# DISCOVERING IMPLICIT CLASSES ACHIEVES OPEN SET DOMAIN ADAPTATION

*Jingyu Zhuang, Ziliang Chen, Pengxu Wei, Guanbin Li and Liang Lin**

Sun Yat-sen University
zhuangjy6@mail2.sysu.edu.cn; c.ziliang@yahoo.com;
{weipx3, liguanbin}@mail.sysu.edu.cn; linliang@ieee.org

2022 IEEE International Conference on Multimedia and Expo (ICME) | 978-1-6654-8563-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICME52920.2022.9859594

## ABSTRACT

In *Open Set Domain Adaptation* (**OSDA**), large amounts of target samples are drawn from the implicit categories that never appear in the source domain. Due to the lack of their specific belonging, existing methods indiscriminately regard them as a single class "*unknown*". We challenge this broadly-adopted practice that may arouse unexpected detrimental effects because the decision boundaries between the implicit categories have been fully ignored. Instead, we propose Self-supervised Class-Discovering Adapter (**SCDA**) that attempts to achieve OSDA by gradually discovering those implicit classes, then incorporating them to restructure the classifier and update the domain-adaptive features iteratively. SCDA performs two alternate steps to achieve implicit class discovery and self-supervised OSDA, respectively. By jointly optimizing for two tasks, SCDA achieves the state-of-the-art in OSDA and shows a competitive performance to unearth the implicit target classes.

***Index Terms***— Domain Adaptation, Open Set Recognition, Class Discovery, Unsupervised Learning

## 1. INTRODUCTION

Unsupervised Domain Adaptation (**UDA**) methods aim to transfer the knowledge from a labeled source domain to classify unlabeled samples in a target domain by minimizing the cross-domain distribution discrepancy. Despite their impressive progress, UDA methods usually operate under the *close-set* assumption, *i.e.*, the source categories and the target categories should be exactly identical. However, this assumption is probably violated in the real world since target samples are collected from diverse classes even beyond source categories. Therefore, *Open Set Domain Adaptation* (**OSDA**) [1, 2] has attracted increasing attention, where the target domain contain ***implicit classes*** that never appear in the source domain.

Due to the absence of both the corresponding categories for target samples and the number for implicit classes, existing OSDA methods regard all target samples of implicit classes as a single class "*unknown*". This practice is straightforward but probably problematic. Specifically, existing OSDA methods [3–5] aim to minimize the cross-entropy losses of known classes and the "*unknown*" class. Under the cluster assumption [6], the features of unknown target samples are optimized to converge to an identical center due to the shared labels. Whereas, since their intrinsic structure and diversity have been ignored, the convergence is hard to achieve in practice. Especially when unknown target samples contain more categories, the features of unknown samples will be more probably mixed up with the known-class features around the decision boundaries between known classes and the "*unknown*" class (Figure.1.b), hence, breeding the potential performance drop.

In this paper, we focus on a new methodology to achieve OSDA from another point of view. Instead of fabricating the "*unknown*", we aim to transfer the source knowledge along with discovering implicit classes [7] in unknown target samples. The process automatically estimates the number of implicit classes and how they are distributed, then leverages

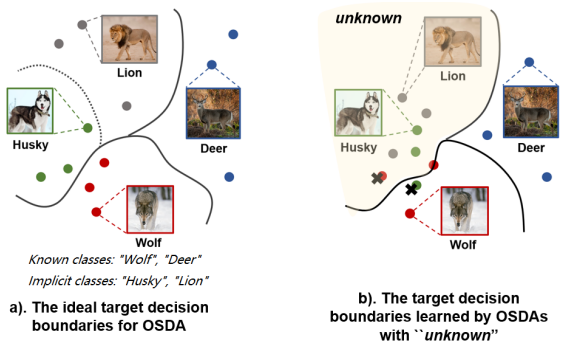


**Fig. 1**: Existing OSDA methods unitedly treat implicit classes as a single negative class "*unknown*". Our work challenges this broadly-accepted practice, since ignoring decision boundaries between implicit classes results in all implicit-class features converging together due to the cluster assumption, which further causes a detrimental effect. For instance, suppose *husky*, *lion* denote the implicit classes and *wolf*, *deer* denote the known classes. Given a known class similar with an implicit class, *e.g.*, *wolf* v.s. *husky*, the *wolf* target features could be attracted to the feature field of *lion* since *husky* and *wolf* features are hard to distinguish, yet *husky* and *lion* tend to converge to the identical center in the "*unknown*".

* Liang Lin is the corresponding author. This work is supported by NSFC (No.62006253, No.61976250, No.U1811463) and Guangdong Basic and Applied Basic Research Foundation under Grant No.2020B1515020048.

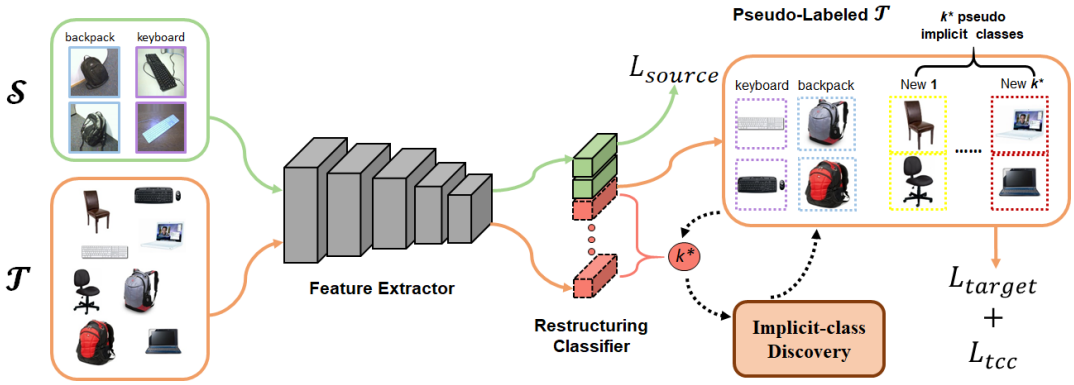


**Fig. 2**: An overview of SCDA. SCDA starts from a pre-training to obtain unknown target samples. Then SCDA designs an algorithm to discover the implicit classes in the unknown target samples; Afterwards, SCDA restructures the classifier and updates it to recognize them.

this self-supervised information to progressively update the OSDA model to improve the performance of OSDA.

To this end, we propose Self-supervised Class-Discovering Adapter (SCDA). After an adversarial pre-training, SCDA separates unknown target samples roughly. Unlike previous OSDA methods that regard all unknown target samples as the "*unknown*" and neglect their inter-class structure, SCDA designs an unsupervised algorithm, *i.e.*, implicit class discovery, to estimate the number of the implicit classes and group the unknown samples belonging to each implicit class. Based on the results of implicit class discovery, SCDA restructures and updates the model to push features of different classes away from each other to reduce the risk of confusing them. Our contributions are summarized in two aspects:

- We consider OSDA from a new point of view: Instead of regarding target samples beyond categories in the source domains as the "*unknown*" class, we attempt to discover their structure along with the domain adaption.
- We propose Self-supervised Class-Discovering Adapter (SCDA) which combines discovering the implicit classes and learning domain-invariant features in a framework.
- Extensive experiments on three OSDA benchmarks are conducted to evaluate SCDA. The results evidence the superiority of SCDA on both OSDA and implicit class discovery.

## 2. RELATED WORK

**Close-set domain adaptation (UDA).** Assuming labeled source and the unlabeled target domains share their label sets, closed-set UDA methods aim at reducing the domain shift across the source and target domains. Most existing algorithms are based on either domain discrepancy matching or adversarial learning. Under this assumption, [8] proposes multiple source domain adaptation and [9] proposes blending-target domain adaptation. UDA is also applied in other visual task, like scene parsing [10] and segmentation [11].

**Open-set domain adaptation (OSDA).** OSDA learners categorize target samples into the known classes or *unknown*. OSBP [2] learns representations to separate unknown target samples through an adversarial method. STA [12] trains a binary classifier to perform fine separation on all target samples and weight each target sample to alleviate negative-transfer caused by unknown target samples. Besides, there are a few works, *e.g.*, TIM [4], SHOT [3], JPOT [5] and PGL [13]. However, all these approaches simply assign target samples beyond the label-set of source classes to the *unknown*.

**Novel-class discovery.** Provided the labeled data from related but different classes, novel-class discovery aims to find novel classes in unlabeled data. However, existing methods, *e.g.*, DTC [14] and [7], assume that the labeled and unlabeled data are drawn from identical distribution and non-overlapping classes. They can not be directly adopted for OSDA, since the domain shift between the labeled and unobserved data would incur a severe performance degeneration.

## 3. METHODOLOGY

### 3.1. Problem Setting

Suppose $n_s$ labeled images $\mathcal{S} = (\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)_{i=1}^{n_s}$ are drawn from a source density $P_{\mathcal{S},\mathcal{C}_\mathcal{S}}(\boldsymbol{x}, \boldsymbol{y})$ and $n_t$ unlabeled images $\mathcal{T} = (\boldsymbol{x}_i^t)_{i=1}^{n_t}$ are drawn from a target density $P_{\mathcal{T},\mathcal{C}_\mathcal{S}}(\boldsymbol{x}) = \int P_{\mathcal{T},\mathcal{C}_\mathcal{S}}(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{y}$. $\mathcal{C}_\mathcal{S}$ denotes the set of the source classes, $\mathcal{C}_\mathcal{T}$ denotes that of the target, and $\mathcal{C}_\mathcal{T}/\mathcal{C}_\mathcal{S}$ denotes the implicit classes in $\mathcal{T}$. In OSDA, due to $\mathcal{C}_\mathcal{T}/\mathcal{C}_\mathcal{S} \neq \emptyset$, we are required to classify target samples of $|\mathcal{C}_\mathcal{S}|$ known classes correctly ($|A|$ indicates the number of members in $A$) and simultaneously reject the unknown target samples belonging to $\mathcal{C}_\mathcal{T}/\mathcal{C}_\mathcal{S}$.

### 3.2. Overall Architecture

SCDA can be flexibly deployed to existing neural network architectures. As shown in Figure.2, the architecture of SCDA consists of two modules: a *feature extractor* $F$ proposed to learn class-aware domain-invariant features; a *dynamically restructuring classifier* $C$ proposed to classify unlabeled target samples into the known classes $\mathcal{C}_\mathcal{S}$ and the implicit classes $\mathcal{C}_\mathcal{T}$. The output dimension of $C$ is initialized as $|\mathcal{C}_\mathcal{S}| + 1$, where the $(|\mathcal{C}_\mathcal{S}| + 1)'$-th class indicates the unknown target samples. It is worth noting that, according to the results of

implicit class discovery, the output dimension of $C$ alters to $|\mathcal{C}_{\mathcal{S}}| + k_t^*$, where $k*$ refers to the newly discovered classes.

### 3.3. Self-supervised Class-Discovering Adapter

The pipeline of SCDA mainly consists of two alternate steps, *i.e.*, implicit class discovery (Sec.3.3.2) and self-supervised OSDA (Sec.3.3.3). Briefly, SCDA employs a pre-training to roughly separate the unknown target samples. Then, it alternately performs two steps to discover the implicit classes in $\mathcal{C}_{\mathcal{T}}/\mathcal{C}_{\mathcal{S}}$ and further improve the performance of OSDA based on the results of discovery. First, SCDA discovers the implicit classes in $\mathcal{C}_{\mathcal{T}}/\mathcal{C}_{\mathcal{S}}$ by estimating their number and constructing the pseudo implicit classes through a clustering assignment. Then based on the pseudo implicit classes, SCDA restructures the $C$ to recognize newly discovered classes; SCDA trains $C$ along with $F$ to diminish the domain gap so that the model can be generalized to classify the target samples in implicit classes. SCDA repeatedly executes two alternate steps until the maximal epoch is reached. We elaborate the pre-training and the two steps in the following three subsections. The pipeline of SCDA is found in Supplementary Algorithm.2.

#### *3.3.1.* ***Pre-training***

Due to the absence of target samples' labels, SCDA utilizes adversarial learning in this pre-training step to preliminarily separate the unknown target samples. In brief, $C$ is trained to confuse the known and unknown target samples while $F$ is trained oppositely to distinguish them. We utilize the correlation confusion derived from [15] to implement the adversarial training. Specifically, given a mini-batch of $m$ target samples, each element in a class correlation matrix $\mathbf{R}$ presents as:

$$\mathbf{R}_{i,j}= \hat{\mathbf{y}}_{i,\cdot}^{\top} \frac{m\Big(1+\exp\big(-H(\boldsymbol{x}_i^{(t)};F,C)\big)\Big)}{\sum_{i'=1}^{m}\Big(1+\exp\big(-H(\boldsymbol{x}_{i'}^{(t)};F,C)\big)\Big)} \hat{\mathbf{y}}_{j,\cdot} \quad (1)$$

where $\hat{\mathbf{y}}_{j,\cdot}$ represents the softmax output for the class-$j$ prediction over $m$ target examples in the mini-batch and $H$ (Eq.7) is a measure to increase the weight of the reliable examples.

The $j'$-th column in $\mathbf{R}$ measures the correlation between the $j'$-th class and other classes when $C$ classifies a mini-batch of samples. The higher $\mathbf{R}_{j,j'}$ implies that $C$ will more probably classify the samples drawn from the $j$-th class to the $j'$-th class. So we can adjust the value of $\mathbf{R}$ between known and unknown classes to cause confusion. To be specific, after normalizing $\mathbf{R}$ by the sum of each row, we obtain $\hat{\mathbf{R}}$ where the summation over each row is 1. Then we optimize the value of $\hat{\mathbf{R}}_{j,|\mathcal{C}_{\mathcal{S}}|+1}$ to 0.5 which means the probability that $C$ classifies samples into the unknown class or $j'$-th known class is equal, *i.e.*, $C$ can not distinguish the known and unknown samples. While, we train $F$ in the opposite direction by inserting a reversed gradient layer [16] between $C$ and $F$. The adversarial learning loss is defined as:

$$L_{adv} = \mathbb{E}_{(\boldsymbol{x})\sim\mathcal{T}} L_{bce}\Big(\frac{1}{|\mathcal{C}_{\mathcal{S}}|}\sum_{j=1}^{|\mathcal{C}_{\mathcal{S}}|}\hat{\mathbf{R}}_{j,|\mathcal{C}_{\mathcal{S}}|+1}, \frac{1}{2}\Big) \quad (2)$$

**Algorithm 1** Implicit Class Discovery in Sec.3.3.2

**Input:** Target dataset $\mathcal{T}$; pre-trained feature extractor $F$ and classifier $C$; max implicit classes number $k_{max}$.

**Output:** The estimation number $k^*$ of implicit classes; pseudo-labeled known target data $\hat{\mathcal{T}}_{\text{kn}}$; pseudo-labeled newly discovered target data $\{\hat{\mathcal{T}}_i\}_{i=1}^{k^*}$.

1: Compute the entropy for $\boldsymbol{x}^t \sim \mathcal{T}$ by Eq. 7. Sort $\boldsymbol{x}^t$ based on their entropies. Select samples according to the entropy to build $\hat{\mathcal{T}}_{\text{kn}}$ with pseudo labels $\hat{\boldsymbol{y}}_t$ and $\hat{\mathcal{T}}_{\text{im}}$.
2: Extract feature of $\hat{\mathcal{T}}_{\text{kn}}$ and $\hat{\mathcal{T}}_{\text{im}}$ using $F$.
3: **For** $0 \le k \le k_{max}$ **do**
4: Run k-means++ on the extracted feature with $k$ clusters.
5: Compute CA for $\hat{\mathcal{T}}_{\text{kn}}$ and SSE for $\hat{\mathcal{T}}_{\text{kn}} \cup \hat{\mathcal{T}}_{\text{im}}$.
6: **End for**
7: Let $\hat{k}$=$(k^*_{\text{CA}}+k^*_{\text{elbow}})/2$. $k^*_{\text{CA}}$ is the value of $k$ maximizes CA. $k^*_{\text{elbow}}$ is generated by the elbow method.
8: Let $k^*$=$\hat{k}-|\mathcal{C}_{\mathcal{S}}|$. Run k-means++ on the features of $\hat{\mathcal{T}}_{\text{im}}$ to obtain $k^*$ clusters $\{\hat{\mathcal{T}}_i\}_{i=1}^{k^*}$. Categorize $\hat{\mathcal{T}}_{\text{im}}$ into $\{\hat{\mathcal{T}}_i\}_{i=1}^{k^*}$ with pseudo labels.
9: **Return** $k^*$=$|\mathcal{C}_{\mathcal{O}}|$; $\hat{\mathcal{T}}_{\text{kn}}$; $\{\hat{\mathcal{T}}_i\}_{i=1}^{k^*}$

Simultaneously, provided with source labeled data $\mathcal{S}$, we have a standard cross-entropy loss $L_{\text{s}}$ to correctly categorize the known classes in $\mathcal{C}_{\mathcal{S}}$:

$$L_{\text{s}} = -\,\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{S}}\, \boldsymbol{y}^T \log\, C\big(F(\boldsymbol{x})\big) \quad (3)$$

Besides, to alleviate the cross-known-class confusion caused by domain shift, we optimize $L_{\text{kcc}}$ (Eq.4). It is worth noting that, $L_{\text{kcc}}$ does not punish the confusion to $\mathcal{C}_{\mathcal{T}}/\mathcal{C}_{\mathcal{S}}$, leading to the cross-domain features only aligned on $\mathcal{C}_{\mathcal{S}}$.

$$L_{kcc} = \mathbb{E}_{(\boldsymbol{x})\sim\mathcal{T}}\, \frac{1}{|\mathcal{C}_{\mathcal{S}}|}\sum_{j=1}^{|\mathcal{C}_{\mathcal{S}}|}\sum_{j'\neq j}^{|\mathcal{C}_{\mathcal{S}}|}\hat{\mathbf{R}}_{j,j'} \quad (4)$$

Combing the above items, the pre-train objective is:

$$\min_{F} L_{\text{s}} - L_{\text{adv}} + L_{\text{kcc}} \quad (5)$$

$$\min_{C} L_{\text{s}} + L_{\text{adv}} + L_{\text{kcc}} \quad (6)$$

After preparation, SCDA alternately runs two steps to achieve implicit class discovery and self-supervised OSDA.

#### *3.3.2.* ***Implicit class discovery***

In this step, SCDA attempts to determine the number of implicit classes in $\mathcal{T}$ with the help of the labeled data. However, if we directly use the labeled source data, the domain shift between source and target domain would affect the accuracy of the estimation. Hence, SCDA first constructs two ***high-confident target candidates*** sets $\hat{\mathcal{T}}_{\text{kn}}$ and $\hat{\mathcal{T}}_{\text{im}}$ with pseudo labels, indicating target samples in known classes and implicit classes, respectively. Then SCDA estimates $|\mathcal{C}_{\mathcal{T}}/\mathcal{C}_{\mathcal{S}}|$ by evaluating the ***clustering consistency*** between $\hat{\mathcal{T}}_{\text{kn}}$ and $\hat{\mathcal{T}}_{\text{im}}$, and assigns pseudo labels to the newly discovered classes. This step has been summarized in Algorithm.1.

**High-confident target candidates.** Instead of analyzing whole $\mathcal{T}$, we select target candidates with higher cross-domain classification consistency, because the target samples with higher consistency are more reliable. The cross-domain classification consistency can be measured by Eq.7. The lower entropy to classify target samples with a source classifier implies the higher consistency.

$$H(\boldsymbol{x}^t; F, C) = -\sum_{i=1}^{|\mathcal{C}_\mathcal{S}|+k^*} C_i(F(\boldsymbol{x}^{(t)})) \log C_i(F(\boldsymbol{x}^{(t)})) \quad (7)$$

where $C_i(F(\boldsymbol{x}^{(t)}))$ denotes the softmax value of $\boldsymbol{x}^t$ with respect to the $i'$-th class; $k^*$=$\mathcal{C}_\mathcal{D}$ denotes the optimal estimation of $|\mathcal{C}_\mathcal{T}/\mathcal{C}_\mathcal{S}|$ in the previous epoch ($k^*$=1 in the initialization). $\mathcal{C}_\mathcal{D}$ indicates the newly discovered target classes and the goal of SCDA is to iteratively update $\mathcal{C}_\mathcal{D}$ to approximate $\mathcal{C}_\mathcal{T}/\mathcal{C}_\mathcal{S}$.

Specifically, for each class in $\mathcal{C}_\mathcal{S}$, SCDA picks out the samples in with $\mathcal{T}$ the corresponding pseudo label and selects the first half of them with low entropies to construct the target candidate subset of known classes $\hat{\mathcal{T}}_{\rm kn}$. Similarly, as for those in $\mathcal{C}_\mathcal{D}$, SCDA also selects the half of them to construct the target candidate subset of implicit classes $\hat{\mathcal{T}}_{\rm im}$. Obviously, the domain gap between $\hat{\mathcal{T}}_{\rm kn}$ and $\hat{\mathcal{T}}_{\rm im}$ has been erased, and thus, SCDA executes a dynamical cluster algorithm which splits the features of $\hat{\mathcal{T}}_{\rm kn} \cup \hat{\mathcal{T}}_{\rm im}$ by varying the clustering number $k$ then compares their clustering consistency value to obtain an optimal $k$ as the estimation of $|\mathcal{C}_\mathcal{T}/\mathcal{C}_\mathcal{S}|$.

**Criteria for the clustering consistency.** We employ two criteria to evaluate the clustering consistency value. The first criterion refers to the elbow method widely adopted in clustering analysis. It plots the sum of squared error (SSE) as a function of $k$ to search the elbow point. Specifically, we take the kneedle algorithm [17] to locate the point with the ideal cluster number $k^*_{\rm elbow}$. The elbow method balances the diversity and the granularity of the clusters, but it can not reflect the prior knowledge of $\mathcal{C}_\mathcal{S}$.

Hence, as a supplement, we compute the clustering accuracy (Eq.8) on $\hat{\mathcal{T}}_{\rm kn}$ to measure the clustering quality. CA measures the clustering accuracy between the clustering assignment and the pseudo label over $\hat{\mathcal{T}}_{\rm kn}$. Higher CA indicates the clustering results are more consistent with $\mathcal{C}_\mathcal{S}$ in the target domain. We select the $k^*_{\rm CA}$ with the highest CA.

$$k_{\rm CA} = \underset{k\in\{1+|\mathcal{C}_\mathcal{S}|,\cdots,k_{\max}+|\mathcal{C}_\mathcal{S}|\}}{\arg\max} \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\hat{\boldsymbol{y}}_i = M(\boldsymbol{c}_i)} \quad (8)$$

where $\mathbf{1}$ denotes the indicator function and $M(\boldsymbol{c}_i)$ is permutation mapping that maps each cluster label $\boldsymbol{c}_i$ to the pseudo label $\hat{\boldsymbol{y}}_i$ over total $n$ $\boldsymbol{x}_i \in \hat{\mathcal{T}}_{\rm kn}$.

SCDA takes $\hat{k} = (k^*_{\rm CA} + k^*_{\rm elbow})/2$ as the optimal clustering number. Excluding $|\mathcal{C}_\mathcal{S}|$ source classes, we consider $k^* = \hat{k} - |\mathcal{C}_\mathcal{S}|$ as the optimally estimated number of $\mathcal{C}_\mathcal{T}$/$\mathcal{C}_\mathcal{S}$, and use the corresponding clustering assignment to divide $\hat{\mathcal{T}}_{\rm im}$ into $k^*$ pseudo classes $\{\hat{\mathcal{T}}_i\}_{i=1}^{k^*}$. It refers to the update of $\mathcal{C}_\mathcal{D}$.

### *3.3.3. Self-supervised open-set adaptation*

Step 1 has provided an estimation result for the open-set class discovery, while it is not able to improve OSDA since the result has not been fed back to the domain-invariant feature learning yet. Provided with this, we develop the self-supervised OSDA which enables the classifier $C$ to recognize more target samples that belong to the classes in $\mathcal{C}_\mathcal{T}/\mathcal{C}_\mathcal{S}$, further improving the OSDA performance.

**Restructuring $C$.** Since $\mathcal{C}_\mathcal{D}$ dynamically changes to approximate $\mathcal{C}_\mathcal{T}/\mathcal{C}_\mathcal{S}$, the softmax classifier $C$ is also dynamically restructured in order to classify $\mathcal{C}_\mathcal{D}$: Its output dimension alters from $|\mathcal{C}_\mathcal{S}| + k^*_{t-1}$ to $|\mathcal{C}_\mathcal{S}| + k^*_t$, in which the first $|\mathcal{C}_\mathcal{S}|$ corresponds the number of classes in $\mathcal{C}_\mathcal{S}$ and the latter refers to $k*$ classes in $\mathcal{C}_\mathcal{D}$. The parameters are reset and trained in terms of $\mathcal{C}_\mathcal{S}$ and the current $\hat{\mathcal{C}}_\mathcal{D}$.

**Dynamic class correlation matrix.** The crucial problem is to improve the generalization ability of $F$ and $C$ in terms of the newly discovered classes. To this end, we reconsider the class correlation matrix proposed in Eq.1. Indeed, this class decorrelating technique augments various close-set UDAs to reap the transfer gain. However, it is rarely applied in OSDA since its confusion mechanism naturally repels the "*unknown*", thus, reducing their confusion would eliminate the intrinsic diversity of $\mathcal{C}_\mathcal{T}/\mathcal{C}_\mathcal{S}$. This concern found an echo of our motivation, inspiring us to generalize the class correlation matrix to suit OSDA. Specifically, we reconfigure Eq.1 by using our restructuring softmax classifier output as $\hat{\mathbf{y}}_{i,\cdot}$. Hence, the dimension of $\mathbf{R}$ changes from $(|\mathcal{C}_\mathcal{S}|+1)\times(|\mathcal{C}_\mathcal{S}|+1)$ to $(|\mathcal{C}_\mathcal{S}|+k^*)\times(|\mathcal{C}_\mathcal{S}|+k^*)$, extending the confusion measurement from the known classes in $\mathcal{C}_\mathcal{S}$ and the "*unknown*" class, to the newly discovered classes in $\mathcal{C}_\mathcal{D}$ and the correlation between $\mathcal{C}_\mathcal{S}$ and $\mathcal{C}_\mathcal{D}$. OSDA with the dynamic class correlation matrix aims to minimize:

$$\min_{C,F} L_{\rm tcc} = \mathbb{E}_{\{\boldsymbol{x}_i^{(t)}\}_{i=1}^m \sim \mathcal{T}} \frac{1}{|\mathcal{C}_\mathcal{S}|+k^*} \sum_{j=1}^{|\mathcal{C}_\mathcal{S}|+k^*} \sum_{j'\neq j}^{|\mathcal{C}_\mathcal{S}|+k^*} \hat{\mathbf{R}}_{j,j'} \quad (9)$$

Compared with Eq.4, Eq.9 iteratively changes its dimension to measure the confusion of $\mathcal{C}_\mathcal{S}\cup\hat{\mathcal{C}}_\mathcal{D}$. It disambiguates the pseudo class assignment produced by Algorithm.1 and helps $F$ learn more discriminative features.

To preserve the knowledge from the known classes, we also keep training C with the source samples (Eq.3). In order to further approach an ideal performance, we simultaneously incorporate the pseudo-labeled target samples drawn from $\hat{\mathcal{T}}_{\rm kn} \cup \hat{\mathcal{T}}_{\rm im}$ to learn transferable features:

$$L_t = -\mathbb{E}_{(\boldsymbol{x},\hat{\boldsymbol{y}})\sim \hat{\mathcal{T}}_{\rm kn}\cup\hat{\mathcal{T}}_{\rm im}} \hat{\boldsymbol{y}}^T \log\, C(F(\boldsymbol{x})) \quad (10)$$

where $\hat{\boldsymbol{y}}$ denotes the corresponding pseudo label.

$$\min_{F,C}\ L_{\rm s} + L_{\rm t} + L_{\rm tcc} \quad (11)$$

In summary, the final objective of this step is formulated as Eq.11. Note that we do not use any hyperparameter to balance each term in all objectives (Eq.6, Eq.5 and Eq.11).

**Table 1**: Results on Office-31 for OSDA. ° indicates our re-implementation with the officially released code.

| Method | A→W | | A→D | | D→W | | W→D | | D→A | | W→A | | Avg | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OS | OS* | OS | OS* | OS | OS* | OS | OS* | OS | OS* | OS | OS* | OS | OS* |
| OSBP | 86.5±2.0 | 87.6±2.1 | 88.6±1.4 | 89.2±1.3 | 97.0±1.0 | 96.5±0.4 | 97.9±0.9 | 98.7±0.6 | 88.9±2.5 | 90.6±2.3 | 85.8±2.5 | 84.9±1.3 | 90.8 | 91.3 |
| STA | 89.5±0.6 | 92.1±0.5 | 93.7±1.5 | 96.1±0.4 | 97.5±0.2 | 96.5±0.5 | 99.5±0.2 | 99.6±0.1 | 89.1±0.5 | 93.5±0.8 | 87.9±0.9 | 87.4±0.6 | 92.9 | 94.1 |
| TIM | 91.3±0.7 | 93.2±1.2 | 94.2±1.1 | **97.1±0.8** | 96.5±0.5 | 97.4±0.7 | 99.5±0.2 | 99.4±0.3 | 90.1±0.2 | 91.5±0.2 | 88.7±1.3 | 88.1±0.9 | 93.4 | 94.5 |
| JPOT | 92.8±0.6 | 92.2±0.4 | 95.2±0.9 | 96.0±0.6 | 98.1±0.3 | 96.2±0.4 | 99.5±0.1 | 98.6±0.2 | **93.0±0.7** | **94.1±0.4** | 88.9±1.0 | 88.4±0.4 | 94.6 | 94.3 |
| SHOT° | 88.8±0.7 | 91.4±0.4 | 90.3±0.5 | 92.6±0.3 | 96.2±0.4 | 97.0±0.4 | 97.4 ±0.3 | 97.9±0.5 | 91.6±0.5 | 93.4±0.7 | 91.2±0.5 | 93.5±0.4 | 92.6 | 94.3 |
| PGL° | 89.2±0.5 | 90.1±0.7 | 89.6±0.8 | 91.6±0.5 | 95.3±0.4 | 95.1±0.5 | 96.7±0.5 | 97.6±0.6 | 71.0±0.9 | 72.0±0.5 | 73.0±0.4 | 77.6±0.6 | 85.8 | 87.4 |
| **SCDA** | **95.7±0.1** | **97.5±0.2** | **95.9±0.4** | 96.5±0.4 | **99.2±0.1** | **99.7±0.1** | **99.8±0.1** | **100±0.0** | 92.1±0.1 | 93.7±0.3 | **92.2±0.1** | **93.6±0.1** | **95.8** | **96.8** |

**Table 2**: Results on Office-Home for OSDA. △ indicates the method does not report the variance of their results.

| Method | Ar→Cl | Pr→Cl | Rw→Cl | Ar→Pr | Cl→Pr | Rw→Pr | Cl→Ar | Pr→Ar | Rw→Ar | Ar→Rw | Cl→Rw | Pr→Rw | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OSBP | 56.7±1.9 | 51.5±2.1 | 49.2±2.4 | 67.5±1.5 | 65.5±1.5 | 74.0±1.5 | 62.5±2.0 | 64.8±1.1 | 69.3±1.1 | 80.6±0.9 | 74.7±2.2 | 71.5±1.9 | 65.7 |
| STA | 58.1±0.6 | 53.1±0.9 | 54.4±1.0 | 71.6±1.2 | 69.3±1.0 | 81.9±0.5 | 63.4±0.5 | 65.2±0.8 | 74.9±1.0 | 85.0±0.2 | 75.8±0.4 | 80.8±0.3 | 69.5 |
| TIM | 60.1±0.7 | 54.2±1.0 | 56.2±1.7 | 70.9±1.4 | 70.0±1.7 | 78.6±0.6 | 64.0±0.6 | 66.1±1.3 | 74.9±0.9 | 83.2±0.9 | 75.7±1.3 | 81.3±1.4 | 69.6 |
| JPOT | 59.6±0.5 | 54.2±0.7 | 54.6±0.9 | 72.3±1.1 | 70.1±0.6 | 82.1±0.9 | 62.9±0.7 | 68.3±0.8 | 75.1±1.1 | 84.8±0.4 | 77.4±0.5 | 81.2±0.4 | 70.2 |
| SHOT△ | 64.5±0.0 | **59.3±0.0** | 64.6±0.0 | **80.4±0.0** | **75.4±0.0** | 82.3±0.0 | 63.1±0.0 | 65.3±0.0 | 69.6±0.0 | 84.7±0.0 | 81.2±0.0 | 83.3±0.0 | 72.8 |
| PGL△ | **61.6±0.0** | 58.4±0.0 | **65.0±0.0** | 77.1±0.0 | 72.0±0.0 | 83.0±0.0 | 68.8±0.0 | 72.2±0.0 | **78.6±0.0** | **85.9±0.0** | 82.8±0.0 | 82.6±0.0 | 74.0 |
| **SCDA** | 59.9±0.3 | 59.0±0.3 | 62.8±0.5 | 79.6±0.4 | 73.8±1.0 | **83.7±0.8** | **70.9±0.5** | **72.3±0.6** | 75.5±0.4 | 85.3±0.6 | **82.9±0.3** | **85.7±0.9** | **74.3** |

## 4. EXPERIMENT

In this section, we evaluate SCDA on three benchmarks to demonstrate its superior performance. More implement details can be found in supplementary material. Code is available at https://github.com/zjy526223908/SCDA.

**Benchmarks.** We use two famous datasets: **Office-31** and **Office-Home**, and choose the same label sets of classes to build $\mathcal{C}_{\mathcal{S}}$ and $\mathcal{C}_{\mathcal{T}}$ following [12]. Besides, we introduce the challenging **DomainNet**. To simulate a real-world adaptation scenario, we combine the *Real* and *Clipart* domains in DomainNet with the **Rw** and **Cl** in Office-Home, respectively, to build two target domains **Rw**$^\star$ and **Cl**$^\star$. After merging the same categories, the combined target domains have 362 categories, 279 classes of which are **DomainNet**-specific. Then we randomly select $1/4$ classes from them to induce the scarcity: we select 10 samples for each of the classes and abandon the rest. It breeds a benchmark **DomainNet**$^\star$ with extremely imbalanced target domains **Rw**$^\star$ and **Cl**$^\star$.

**Baselines.** We compare SCDA with a variety of state-of-the-art OSDA approaches, including **OSBP**, **STA**, **TIM**, **JOPT**, **SHOT**, and **PGL**. We are also interested in the performance of discovering implicit classes. To this, we compare SCDA's class-discovering ability with some state-of-the-art baselines, *i.e.*, Silhouette coefficient (**SC**) and **DTC**. We use ResNet-50 pre-trained on ImageNet as the backbone.

**Evaluation Criteria.** For a fair comparison, we employ two evaluation metrics in line with [2], *i.e.*, **OS:** averaging the class-wise target accuracy for all the classes including the unknown as one class; **OS*:** averaging the class-wise target accuracy only on known classes. Besides, in terms of class discovery, we compare SCDA with SC and DTC to estimate the number of unknown implicit classes $k^*$.

### 4.1. Results for OSDA

**Office-31 and Office-Home.** In Table 1, SCDA outperforms other baselines on most transfer tasks in **Office-31** with significant margins. In the hard tasks, *e.g.*, A→W, SCDA outperforms the second with a larger gap (**2.9%**). As illustrated in Table 2, our SCDA still achieves the best performance in **Office-Home** dataset. Notably, the second best model in **Office-31** (TIM) and **Office-Home** (PGL) both perform poorly in the other dataset. It is probably due to the changing setup of *unknown*. In a comparison, SCDA is designed to analyze the inter-class structure of $\mathcal{C}_{\mathcal{T}}$/$\mathcal{C}_{\mathcal{S}}$. Thus, our method presents the more robust generalization ability in OSDA.

**Real-world Scenarios.** To further investigate the baselines in more complicated real world applications, we set up the transfer tasks from **Pr** and **Ar** in **Office-Home** to the challenging blending-target domains **Rw**$^\star$ and **Cl**$^\star$ in **DomainNet**$^\star$. As shown in Table 3, although **Rw**$^\star$ and **Cl**$^\star$ are noisy, and extremely imbalanced, SCDA still achieves the state of the art in 3 from 4 transfer combination. Besides, SCDA also presents a faster convergence rate and a higher upper-bound performance in the complicated scenarios (see our supplementary).

### 4.2. Results for Implicit Class Discovery

In Table 4, we report the results for unknown class number estimation by SC, DTC, and SCDA. SC performs the worst across all the benchmarks. DTC is poor in **Office-Home** and **DomainNet**$^\star$ with numerous implicit classes. By contrast, SCDA shows surprisingly accurate results to estimate the class number in **Office-31**, where the average error is less than 1. Despite the large implicit class number in **Office-Home** and **DomainNet**$^\star$, SCDA produces a low average error. The results validate the reliability of SCDA to estimate the implicit class number. More experiments for class discovery can be found in supplementary material.

**Table 3**: OSDA from **OfficeHome** to **DomainNet**$^\star$

| Method | Ar→Rw$^\star$ | | Ar→Cl$^\star$ | | Pr→Rw$^\star$ | | Pr→Cl$^\star$ | | Avg | |
|---|---|---|---|---|---|---|---|---|---|---|
| | OS | OS* | OS | OS* | OS | OS* | OS | OS* | OS | OS* |
| OSBP | 58.1 | 57.8 | 33.0 | 32.3 | 59.5 | 59.3 | 30.5 | 29.8 | 45.4 | 44.8 |
| STA | 60.5 | 60.4 | 40.1 | 39.6 | 59.1 | 59.0 | 32.3 | 33.9 | 48.0 | 48.2 |
| SHOT | 64.6 | 65.1 | **45.2** | **45.7** | 65.4 | 65.9 | 40.3 | 40.4 | 53.9 | 54.3 |
| Ours | **67.8** | **68.0** | 44.2 | 44.3 | **68.2** | **69.1** | **40.9** | **41.0** | **55.3** | **55.6** |

**Table 4**: Unknown categories number estimation results

| Dataset | GT | SC | | DTC | | Ours | |
|---|---|---|---|---|---|---|---|
| | | $\hat{k}$ | Error | $\hat{k}$ | Error | $\hat{k}$ | Error |
| Office-31 | 11 | 8∼33 | 6.8 | 4∼9 | 4.2 | 9∼11 | **0.9** |
| Office-Home | 40 | 0∼7 | 38.5 | 9∼23 | 21.4 | 32∼37 | **6.1** |
| DomainNet$^\star$ | 297 | 5∼8 | 290.7 | 46∼71 | 238.2 | 227∼255 | **58.5** |

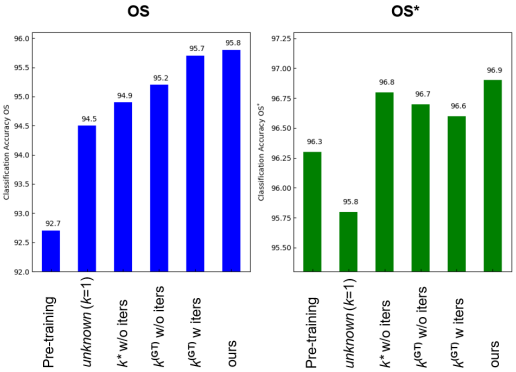


**Fig. 3**: Average results of SCDA

### 4.3. Ablation study

To justify the conjecture that the unobserved class discovery may help OSDA, we compare SCDA with the following modifications. (**1**) **Pre-training**: we use the network only pre-trained by subsection 3.3.1. (**2**) ***unknown*** **($k$=1)**: we utilize the pseudo labels to update the model. However, without class discovery, we regard them as a single negative class *unknown*. (**3**) $k^*$ **w/o iters**: SCDA trains the model with pseudo discovered classes but without further iteration. (**4**) $k^{(GT)}$ **w/o iters**: we provide with the true number of $\mathcal{C}_\mathcal{T}$ but without further iteration. (**5**) $k^{(GT)}$ **w iters**: the algorithm is provided with $k^{(GT)}$, then we alternatively train the model.

As illustrated in Figure 3, we report the average OS and OS* across all six transfer tasks in Office-31. By comparing (**2**) with (**3**), discovering the unobserved classes has a better performance than regarding them as the "*unknown*". It further verifies our motivation: discovering the structure of the unobserved classes can improve the performance of OSDA. The results of (**3**-**5**) draw an interesting conclusion. Without further iterations, training with $k^{(GT)}$ outperforms training with the estimated class number. However, their results are almost the same when we train model iteratively. It suggests that the precise prediction of the unobserved classes number and the iterative optimization are both important and their combination play a key role in addressing OSDA. Besides, more **analysis** and **visualization** are illustrated in SM.

## 5. CONCLUSION

In this paper, we pay attention to a nontrivial challenge in OSDA: discovering all implicit classes in the unknown target samples. The mixed unknown chunk conceives category mismatching risk. To tackle the problem, we propose Self-supervised Class-Discovering Adapter (**SCDA**). SCDA utilizes adversarial learning to preliminarily separate unknown target samples. Then, SCDA employs an alternate approach to discover novel target categories and update our model with the discovery results. Through extensive empirical evaluations, we demonstrate the superiority of our SCDA by the state-of-the-art OSDA performance.

## 6. REFERENCES


[1] Pau Panareda Busto and Juergen Gall, "Open set domain adaptation," in *ICCV*, 2017, pp. 754–763.

[2] Kuniaki Saito, Shohei Yamamoto, et al., "Open set domain adaptation by backpropagation," in *ECCV*, 2018.

[3] Jian Liang et al., "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," in *ICML*, 2020, pp. 6028–6039.

[4] Jogendra Nath Kundu et al., "Towards inheritable models for open-set domain adaptation," in *CVPR*, 2020.

[5] Renjun Xu et al., "Joint partial optimal transport for open set domain adaptation.," in *IJCAI*, 2020.

[6] Rui Shu et al., "A dirt-t approach to unsupervised domain adaptation," *arXiv preprint arXiv:1802.08735*.

[7] Han et al., "Automatically discovering and learning new visual categories with ranking statistics," in *ICLR*, 2020.

[8] Ziliang Chen et al., "Deep cocktail networks," *IJCV*, vol. 129, no. 8, pp. 2328–2351, 2021.

[9] Chen et al., "Blending-target domain adaptation by adversarial meta-adaptation networks," in *CVPR*, 2019.

[10] Junyi Zhang et al., "Few-shot structured domain adaptation for virtual-to-real scene parsing," in *ICCV*, 2019.

[11] Jingyu Zhuang et al., "Domain adaptation for retinal vessel segmentation using asymmetrical maximum classifier discrepancy," in *ACM Turing*, 2019.

[12] Hong Liu et al., "Separate to adapt: Open set domain adaptation via progressive separation," in *CVPR*, 2019.

[13] Yadan Luo, Zijian Wang, Zi Huang, and Mahsa Baktashmotlagh, "Progressive graph learning for open-set domain adaptation," in *ICML*, 2020, pp. 6468–6478.

[14] Kai Han et al., "Learning to discover novel visual categories via deep transfer clustering," in *ICCV*, 2019.

[15] Ying Jin et al., "Minimum class confusion for versatile domain adaptation," in *ECCV*, 2020.

[16] Yaroslav Ganin et al., "Domain-adversarial training of neural networks," *Domain Adaptation in Computer Vision Applications*, 2017.

[17] V. Satopaa et al., "Finding a "kneedle" in a haystack: Detecting knee points in system behavior," in *ICDCS Workshops*.