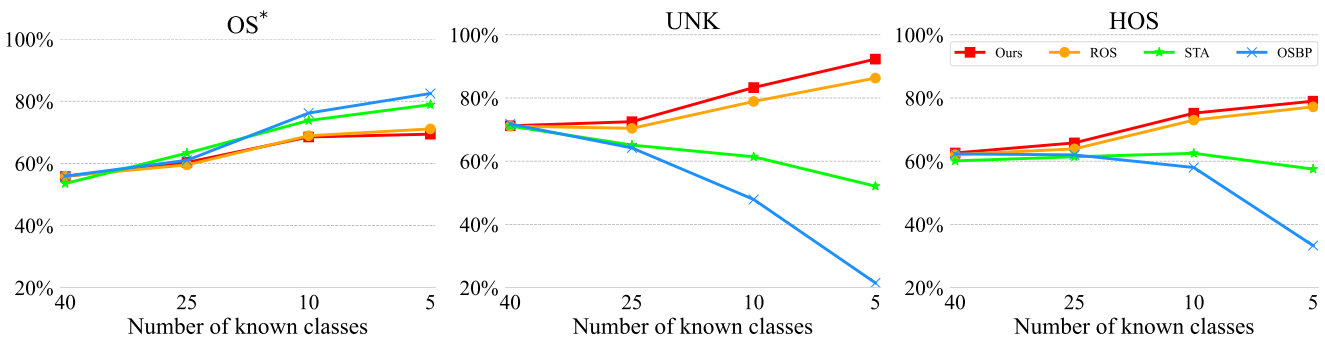
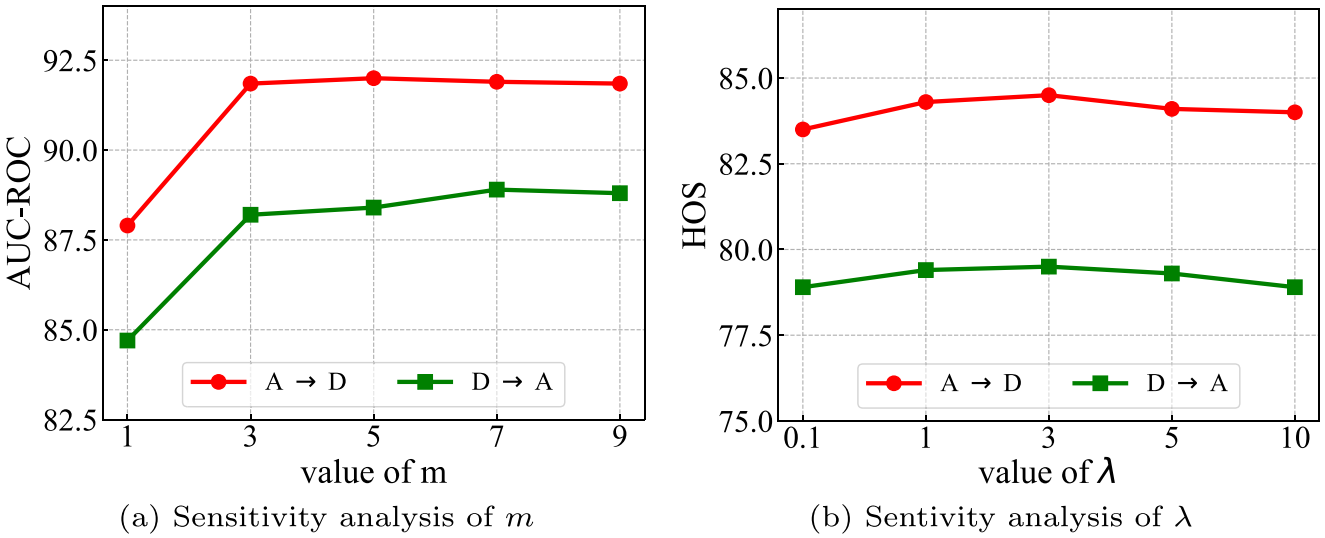


Fig. 7 Accuracy (%) over the four different openness levels



(a) Sensitivity analysis of m (b) Sentivity analysis of λ

Fig. 8 Hyper-parameters sensitivity analysis

$D \rightarrow W$. From the Fig. 6, we can observe that $d_{\mathcal{A}}$ for *source-known* using our method is much smaller than the ResNet-50 (source-only), while that for *known-unknown* is larger than ResNet-50. The above observations demonstrate that our method can align the source and target known data while filtering out target unknown samples.

Sensitivity to Varying Openness. The openness is defined as $\mathbb{O} = 1 - \frac{|\mathcal{C}_s|}{|\mathcal{C}_t|}$, and the value of openness in the standard OSDA setting is around 0.5 which means the number of the known and unknown target classes are close. For example, the openness of Office-31 is $\mathbb{O} = 1 - \frac{10}{21} = 0.52$ and that of Office-Home is $\mathbb{O} = 1 - \frac{25}{65} = 0.62$. In practical applications, the number of unknown target classes may exceed the number of known classes by a large margin, with openness approaching 1. To testify the robustness of our method, we conduct experiments on Office-Home with the following different openness levels: $\mathbb{O} = 0.38$ (40 known classes), $\mathbb{O} = 0.62$ (25 known classes), $\mathbb{O} = 0.85$ (10 known classes), $\mathbb{O} = 0.92$ (5 known classes). As shown in Fig. 7, the performance of OSBP and STA drops a lot with larger $\mathbb{O}$, as they are unable to reject the unknown instances well. In contrast, our method and ROS are resistant to the change in openness. In addition, our method outperforms ROS consistently, owing to its advanced ability of separating between known and unknown samples.

Sensitivity to Hyper-parameters. We investigate the sensitivity of two hyper-parameters: the number of transformations m (in (4) and (5)) and the trade-off weight λ (in (12)). The experiments are performed on two tasks $A \rightarrow D$ and $D \rightarrow A$ with ResNet-50 as the backbone. We plot the relationship of the $AUC - ROC$ and the value of m in Fig. 8a, and the relationship of the HOS and the value of λ in Fig. 8b. Specifically, $m = 0$ denotes the ablation where EMT is not used. We can observe that our method is not sensitive to both hyper-parameters. We underline that the same hyper-parameters are used for all 18 domain pairs demonstrating that the choice of the hyperparameters' value is robust across datasets.

5 Conclusions

In this paper, we propose a novel OSDA method that can conduct effective known and unknown separation. Specifically, we propose an explicit module of deconfounding domain gaps (DDP) that enables the model to recognize a class as known (or unknown) because of the class semantics rather than the confusion caused by spurious styles or contexts. In addition, to obtain the reliable normality scores, we also propose a module of ensembling multiple transformations (EMT) to calibrate the model output. The accurate known/unknown separation results boost the overall performance of the OSDA model. Experimental results on two standard datasets show that the proposed method outperforms the state-of-the-art OSDA methods, especially with a large margin on recognizing unknown samples.

Acknowledgements This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME YIRG Grant (Project No. A20E6c0101), Graduate Innovation Fund of Jilin University (101832020CX179), the Innovation Capacity Construction Project of Jilin Province Development and Reform Commission(2021FGWCXNLJSSZ10), the National Key Research and Development Program of China (No. 2020YFA0714103), the Science & Technology Development Project of Jilin Province, China (20190302117GX) and the Fundamental Research Funds for the Central Universities, JLU.

Data availability The datasets used in this study are publicly available online.

References

1. 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
2. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848
3. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
4. Sun B, Feng J, Saenko K (2016) Return of frustratingly easy domain adaptation. In: Proceedings of the AAAI conference on artificial intelligence. vol 30, pp 2058–2065
5. Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
6. Panareda Busto P, Gall J (2017) Open set domain adaptation. In: Proceedings of the IEEE international conference on computer vision. pp 754–763
7. Saito K, Yamamoto S, Ushiku Y, Harada T (2018) Open set domain adaptation by backpropagation. In: Proceedings of the european conference on computer vision. pp 153–168
8. Liu H, Cao Z, Long M, Wang J, Yang Q (2019) Separate to adapt: Open set domain adaptation via progressive separation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 2927–2936
9. Wang T, Zhou C, Sun Q, Zhang H (2021) Causal attention for unbiased visual recognition. In: Proceedings of the IEEE international conference on computer vision. pp 3091–3100
10. French G, Mackiewicz M, Fisher M (2018) Self-ensembling for visual domain adaptation. In: Proceedings of the international conference on learning representations
11. Long M, Cao Y, Wang J, Jordan M (2015) Learning transferable features with deep adaptation networks. In: Proceedings of the international conference on machine learning. pp 97–105
12. Chen C, Fu Z, Chen Z, Jin S, Cheng Z, Jin X, Hua X-S (2020) Homm: higher-order moment matching for unsupervised domain adaptation. In: Proceedings of the AAAI conference on artificial intelligence. vol 34, pp 3422–3429
13. Courty N, Flamary R, Tuia D, Rakotomamonjy A (2017) Optimal transport for domain adaptation. IEEE Trans Pattern Anal Mach Intell 39(9):1853–1865
14. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V (2016) Domain-adversarial training of neural networks. J Mach Learn Res 17(1):2096–2030
15. Long M, Cao Z, Wang J, Jordan MI (2018) Conditional adversarial domain adaptation. In: Advances in neural information processing systems. vol 31, pp 1647–1657
16. 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 conference on computer vision and pattern recognition. pp 627–636
17. Cui S, Wang S, Zhuo J, Su C, Huang Q, Tian Q (2020) Gradually vanishing bridge for adversarial domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 12455–12464
18. Yang G, Ding M, Zhang Y (2022) Bi-directional class-wise adversaries for unsupervised domain adaptation, vol 52, pp 3623–3639
19. Bousmalis K, Silberman N, Dohan D, Erhan D, Krishnan D (2017) Unsupervised pixel-level domain adaptation with generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 3722–3731
20. Yang G, Xia H, Ding M, Ding Z (2020) Bi-directional generation for unsupervised domain adaptation. In: Proceedings of the AAAI conference on artificial intelligence. vol 34, pp 6615–6622
21. Carlucci FM, D'Innocente A, Bucci S, Caputo B, Tommasi T (2019) Domain generalization by solving jigsaw puzzles. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 2229–2238
22. Ghifary M, Kleijn WB, Zhang M, Balduzzi D, Li W (2016) Deep reconstruction-classification networks for unsupervised domain adaptation. In: Proceedings of the european conference on computer vision. pp 597–613
23. Zhao X, Wang S (2019) Adversarial learning and interpolation consistency for unsupervised domain adaptation. IEEE Access 7:170448–170456
24. Cai R, Li Z, Wei P, Qiao J, Zhang K, Hao Z (2019) Learning disentangled semantic representation for domain adaptation. In: Proceedings of the international joint conference on artificial intelligence. pp 2060–2066
25. Peng X, Huang Z, Sun X, Saenko K (2019) Domain agnostic learning with disentangled representations. In: Proceedings of the international conference on machine learning. pp 5102–5112
26. Bousmalis K, Trigeorgis G, Silberman N, Krishnan D, Erhan D (2016) Domain separation networks. In: Advances in neural information processing systems. pp 343–351
27. Feng Q, Kang G, Fan H, Yang Y (2019) Attract or distract: exploit the margin of open set. In: Proceedings of the IEEE International Conference on Computer Vision. pp 7990–7999
28. Bucci S, Loghmani MR, Tommasi T (2020) On the effectiveness of image rotation for open set domain adaptation. In: Proceedings of the european conference on computer vision. pp 422–438
29. You K, Long M, Cao Z, Wang J, Jordan MI (2019) Universal domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 2720–2729
30. Fu B, Cao Z, Long M, Wang J (2020) Learning to detect open classes for universal domain adaptation. In: Proceedings of the european conference on computer vision. pp 567–583
31. Zhai S, Cheng Y, Lu W, Zhang Z (2016) Deep structured energy based models for anomaly detection. In: Proceedings of the international conference on machine learning. pp 1100–1109
32. Zong B, Song Q, Min MR, Cheng W, Lumezanu C, Cho D, Chen H (2018) Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: Proceedings of the international conference on learning representations
33. Xia Y, Cao X, Wen F, Hua G, Sun J (2015) Learning discriminative reconstructions for unsupervised outlier removal. In: Proceedings of the IEEE international conference on computer vision. pp 1511–1519
34. Zhou C, Paffenroth RC (2017) Anomaly detection with robust deep autoencoders. In: Proceedings of the ACM international conference on knowledge discovery and data mining. pp 665–674
35. Liang S, Li Y, Srikant R (2018) Enhancing the reliability of out-of-distribution image detection in neural networks. In: Proceedings of the international conference on learning representations
36. Yu Q, Kavitha MS, Kurita T (2021) Mixture of experts with convolutional and variational autoencoders for anomaly detection. Appl Intell 51(6):3241–3254
37. Zhou K, Yang Y, Qiao Y, Xiang T (2021) Domain generalization with mixstyle. In: Proceedings of the international conference on learning representations
38. Nam H, Lee H, Park J, Yoon W, Yoo D (2021) Reducing domain gap by reducing style bias. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 8690–8699
39. Huang X, Belongie S (2017) Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE international conference on computer vision. pp 1501–1510

40. Karras T, Laine S, Aila T (2019) A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 4401–4410
41. Müller R, Kornblith S, Hinton GE (2019) When does label smoothing help? In: Advances in neural information processing systems. vol 32, pp 4694–4703
42. Saenko K, Kulis B, Fritz M, Darrell T (2010) Adapting visual category models to new domains. In: Proceedings of the european conference on computer vision. pp 213–226
43. 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
44. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 248–255
45. Jiang J, BoFu ML (2020) Transfer-Learning-library. GitHub
46. Maaten Lvd, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res 9(Nov):2579–2605
47. Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW (2010) A theory of learning from different domains, vol 79, pp 151–175

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.

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.

Qianru Sun received the PhD degree from Peking University. She is currently a Tenure-Track assistant professor with the School of Information Systems, Singapore Management University, since 2019. From 2018 to 2019, she was a joint research fellow with the National University of Singapore and the MPI for Informatics. From 2016 to 2018, she held the Lise Meitner Award Fellowship and worked at the MPI for Informatics. In 2014, she was a visiting student with the University of Tokyo. Her research interests include computer vision and machine learning that aim to develop efficient algorithms and systems for visual understanding.