Noname manuscript No.
(will be inserted by the editor)

Exploiting Inter-Sample Affinity for Knowability-Aware Universal Domain Adaptation

Yifan Wang[1]* · Lin Zhang[1]* · Ran Song[1] · Hongliang Li[2] · Paul L. Rosin[3] · Wei Zhang[1]

Received: date / Accepted: date

Abstract Universal domain adaptation (UniDA) aims to transfer the knowledge of common classes from the source domain to the target domain without any prior knowledge on the label set, which requires distinguishing in the target domain the unknown samples from the known ones. Recent methods usually focused on categorizing a target sample into one of the source classes rather than distinguishing known and unknown samples, which ignores the inter-sample affinity between known and unknown samples, and may lead to suboptimal performance. Aiming at this issue, we propose a novel UniDA framework where such inter-sample affinity is exploited. Specifically, we introduce a knowability-based labeling scheme which can be divided into two steps: 1) Knowability-guided detection of known and unknown samples based on the intrinsic structure of the neighborhoods of samples, where we leverage the first singular vectors of the affinity matrix to obtain the knowability of every target sample. 2) Label refinement based on neighborhood consistency to relabel the target samples, where we refine the labels of each target sample based on its neighborhood consistency of predictions. Then, auxiliary losses based on the two steps are used to reduce the inter-sample affinity between the unknown and the known target samples. Finally, experiments on four public datasets demonstrate that our method significantly outperforms existing state-of-the-art methods.

Keywords Domain adaptation · Representation Learning · Transfer Learning · Out-of-Distribution Detection

1 Introduction

Unsupervised domain adaptation (UDA) [9,37,27,11,55] aims to transfer the learned knowledge from the labeled source domain to the unlabeled target domain so that the inter-sample affinities in the latter can be properly measured.

The assumption of traditional UDA, i.e., closed-set DA, is that the source domain shares an identical label set with the target domain, which significantly limits its applications in real-world scenarios. Thus, several relaxations to this assumption have been investigated. Partial-set DA (PDA) [3,4,53,24] assumes that the target domain is not identical to the source domain but is a subset. On the contrary, Open-set DA (ODA) [30,38,26] assumes that the target domain contains classes

✉ Ran Song (corresponding author)
ransong@sdu.edu.cn

Yifan Wang
yi.fan.wang1216@gmail.com

Lin Zhang
zl935546110@gmail.com

Hongliang Li
hlli@uestc.edu.cn

Paul L. Rosin
rosinpl@cardiff.ac.uk

Wei Zhang
davidzhang@sdu.edu.cn

[1] School of Control Science and Engineering, Shandong University, Jinan, China

[2] School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China

[3] School of Computer Science and Informatics, Cardiff University, Cardiff, UK

* These authors contributed equally to this work.

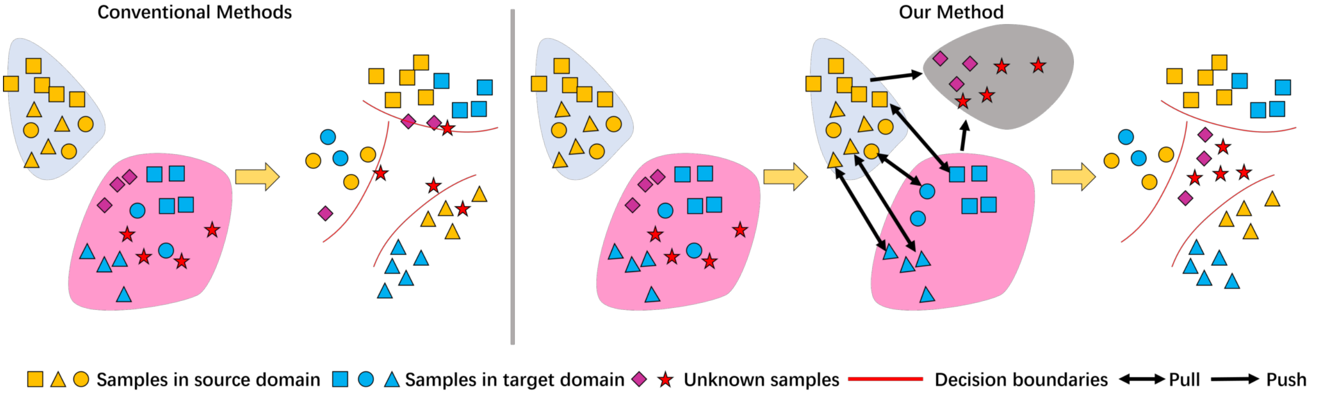


Fig. 1 Illustration of our method. Conventional methods usually focused on the known samples and might falsely recognise the unknown samples or ignore the inter-sample affinity between samples. Our method exploits the inter-sample affinity between known and unknown samples. The known samples in the target domain are pulled towards the corresponding samples in the source domain while the unknown samples are pushed away from any source samples.

unknown to the source domain such that the source domain is a subset of the target domain. Open-partial DA (OPDA) [36,22,8] introduces private classes in both domains, where the private classes in the target domain are unknown, and assumes that the common classes shared by the two domains have been identified. Universal DA (UniDA) [1,36,22] treats unsupervised DA in the most general setting, where no prior knowledge is required on the label set relationship between domains.

A popular method [49,1,8,36] for UniDA is to employ a classifier which produces a confidence for each sample to determine whether it belongs to a particular known class seen in the source domain or the unknown class. Such methods mostly rely on the posterior probability of a classifier, which may obtain satisfactory performance on the known samples. However, as shown in the left half of Fig. 1, once the known samples have been identified, simply ignoring unknown samples can easily lead to suboptimal classification performance for the unknown samples since such samples still contain meaningful information that can be leveraged to improve the learned representations. In addition, the classifier-based methods may generate overconfident predictions for the known classes, leading to bias towards the known samples and the failure to identify the unknown ones.

To solve this problem, some recent approaches aim to increase the inter-sample affinity within a known class to improve the reliability of the classification. For instance, Saito *et al.* [35] proposed to assign each target sample to either a target neighbor or a prototype of a source class via entropy optimisation. Li *et al.* [22] replaced the classifier-based framework with a clustering-based one to increase the inter-sample affinity within a known class. It exploited the intrinsic structure of neighbors to directly match the clusters in the source domain and those in the target domain to discovery common and private classes. Thus, they both increased the inter-sample affinity in known classes. However, since the inter-sample affinity between unknown samples can be greater than that between unknown and known samples due to the less discriminative features, this may lead to the misalignment between unknown samples and the prototypes in the source domain or the mismatch between the unknown clusters and the clusters in the source domain.

To mitigate such issues, we propose a novel UniDA framework which exploits the inter-sample affinity between unknown and known samples. We propose a knowability-based labeling scheme to distinguish known and unknown samples via knowability-guided detection and refine sample labels based on the neighborhood consistency of the predicted labels. Specifically, the scheme can be divided into two steps: 1) knowability-guided detection of known and unknown samples, where we decompose the affinity matrix of every target sample based on the k-nearest neighbors to obtain the first singular vectors as the robust representation of the local neighborhood structure and then compute the similarity between the first singular vector of each domain for every target sample to obtain the knowability; 2) label refinement based on neighborhood consistency to relabel the target samples, where each target sample is labeled via a credibility score, based on the predictions of its neighbors. Then, a target sample is labeled as known, unknown or uncertain through an automatic thresholding scheme to produce the threshold on-the-fly for the credibility score, which avoids setting the threshold manually as many existing works [35,22,8] did.

Next, we design three losses to impose a restriction on the target samples based on the above scheme. As illustrated in right half of Fig. 1, the restriction aims to 1) reduce the inter-sample affinity between the unknown and the known samples in the target domain and 2) increase the inter-sample affinity between the known samples in the target domain and some particular samples found by the k-NN algorithm in the source domain where such target and source samples are supposed to belong to the same known class.

In summary, the contributions of this paper are thus fourfold:

- We propose a novel method to exploit the inter-sample affinity between unknown and known samples for UniDA.
- We propose the knowability-guided detection of known and unknown samples and the label refinement based on the neighborhood consistency of each sample.
- We evaluate our method on four widely used UniDA benchmarks, i.e., Office-31 [34], OfficeHome [32], VisDA [33] and DomainNet [46] and the results demonstrate that our method considerably outperforms the state-of-the-art UniDA methods.

2 Related Work

We briefly review recent unsupervised DA methods in this section. According to the assumption made about the relationship between the label sets of different domains, we group these methods into three categories, namely PDA, ODA and UniDA. We also briefly review a related problem named Out-of-Distribution Detection as it is also closely related our work.

2.1 Partial-set Domain Adaptation

PDA requires that the source label set is larger than and contains the target label set. Many methods for PDA have been developed [2,3,53,4,24,23]. For example, Cao *et al.* [2] presented the selective adversarial network (SAN), which simultaneously circumvented negative transfer caused by private source classes and promoted positive transfer between common classes in both domains to align the distributions of samples in a fine-grained manner. Zhang *et al.* [53] proposed to identify common samples associated with domain similarities from the domain discriminator, and conducted a weighting operation based on such similarities for adversarial training. Cao *et al.* [4] proposed a progressive weighting scheme to estimate the transferability of source samples. Liang *et al.* [24] introduced balanced adversarial alignment and adaptive uncertainty suppression to avoid negative transfer and uncertainty propagation.

2.2 Open-set Domain Adaptation

ODA, first introduced by Busto *et al.* [30], assumes that there are private and common classes in both source and target domains, and the labels of the common classes are known as a priori knowledge. They introduced the Assign-and-Transform-Iteratively (ATI) algorithm to address this challenging problem.

Recently, one of the most popular strategies [26,7] for ODA is to draw the knowledge from the domain discriminator to identify common samples across domains. Saito *et al.* [38] proposed an adversarial learning framework to train a classifier to obtain a boundary between source and target samples whereas the feature generator was trained to make the target samples lie far from the boundary. Bucci *et al.* [1] employed self-supervised learning technique to achieve the known/unknown separation and domain alignment.

2.3 Universal Domain Adaptation

UniDA, first introduced by You *et al.* [49] is subject to the most general setting of unsupervised DA, which involves no prior knowledge about the difference of object classes between the two domains. You *et al.* also presented an universal adaptation network (UAN) to evaluate the transferability of samples based on uncertainty and domain similarity for solving the UniDA problem. However, the uncertainty and domain similarity measurements are sometimes unreliable and insufficiently discriminative. Thus, Fu *et al.* [8] proposed another transferability measure, known as Calibrated Multiple Uncertainties (CMU), estimated by a mixture of uncertainties which accurately quantified the inclination of a target sample to the common classes. Li *et al.* [22] introduced Domain Consensus Clustering (DCC) to exploit the domain consensus knowledge for discovering discriminative clusters in the samples, which differentiated the unknown classes from the common ones. The latest work OVANet [36], proposed by Saito *et al.*, trained a one-vs-all classifier for each class using labeled source samples and adapted the open-set classifier to the target domain.

2.4 Out-of-Distribution Detection

Out-of-Distribution (OOD) detection aims to detect OOD samples during the inference process which is enlightening to the UniDA problem of detecting unknown samples. Hendrycks *et al.* [14] first proposed a baseline method for detecting OOD samples using the confidence of classification. Recently, some methods [25,21, 39,17] built advanced detectors in a post-hoc manner. For example, Lee *et al.* [21] utilised the Mahalanobis distance between the features of test and the train samples to obtain the confidence score with respect to the closest class conditional distribution. However, these methods require many labeled samples for training. To better exploit the unlabeled data for OOD detection, Hendrycks *et al.* [15] enforced the model to produce the low confidence output on the pure unlabeled OOD data. Some other works [10,16,48,45,40] employed self-supervised learning on the pure unlabeled data to improve the performance. For instance, Sehwag *et al.* [40] combined contrastive learning and the Mahalanobis distance for OOD detection.

There also exist a line of works [29,18,42] which employed deep generative models on the pure unlabeled data. However, all of these methods require that the unlabeled data must be pure or OOD, which can hardly be met in realistic applications. Recently, some methods [5,51,12] considered the class distribution mismatch between labeled and unlabeled data, where the mismatched samples in the unlabeled data can be regarded as OOD samples. For example, Chen *et al.* [5] filtered out OOD samples in the unlabeled data with a confidence threshold and only utilised the remaining data for training. Yu *et al.* [51] proposed a joint optimisation framework to classify identification samples and filter out OOD samples concurrently. Guo *et al.* [12] employed bi-level optimization to weaken the weights of OOD samples. But these methods were developed for classifying identification samples and there were no OOD samples involved during the inference process. Yu *et al.* [50] attempted to utilise mixed unlabeled data for OOD detection, which encouraged two classifiers to maximally disagree on the mixed unlabeled data. However, since each unlabeled sample was treated equally, the model still required many labeled samples to distinguish between identification and OOD samples.

3 Method

In this section, we elaborate the major components of the proposed knowability-aware UniDA framework which sufficiently exploits the inter-sample affinity as stated in the introduction.

Notation Assume that we have the labeled set of source samples $\mathcal{X}^s = \{x_i^s\}_{i=1}^{n^s}$ defined with the known space of the source label set $\mathcal{Y}^s$ and the unlabeled set of target samples $\mathcal{X}^t = \{x_i^t\}_{i=1}^{n^t}$ where n^s and n^t indicate the numbers of the source and the target samples, respectively. Since the label spaces of the two domains are not aligned, we have the space of the target label set $\mathcal{Y}^t = \mathcal{Y}^{com} \cup \mathcal{Y}^{unk}$ with $\mathcal{Y}^{com} \subseteq \mathcal{Y}^s$. $\mathcal{Y}^{com}$ and $\mathcal{Y}^{unk}$ denote the spaces for the common label set which we called the known target label set and the unknown label set respectively where $\mathcal{Y}^{unk} \cap \mathcal{Y}^s = \emptyset$. The known classes are the classes that exist in the source domain, where the learned model is expected to have knowledge of the labels for such classes. The known samples refer to the target samples that belong to the known classes. The unknown classes include the objective classes of some target samples that do not exist in the source domain, where the model does not learn the label information of such classes. The unknown samples refer to the target samples whose labels are unknown to the model. With the training samples from both domains, the goal of UniDA is to learn an optimal classifier $C^t : \mathcal{X}^t \to \mathcal{Y}^t$ which categorises a target sample into either the *'unknown'* class or an object class belonging to $\mathcal{Y}^{com}$.

3.1 Overall Workflow

As shown in Fig. 2, we first extract a feature f_i from a sample x_i by the feature extractor $\mathcal{F}(\cdot \mid \phi)$ where $\cdot$ represents an input sample and ϕ denotes the set of trainable parameters of the feature extractor. To perform an effective k-nearest neighbor search, we first build two memory banks $\mathcal{M}^s$ and $\mathcal{M}^t$ to store the features in the source and the target domains respectively:

$$\mathcal{M}^s = [z_1^s, z_2^s, \cdots, z_{n^s}^s], \ \mathcal{M}^t = [z_1^t, z_2^t, \cdots, z_{n^t}^t]. \tag{1}$$

which are updated by a momentum strategy:

$$z_i^d = \alpha z_i^d + (1-\alpha) f_i^d, \quad f_i^d = F(x_i^d \mid \phi). \tag{2}$$

where α is the updating coefficient, $d \in \{s, t\}$.

We then search the neighbors for each target sample from the two memory banks $\mathcal{M}^s$ and $\mathcal{M}^t$ to establish the affinity relationship between samples. Updating the memory banks is crucial for ensuring effective discrimination between features from different classes to find reliable neighbors by the k-nearest neighbors algorithm. The updating strategy of the memory bank in Eq. (2) can progressively enhance the discrimination of features stored in the memory banks and reduce the intra-class variance between the given sample and its associated neighbors belonging to the same class from two domains. And the features with lower intra-class variance in the memory banks can effectively make the k-nearest neighbors algorithm more reliable.

Next, we utilise the affinity relationship to perform the knowability-guided detection of known/unknown samples and the label refinement based on neighborhood consistency. For the target samples, we categorise them into *known*, *unknown* and *uncertain* classes based

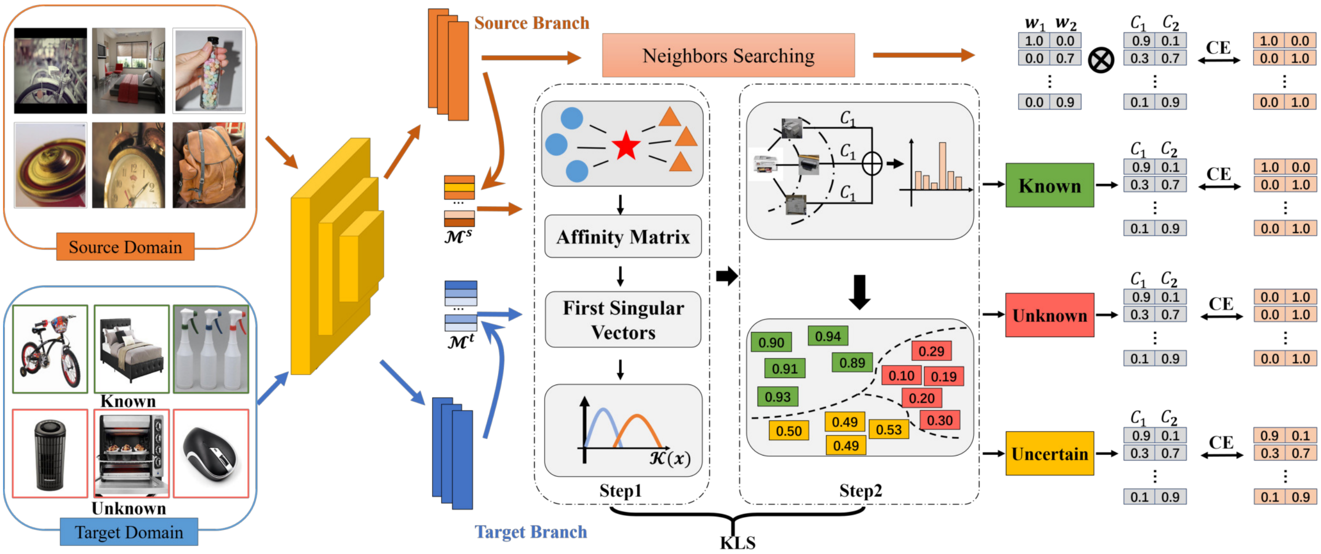


Fig. 2 The overall workflow of the proposed knowability-aware UniDA framework which exploits the inter-sample affinity. It leverages the knowability-guided detection of known/unknown samples and the label refinement based on neighborhood consistency to identify known samples and relabel them respectively. Both steps exploit the inter-sample affinity to obtain richer semantic information for every target sample. Finally, we use auxiliary losses to perform optimisation for our model to reduce the inter-sample affinity between the unknown and the known target samples.

on the above two steps. We then design three losses, expressed as $\mathcal{L}_k$, $\mathcal{L}_{unk}$, and $\mathcal{L}_{unc}$ for the three classes of samples, which set desired restrictions on them respectively by exploiting the inter-sample affinities. Meanwhile, we establish an inter-sample affinity weight matrix W_i for each sample in the source domain based on its neighbors, and then incorporate W_i into the total loss $\mathcal{L}_s$. Through minimising $\mathcal{L}_s$ during the training, the proposed method increases the inter-sample affinity within each class in the source domain whilst decreasing the inter-sample affinity between the samples of different classes in the source domain. Finally, we employ one classifier $\mathcal{C}(\cdot \mid \theta)$ defined in Eq. (3) to classify all samples subject to the four losses:

$$\mathcal{C}(\cdot \mid \theta) : \boldsymbol{x} \rightarrow \begin{bmatrix} \mathcal{C}_1^{(1)}(\cdot \mid \theta), \ldots, \mathcal{C}_1^{(Y)}(\cdot \mid \theta) \\ \mathcal{C}_2^{(1)}(\cdot \mid \theta), \ldots, \mathcal{C}_2^{(Y)}(\cdot \mid \theta) \end{bmatrix}^T \tag{3}$$

where the symbol θ denotes the set of parameters of the classifier implemented through a fully-connected layer. $\mathcal{C}_1^{(j)}(\cdot \mid \theta) + \mathcal{C}_2^{(j)}(\cdot \mid \theta) = 1$, and $\mathcal{C}_1^{(j)}$ and $\mathcal{C}_2^{(j)}$ represent the probabilities that a sample x_i^t is accepted or rejected as a member of an object class with index y in $\mathcal{Y}^s$ containing Y object classes, respectively. Since $\mathcal{C}_1^{(j)}$ and $\mathcal{C}_2^{(j)}$ are output together, we use $\mathcal{C}_2^{(j)}$ to represent $1-\mathcal{C}_1^{(j)}$ for readability. In the testing stage, for a target sample x_i^t, we define the reject score of x_i^t as the minimum value of reject probabilities. If $min_{j\in[1\ldots Y]}(\mathcal{C}_2^{(j)}(x_i^t \mid \theta)) > 0.5$, we regard x_i^t as an unknown target sample and otherwise a known target sample while the label $y_i = argmax_{j\in[1\ldots Y]}(\mathcal{C}_1^{(j)}(x_i^t \mid \theta))$.

3.2 Knowability-Based Labeling Scheme

In this section, we introduce the knowability-based labeling scheme (KLS) consisting of two steps which explore the label of a target sample based on the inter-sample affinity.

3.2.1 Knowability-Guided Detection of Known/Unknown Samples

To identify known and unknown samples, we explore the similarity of intrinsic structures of the neighborhood composed of source and target samples. With the assumption that the known target samples share similar semantics with the source samples, the distribution of the neighbors of a known sample from the target domain can be similar to that of a known sample from the source domain [22,47,54,43]. To this end, we formulate the knowability-guided detection based on the consistency of intrinsic structures between neighbors searched from two domains. In an effort to capture the intrinsic structure of neighbors, we propose to decompose the affinity matrices based on the k-nearest neighbors searched from both domains respectively to obtain the first singular vectors which robustly represent the intrinsic structures of the neighbors, as shown in Fig. 3. In fact, the first singular vector has already been proven to be used to select representatives of the class [52]. It is also used to obtain the degree of alignment between the representations and the eigenvector of affinity matrices of the representations for all classes, which uses

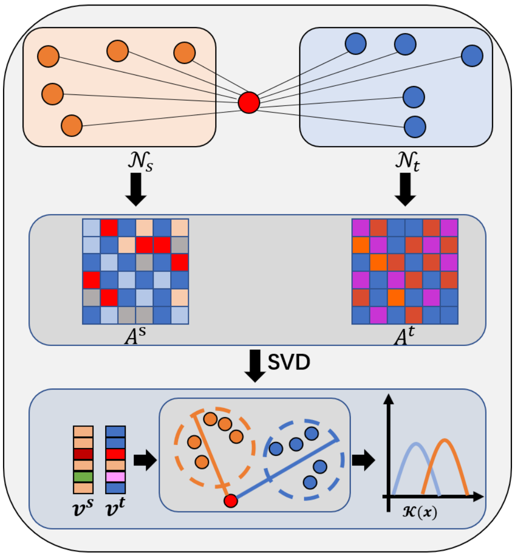


Fig. 3 Illustration of the computation of the knowability score. First, we search the neighbors of a target sample in both source and target domains. Then, we compute the affinity matrices of the neighbors in the source and the target domains, respectively. Next, we decompose each affinity matrix through SVD and obtain the first singular vectors of both matrices. Finally, we compute the knowability score defined as the cosine similarity of the two vectors.

the square of the inner product values between the representations and the first eigenvector to detect credible and incredible instances [19].

Specifically, given a target sample x_i^t, we first retrieve its k nearest neighbors from $\mathcal{M}^s$ and $\mathcal{M}^t$, denoted as $\mathcal{N}_i^s$ and $\mathcal{N}_i^t$, respectively:

$$\mathcal{N}_i^s = [z_{i0}^s, z_{i1}^s, \dots, z_{in}^s]^T, \quad \mathcal{N}_i^t = [z_{i0}^t, z_{i1}^t, \dots, z_{in}^t]^T, \tag{4}$$

where the sizes of $\mathcal{N}_i^s$ and $\mathcal{N}_i^t$ have to be equal. This may be a limitation in some applications. Then, we compute the affinity matrices A_i^s and A_i^t for $\mathcal{N}_i^s$ and $\mathcal{N}_i^t$, respectively:

$$A_i^s = \mathcal{N}_i^s(\mathcal{N}_i^s)^T, \quad A_i^t = \mathcal{N}_i^t(\mathcal{N}_i^t)^T. \tag{5}$$

Next, we compute the first singular vectors of A_i^s and A_i^t via SVD decomposition as

$$A_i^s = U_i^s \Sigma_i^s V_i^s, \; A_i^t = U_i^t \Sigma_i^t V_i^t, \tag{6}$$

where Σ_i^s and Σ_i^t are the decomposed diagonal matrices. We obtain the first eigenvectors v_i^s, v_i^t of V_i^s, V_i^t corresponding to the largest eigenvalues. Note that it is unnecessary to sort $\mathcal{N}_i^s$ and $\mathcal{N}_i^t$ by similarity with x_i^t. We do not care about the sorting order of elements in A_i^s and A_i^t as we utilize the SVD method to decompose them and the decomposition is not affected by the order of the elements in the affinity matrix. If we change the sorting order of the two sets $\mathcal{N}_i^s$ and $\mathcal{N}_i^t$, it is equivalent to performing elementary matrix transformations for the matrices A_i^s and A_i^t. Also, the singular vectors v_i^s and v_i^t are corresponding to the first singular values of A_i^s and A_i^t, respectively, which are free of the orders of elements in A_i^s and A_i^t.

The knowability score for the given samples x_i^t can be produced by cosine similarity between v_i^s and v_i^t:

$$k(x_i^t) = \frac{{v_i^s}^T v_i^t}{\|v_i^s\|_2 \|v_i^t\|_2}, \tag{7}$$

We can observe that $k(x_i^t)$ represents the discrepancy of the semantic distributions between $\mathcal{N}_i^t$ and $\mathcal{N}_i^s$. Generally, when $k(x_i^t)$ becomes large, it means that the major directions of the feature distributions of $\mathcal{N}_i^s$ and $\mathcal{N}_i^t$ are very close. Otherwise, when $k(x_i^t)$ becomes small, v_i^s is likely to be perpendicular to v_i^t, which means that the feature distributions of $\mathcal{N}_i^s$ is unrelated to that of $\mathcal{N}_i^t$. Since the samples sharing the same semantic information (i.e. known target and source samples) are more likely to have similar distributions, $k(x_i^t)$ of known samples are larger than those of unknown samples which do not share any semantic information with source samples. Thus, we divide these samples into known samples $\mathcal{D}_{known}$ and unknown samples $\mathcal{D}_{unknown}$ based on $k(x_i^t)$, respectively.

3.2.2 Label Refinement Based on Neighborhood Consistency

Since the distribution of the known target samples can be less-discriminative compared to that of the source samples due to the domain bias, we propose a label refinement method based on the consistency of the predicted labels of the neighbors. In this stage, we further refine the labels of samples in $\mathcal{D}_{known}$, where we label the credible samples in D_{known} and the samples from D_{known} as the known samples.

In detail, for each sample x_i^t in the target domain, we leverage the accepting probabilities of each sample from $\mathcal{N}_i^s$ produced by the classifier to compute the credibility score c_i:

$$c_i = max_{j \in [1 \dots Y]} \left(\frac{1}{\mid \mathcal{N}_i^s \mid} \sum\nolimits_{k \in \mathcal{N}_i} \mathcal{C}_1^{(j)}(z_k \mid \theta) \right) \tag{8}$$

where $\mathcal{N}_i^s$ denotes the set of indexes of the k-nearest neighbors in the source domain of the target sample x_i^t.

The lower c_i indicates that the predicted label of the target sample is highly dissimilar to any known class, suggesting that the target sample may lie near the decision boundary of the model. We identify such samples as unknown samples. In contrast, a target sample with a higher c_i is likely to be far away from the decision

Algorithm 1 Algorithm of KLS

```
Requirement: $x_i^t$, $c_\tau$, $\mathcal{N}_i^s$, $\mathcal{N}_i^t$
Step 1:
Compute $A_i^s$, $A_i^t$
Decompose $A_i^s$ and $A_i^t$ by Eq. (6)
Obtain $v_i^s$,$v_i^t$
Compute the knowability-score $k(x_i^t)$ by Eq. (7)
If $k(x_i^t) < k_\tau$ do
    Append $x_i^t$ to $\mathcal{D}_{unknown}$
Else do
    Append $x_i^t$ to $\mathcal{D}_{known}$
Step 2:
    Compute $c_\tau$ by Eq.(9)
If $x_i^t \in \mathcal{D}_{known}$ do
    Compute $c_i$ by Eq. (8)
    If $c_i > c_\tau$ do
        Obtain the pseudo label $\hat{y}_i^t$
        Label $x_i^t$ as $\hat{y}_i^t$
    Elif $c_i < 0.8c_\tau$ do
        Label $x_i^t$ as 'Unknown'
    Else do
        Label $x_i^t$ as 'Uncertain'
Elif $x_i^t \in \mathcal{D}_{unknown}$ do
    Label $x_i^t$ as 'Unknown'
end
```

boundary and can derive a more reliable pseudo label from its neighbors. Formally, if $c_i < 0.8c_\tau$, we regard x_i^t as an unknown sample. Note that the threshold c_τ is produced automatically and 0.8 is chosen empirically. Then, if $c_i > c_\tau$, x_i^t is recognised as a known sample. If $0.8c_\tau < c_i < c_\tau$, x_i^t is regarded as an uncertain sample (sensitivity of the scale coefficient for c_τ can be seen in Sec. 4.3).

Distinguishing the unknown samples from the known ones in the target domain is obviously affected by the choice of the threshold c_τ. However, varying sizes and categories of different datasets lead to the change of the optimal threshold. To avoid setting the threshold manually for each dataset, we introduce an auto-thresholding scheme. Notably, the threshold c_τ is calculated as the mean of the maximum values for the accepting probabilities $\mathcal{C}_1(x_i^s \mid \theta)$ of source samples in the mini-batch $\mathcal{B}$:

$$c_\tau = \frac{1}{\mid \mathcal{B} \mid} \sum_{i=1}^{|\mathcal{B}|} \max_{j\in[1..Y]} \left(\mathcal{C}_1^{(j)} \left(x_i^s \mid \theta \right) \right). \tag{9}$$

This scheme avoids setting different thresholds for different datasets manually. This step is also illustrated in Fig. 4 and the full algorithm of KLS is elaborated in Algorithm 1.

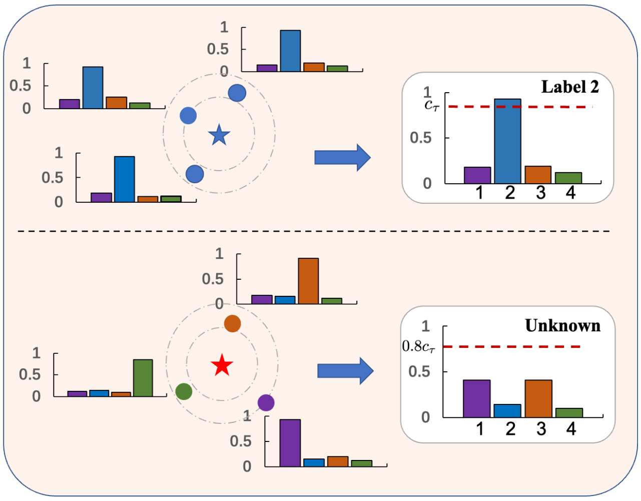


Fig. 4 Overview of the label refinement. We find the k-nearest neighbors from the source domain for each target sample. c_i is computed as the maximum value of the average accepting probabilities of the neighbors of each target sample.

3.3 Training Objectives

3.3.1 Target Domain Losses

Once we derive the known and the unknown samples from the above two steps, we propose the auxiliary losses to reduce the inter-sample affinity between the unknown and the known samples and increase that within a known class. Specifically, for an unknown sample, we hope to push the samples of all known classes away from it for reducing the inter-sample affinity between the unknown and the known samples. Thus we design the target-domain loss for the unknown samples, $\mathcal{L}_{unk}$, which minimizes the entropy of the reject probabilities for all classes:

$$\mathcal{L}_{unk}(x_i^t) = -\frac{1}{Y} \sum_{j=1}^{Y} \mathcal{C}_2^{(j)}(x_i^t \mid \theta) log \left(\mathcal{C}_2^{(j)}(x_i^t \mid \theta) \right). \tag{10}$$

For the known samples in the target domain, we define the pseudo label of x_i^t as:

$$\hat{y}_i^t = argmax_{j\in[1...Y]} (\sum\nolimits_{k\in\mathcal{N}_i} \mathcal{C}_1^{(j)}(x_k^s \mid \theta)) \tag{11}$$

where $argmax(\cdot)$ denotes the index of the biggest value in a vector. Since the discrepancies exist between the source and the target samples belonging to the same object class due to the domain gap, the inter-sample affinity between them cannot be as high as that between the source samples belonging to the same object class. Thus, to increase the inter-sample affinity within a known class in the target domain, we increase the inter-sample affinity between the known samples in the

```
Algorithm 2 Full algorithm of our method
Requirement: ($\mathcal{X}^s$, $\mathcal{Y}^s$), $\mathcal{X}^t$
while step < max step do
  Sample batch $\mathcal{B}^s$ from ($\mathcal{X}^s$, $\mathcal{Y}^s$) and batch $\mathcal{B}^t$ from $\mathcal{X}^t$
  Extract features from each of $\mathcal{B}^s$and $\mathcal{B}^t$
  If step == 0 do
    Initialize $\mathcal{M}^t$, $\mathcal{M}^s$
  Else do
    Update $\mathcal{M}^t$, $\mathcal{M}^s$
  for $x_i^s \in \mathcal{B}^s$ and $x_i^t \in \mathcal{B}^t$ do
    Compute $W_i$ for $x_i^s$
    Compute the source domain loss $\mathcal{L}_s$
    Label $x_i^t$ by KLS
    If $x_i^t$ has label $\hat{y}_i^s$ do
      Compute $\mathcal{L}_k$
    Elif $x_i^t$ has label 'unknown' do
      Compute $\mathcal{L}_{unk}$
    Else do
      Compute $\mathcal{L}_{unc}$
    Compute the overall loss $\mathcal{L}_{all}$
  Update the model
end
```

target domain and the corresponding samples with the pseudo label $\hat{y}_i^t$ in the source domain. This is achieved by designing the target-domain loss $\mathcal{L}_k$ which minimizes the entropy of the accepting probability of class $\hat{y}_i^t$:

$$\mathcal{L}_k(x_i^t) = -\mathcal{C}_1^{(\hat{y}_i^t)}(x_i^t \mid \theta) log\left(\mathcal{C}_1^{(\hat{y}_i^t)}(x_i^t \mid \theta)\right). \tag{12}$$

Moreover, it is difficult to distinguish uncertain samples as known or unknown ones. Therefore, we apply the self-supervised learning to minimize the sum of the average entropy of $\mathcal{C}_1^{(j)}$ and $\mathcal{C}_2^{(j)}$. Since $\mathcal{C}_1^{(j)} + \mathcal{C}_2^{(j)} = 1$ for any given class, by minimizing the entropy, the uncertain samples supposed to be known will have an increase in the confidence of belonging to one source class, while the uncertain samples supposed to be unknown will have an increase in the reject scores of each class. As such, the uncertain samples can be distinguished more reliably. We leverage a loss $\mathcal{L}_{unc}$ to minimize the average entropy of all classifiers to keep the inter-sample affinities low in every known classes:

$$\mathcal{L}_{unc}(x_i^t) = \frac{-1}{2Y}\sum_{k=1,2}\sum_{j=1}^{Y}\mathcal{C}_k^{(j)}(x_i^t \mid \theta) log\Big(\mathcal{C}_k^{(j)}(x_i^t \mid \theta)\Big). \tag{13}$$

The overall algorithm of our method is elaborated in Algorithm 2.

3.3.2 Source Domain Loss based on Inter-sample Affinity

For a sample x_i^s in the source domain with label y_i^s, to deliver a reliable classification, we should increase the inter-sample affinity within class y_i^s and reduce that between class y_i^s and other classes in the source domain. Thus, we propose the inter-sample affinity weight matrix $W_i = [w_1, w_2]^T$ for x_i^s where $w_1, w_2 \in \mathbb{R}^Y$ represent the weights associated with the classes which require to increase or decrease the inter-sample affinity, respectively. In detail, $w_1 = (\mathbf{1}(j = y_i^s))_{j=1}^{Y}$ is the one-hot vector of class y_i^s. And $w_2 = \left(w_2^{(j)}\right)_{j=1}^{Y}$ is computed based on the inter-sample affinities between x_i^s and the samples from other source classes by retrieving the k-nearest neighbors of x_i^s from the samples with the labels different from y_i^s in the source domain, expressed as:

$$w_2^{(j)} = norm(\frac{\mid \mathcal{N}_i^{(j)} \mid}{\mid \mathcal{N}_i \mid} * \frac{\mathcal{C}_1^{(j)}(x_i^s \mid \theta)}{\sum_{k \neq y_i^s}\mathcal{C}_1^{(k)}(x_i^s \mid \theta)^2}) \tag{14}$$

where $norm$ denotes the L1-normalisation and $*$ is the multiplication. $\mid \mathcal{N}_i^{(j)} \mid$ and $\mid \mathcal{N}_i \mid$ represent the number of the neighbors belonging to the label y_j^s and the total number of the retrieving neighbors of x_i^s respectively and note that $w_2^{y_i^s}$ is set to 0. According to the Eq. (14), the larger values in w_2 means that the samples in class j are closer to x_i^s. Then, we compute the source-domain loss $\mathcal{L}_s(x_i^s)$ based on the weighted inter-sample affinity:

$$\mathcal{L}_s(x_i^s) = -\log < W_i, \mathcal{C}(x_i^s \mid \theta) > \tag{15}$$

where $\langle \cdot, \cdot \rangle$ is the dot product operator.

3.4 Overall Loss for Both Domains

Overall, we train the classifier $\mathcal{C}(\cdot \mid \theta)$ and the feature extractor $\mathcal{F}(\cdot \mid \phi)$ with four losses and a hyper-parameter λ. The overall loss is expressed as:

$$\mathcal{L}_{all} = \mathcal{L}_s + \lambda(\mathcal{L}_{unk} + \mathcal{L}_k + \mathcal{L}_{unc}). \tag{16}$$

It is worth mentioning that differing from many existing UniDA methods [22,1,8,35], there is only one hyper-parameter in our method.

4 Experimental Results

We do experiments on several benchmarks, such as Office-31 (Saenko et al. [34]), OfficeHome (Peng et al. [32]), VisDA (Peng et al. [33]) and DomainNet (Venkateswara et al. [46]). In this section, we first introduce our experimental setups, including datasets, evaluation protocols and training details. Then, we compare our method with a set of the state-of-the-art (SOTA) UniDA methods. We also conduct extensive ablation studies to demonstrate the effectiveness of each component of the proposed method. All experiments were implemented on one RTX2080Ti 11GB GPU with PyTorch 1.7.1 [31].

4.1 Experimental Setups

4.1.1 Datasets and Evaluation Protocols

We conduct experiments on four datasets. Office-31 [34] consists of 4,652 images from three domains: DSLR (D), Amazon (A), and Webcam (W). OfficeHome [32] is a more challenging dataset, which consists of 15,500 images from 65 categories. It is made up of 4 domains: Artistic images (Ar), Clip-Art images (CI), Product images (Pr), and Real-World images (Rw). VisDA [33] is a large-scale dataset, where the source domain contains 15,000 synthetic images and the target domain consists of 5,000 images from the real world. DomainNet [46] is a larger DA dataset containing around 0.6 million images.

In this paper, we use the H-score in line with recent UniDA methods [8,22,36]. H-score, proposed by Fu *et al.* [8], is the harmonic mean of the accuracy on the common classes a_{com} and the accuracy on the unknown class a_{unk}:

$$h = \frac{2a_{com} \cdot a_{unk}}{a_{com} + a_{unk}}. \tag{17}$$

4.1.2 Training Details

We employ the ResNet-50 [13] backbone pretrained on ImageNet [6] and optimise the model using Nesterov momentum SGD with momentum of 0.9 and weight decay of 5×10^{-4} . The batch size is set to 36 for all datasets. The initial learning rate is set as 0.01 for the new layers and 0.001 for the backbone layers. The learning rate is decayed with the inverse learning rate decay scheduling. The updating coefficient α is set as 0.9. The number of neighbors retrieved is set differently for different datasets. For Office-31 (4,652 images in 31 categories) and OfficeHome (15,500 images in 65 categories), the numbers of retrieved neighbors (i.e., $\mid \mathcal{N}_i^s \mid$, $\mid \mathcal{N}_i^t \mid$ and $\mid \mathcal{N}_i \mid$) are all set to 10. For VisDA (20,000 images in total) and DomainNet (0.6 million images), we set them to 100, respectively. k_τ is set to 0.5 for all datasets. We set λ to 0.1 for all datasets.

Table 1 Results on Office-31 with UniDA setting (H-score).

Method	Office-31 (10/10/11)						
	A2D	A2W	D2A	D2W	W2D	W2A	Avg
UAN [49]	59.7	58.6	60.1	70.6	71.4	60.3	63.5
CMU [8]	68.1	67.3	71.4	79.3	80.4	72.2	73.1
DANCE [35]	78.6	71.5	79.9	91.4	87.9	72.2	80.3
DCC [22]	88.5	78.5	70.2	79.3	88.6	75.9	80.2
ROS [1]	71.4	71.3	81.0	94.6	95.3	79.2	82.1
USFDA [20]	85.5	79.8	**83.2**	90.6	88.7	81.2	84.8
OVANet [36]	85.8	79.4	80.1	95.4	94.3	84.0	86.5
Ours	**87.4**	**82.5**	80.6	**96.1**	**98.3**	**84.9**	**88.5**

4.2 Comparison with the SOTA Methods

4.2.1 Baselines

We compare our method with several SOTA methods under the same settings on the four datasets in Sec. 4.1.1, such as UAN [49], CMU [8] and DCC [22]. We aim to show that the knowability-based labeling scheme (KLS) is effective for UniDA, which employed a classifier to produce the confidence of each sample to determine whether it belongs to the unknown class or not. Also, we compare our method with OVANet [36] and DANCE [35] to show that it is important to reduce the inter-sample affinity between the unknown and the known samples.

4.2.2 Results in Main Datasets

Tables 1 and 2 list the results on Office-31 and OfficeHome, respectively. On Office-31, our method outperforms the SOTA methods by 2.0% in terms of the H-score on average. For the more challenging dataset OfficeHome which contains much more private classes than common classes, our method also made a significant improvement of 2.8% in terms of the H-score. Our method also achieves the SOTA performance on both VisDA AND DomainNet as shown in Table 3. Overall, according to the results of quantitative comparisons, our method achieves the SOTA performance in every dataset and most sub-tasks, which demonstrates the effectiveness of the main idea of our method that reduces the inter-sample affinity between the unknown and the known samples.

4.3 Ablation Studies

In this section, we provide specific analysis on several important issues and ablated studies to understand the behaviour of our method.

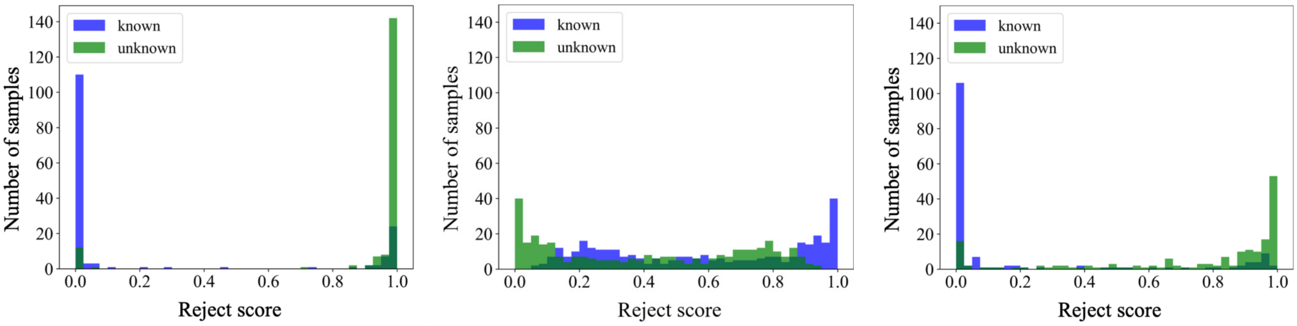
Quantitative Comparison on the Distribution of Reject Scores. To show the improvement on the distribution of reject scores which is the confidence of classifying the unknown samples as introduced in Sec. 3.1, we conducted experiments on Office-31(A2D). First, we plot the distributions of the reject scores of all sample in the target domain at the final epoch in Fig. 5 **(a)**. Then, we compare the plot to that trained on the source domain only in Fig. 5 **(b)**. We can observe that the full version of our method better distinguishes the known samples from the unknown ones. Furthermore, in Fig. 5 **(c)**, we show the corresponding plot produced by OVANet [36] for comparison. Noticeably,

Table 2 Results on OfficeHome with UniDA setting (H-score).

Method	OfficeHome (10/5/50)												
	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	Avg
OSBP[38]	39.6	45.1	46.2	45.7	45.2	46.8	45.3	40.5	45.8	45.1	41.6	46.9	44.5
UAN[49]	51.6	51.7	54.3	61.7	57.6	61.9	50.4	47.6	61.5	62.9	52.6	65.2	56.6
CMU[8]	56.0	56.9	59.1	66.9	64.2	67.8	54.7	51.0	66.3	68.2	57.8	69.7	61.6
OVANet[36]	62.8	75.6	78.6	70.7	68.8	75.0	71.3	58.6	80.5	76.1	64.1	78.9	71.8
Ours	**64.3**	**80.4**	**86.1**	**72.0**	**71.1**	**77.8**	**71.5**	**61.7**	**83.8**	**79.1**	**64.8**	**82.4**	**74.6**

Table 3 Results on DomainNet and VisDA with UniDA setting (H-score).

Method	DomainNet (150/50/145)							VisDA
	P2R	R2P	P2S	S2P	R2S	S2R	Avg	(6/3/3)
DANCE [35]	21.0	47.3	37.0	27.7	**46.7**	21.0	33.5	4.4
UAN [49]	41.9	43.6	39.1	38.9	38.7	43.7	41.0	30.5
CMU [8]	50.8	52.2	45.1	44.8	45.6	51.0	48.3	34.6
DCC [22]	56.9	50.3	43.7	44.9	43.3	56.2	49.2	43.0
OVANet [36]	56.0	51.7	47.1	47.4	44.9	57.2	50.7	53.1
Ours	**59.1**	**52.4**	**47.5**	**48.1**	45.1	**58.6**	**51.8**	**54.7**



(a) Ours (b) Source Only (c) OVANet

Fig. 5 Comparison on the distribution of reject scores. The three plots of histograms show the reject scores at the last epoch produced by the full version of our method, the model trained only on source domain, and the model trained on OVANet [36] in Office-31(A2D) respectively. Each area in dark green indicates that there is an overlap between the green and the blue bars.

our method performs better than OVANet [36] in terms of distinguishing the known samples from the unknown ones. However, it can be seen from Fig. 5 that negative transition also occurs, corresponding to the overlapping regions between the blue and the green bars. Such overlaps indicate that known samples are misclassified as unknown samples, or vice versa. Domain gap is the primary reason for the observed negative transition, which hinders the accurate classification of known and unknown samples.

Ability of Detecting Completely New Unknown Samples. To further show our model's ability of detecting completely new unknown samples not included in the training dataset, we conduct experiments using completely new testing datasets and show the results in Table 4. Specifically, the model was trained on Office-31 and tested on the subsets 'Art' and 'Clipart' of OfficeHome. Both subsets comprise samples that do not belong to any known classes. It can be seen that our method performs well in detecting completely new unknown samples, showcasing superior performance compared to the recent baseline OVANet [36].

Justification of the Knowability-Guided Detection. To justify the knowability-guided detection, we visualise the distributions of the knowability of samples on Office-31 (A2D). As plotted in Fig. 6, the distributions of the knowability of the known and the unknown samples have little overlap, which indicates that the unknown samples can be reliably distinguished from known ones by the knowability-guided detection. We also conduct experiments to monitor the changes in knowability scores throughout the training process for sub-tasks A2W and D2W on Office-31 and show the results in Fig. 7. We can observe that the mean knowability score of known samples consistently increases throughout the training process. It indicates that the inherent distribution of the target samples is progressively becoming more similar to that of the source samples be-

Table 4 Evaluation of models trained on Office-31 using new testing samples from the subsets 'Art' and 'Clipart' of Office-Home.

Testing Datasets	Method	Training Sub-tasks on Office-31 (10/10/11) A2D	A2W	D2A	D2W	W2D	W2A	Avg
Art	Ours	72.2	74.6	100	100	99.1	99.1	96.9
	OVANet [36]	61.6	71.2	99.1	98.3	100	99.1	88.2
Clipart	Ours	67.4	72.0	100	99.6	98.7	100	89.6
	OVANet [36]	69.1	70.4	99.6	100	98.7	99.1	89.4

Table 5 Results produced with different values of k on Office-31.

k	5	7	9	10	11	13	15	20	30	40	50	70	90
A2D	87.0	87.2	87.6	87.4	87.4	87.9	87.8	**88.2**	87.2	86.4	86.0	85.2	84.0
D2A	80.7	80.8	80.5	80.6	80.2	**80.8**	80.1	79.5	78.9	78.0	77.2	75.0	72.5
D2W	96.2	**96.6**	96.3	96.1	96.3	96.5	96.5	95.8	95.3	94.2	93.0	91.2	89.9
W2D	98.1	98.0	98.2	**98.3**	98.3	98.1	97.6	97.3	96.5	96.1	95.2	93.6	92.8

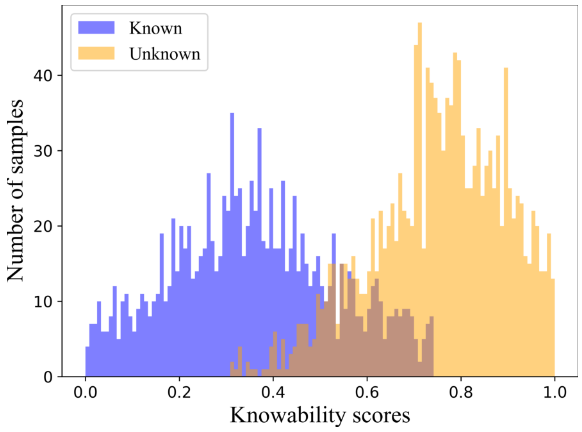


Fig. 6 The distribution of the knowability score.

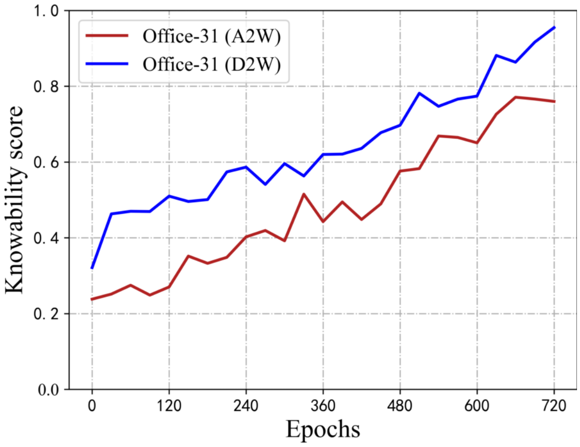
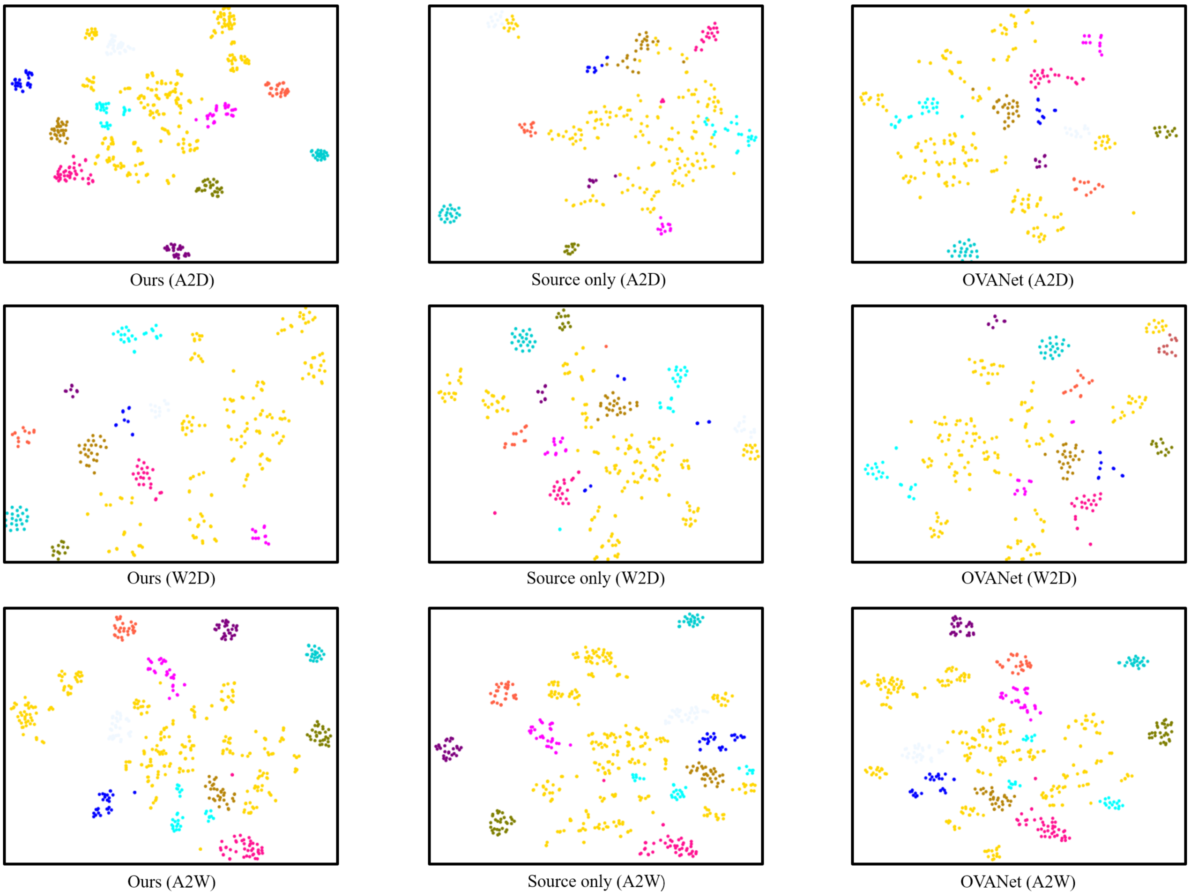


Fig. 7 The knowability score computed as the average of the knowability of all known target samples in one epoch.

longing to the same class. Moreover, the increased similarity also indicates that the inter-sample affinity between the source classes and the known target classes becomes higher.

Effect of the Number of Neighbors. We conduct experiments to explore the influence of different values of k on the k-nearest neighbor calculation. As shown in Table 5, every dataset has an optimal value of k related to the size of source domain. When k is larger than the optimal value, the performance tends to decrease. Although increasing the value of k moderately can enhance the reliability of the first singular vector, setting k to a large value leads to a significant increase of the noise in the neighborhood, which is influenced by the size of each category in the two domains. For example, since the subset 'Amazon' is three times larger than the subset 'Webcam', we can observe that the optimal values in sub-tasks A2D and D2A are larger than those in D2W and W2D. Moreover, increasing k will significantly raise the computational cost. But it does not mean that we can always increase k to pursue a performance gain when we have enough computational resources. Therefore, to achieve the optimal performance on average over all sub-tasks and save the computational resources, we select an appropriate value $k = 10$.

Qualitative Comparison by t-SNE Visualisations. Then, we use t-SNE [28] to visualise the features extracted by the feature extractor $\mathcal{F}(\cdot \mid \phi)$ for the model trained only with the source samples, OVANet, and the proposed method on Office-31 (A2D, W2D and A2W). As shown in Fig. 8, before the adaptation to the target domain (middle column), there exists significant misalignment. After the adaptation with the training via OVANet (right column) and our method (left column), the features become more discriminative. We observe better domain alignment as well as target category separation produced by our method. Note that although OVANet does succeed in aligning the source and the target domains and can detect the unknown class, it does not necessarily produce discriminative features for each known class. Moreover, compared to the model trained only with the source samples, the visualisation



Ours (A2D) Source only (A2D) OVANet (A2D)

Ours (W2D) Source only (W2D) OVANet (W2D)

Ours (A2W) Source only (A2W) OVANet (A2W)

Fig. 8 t-SNE visualisations of the classification results produced with different configuration. Different colours represent different classes. Yellow points represent the unknown samples and the points in other colours represent the known samples of different classes. The black dash lines represent the boundaries between the unknown and the known samples while the dash lines in other colours represent the boundaries of the corresponding known classes.

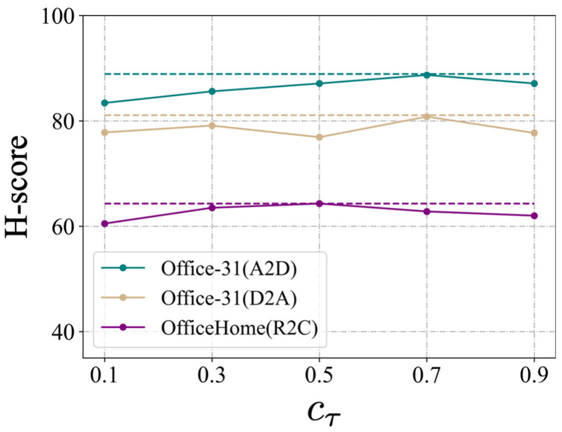


Fig. 9 Comparison of H-score performance subject to different thresholds where the solid lines represent the results of human-picked thresholds and the dash lines represent the results of the proposed auto-thresholding scheme.

of our method shows that the inter-sample affinity in each known class increases while that between different classes decreases.

Effect of the Auto-Thresholding Scheme. To show the effect of the proposed threshold c_τ, we compare it with the human-picked thresholds on Office-31 (A2D, D2A) and OfficeHome (R2C). From Fig. 9, we can observe that it is difficult to choose a consistently optimal threshold for all datasets and sub-tasks as the model is sensitive to the thresholds.

Accuracy of KLS. We conduct experiments on all sub-tasks of Office-31 where we record the accuracy of KLS for detecting the known/unknown samples at different training steps. As plotted in Fig. 10, each bar in the histogram represents the number of steps at which a particular accuracy of detecting known or unknown samples is achieved. For example, the top left plot with regard to the sub-task A2D on Office-31 shows that the number of steps at which the accuracy of known samples achieves 1 is approximately $3,500$. Moreover, there are approximately $1,000$ steps, where the accuracy of unknown samples achieves 1. In Fig. 10, the labeling scheme is consistently estimated with high accuracy which far surpasses 0.7 on average for both

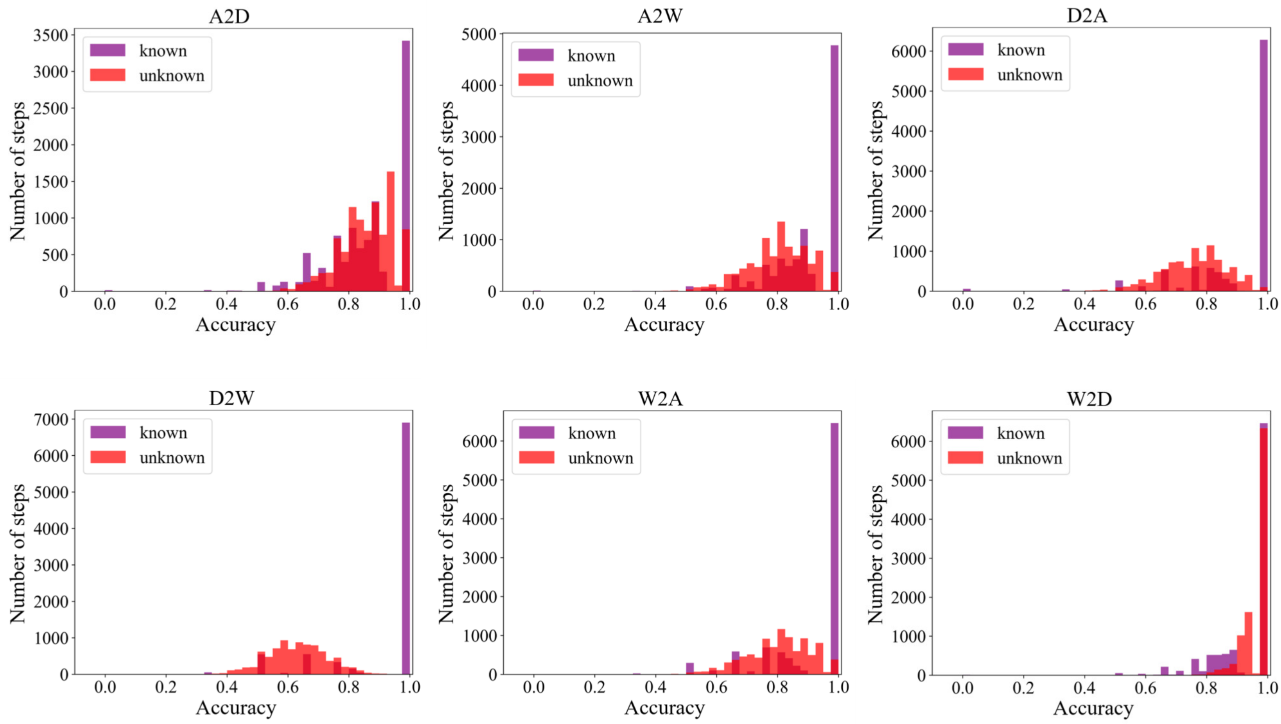
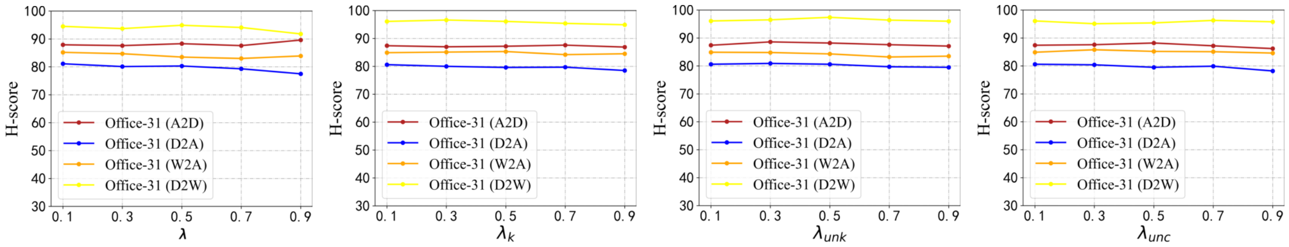


Fig. 10 Accuracy of the label refinement. Each plot is a histogram illustrating the number of steps at which a particular accuracy of detecting known or unknown samples is achieved. Plots from left to right in the top row correspond to the sub-tasks A2D, A2W and D2A on Office-31, respectively. Plots from left to right in the bottom row correspond to the sub-tasks D2W, W2A and W2D on Office-31, respectively.



(a) $\lambda = \lambda_k = \lambda_{unk} = \lambda_{unc}$ (b) $\lambda_k|\lambda_{unk} = \lambda_{unc} = 0.1$ (c) $\lambda_{unk}|\lambda_k = \lambda_{unc} = 0.1$ (d) $\lambda_{unc}|\lambda_k = \lambda_{unk} = 0.1$

Fig. 11 Sensitivity to λ, λ_k, λ_{unk} and λ_{unc} in terms of H-score. **(a)** We show the results with different values of λ where we set λ, λ_k, λ_{unk} and λ_{unc} all the same. **(b), (c) and (d)** We set λ_k, λ_{unk} and λ_{unc} separately and the results show that our model has a stable performance on different testing sub-tasks.

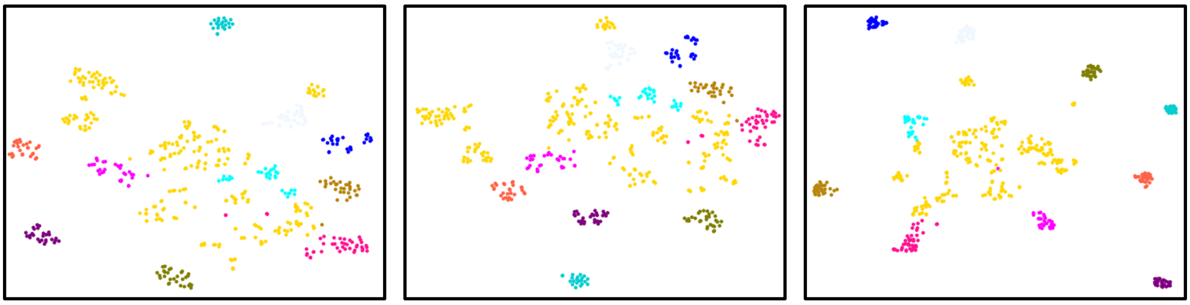
known and unknown samples. Thus, through the proposed knowability-based labeling scheme, our approach reliably finds the unknown and the known samples in the target domain.

Sensitivity of the Hyper-parameter λ. There is only one hyper-parameter λ in our model. To show the sensitivity of λ in the total loss, we conducted experiments on Office-31 with the UniDA setting. Please note the scale of $\mathcal{L}_{unk} + \mathcal{L}_k + \mathcal{L}_{unc}$ is usually much bigger than $\mathcal{L}_s$ because the training on source samples is supervised. Fig. 11 (a) shows that our method has a highly stable performance over different values of λ. To further demonstrate the effect of each loss functions, we replace λ with λ_{unk}, λ_k, and λ_{unc} as follows:

$$\mathcal{L}_{all} = \mathcal{L}_s + \lambda_{unk}\mathcal{L}_{unk} + \lambda_k\mathcal{L}_k + \lambda_{unc}\mathcal{L}_{unc}. \quad (18)$$

We conduct experiments where the hyper-parameters λ_{unk}, λ_k, and λ_{unc} are set separately and show the results in Fig. 11 **(b)**, **(c)** and **(d)**. It can be seen that our method is not sensitive to the change of the hyper-parameters λ_{unk}, λ_k, and λ_{unc}. Thus, we just set them all the same.

Effect of the Proposed Losses. We provide an ablation study to investigate the effect of each loss in our



w/o $\mathcal{L}_{unk}$ (D2W) w/o $\mathcal{L}_k$ (D2W) Ours (D2W)

Fig. 12 t-SNE visualisations on Office-31 (D2W). Different colors represent different classes. Yellow points represent the unknown samples and the points in other colours represent the known samples of different classes.

Table 6 Results of different ablated versions of our method on Office-31.

Method	Office-31 (10/10/11)						
	A2D	A2W	D2A	D2W	W2D	W2A	Avg
w/o $\mathcal{L}_s$	29.2	33.4	31.3	52.5	44.2	27.9	36.4
w/o $\mathcal{L}_{unk}$	81.0	77.5	78.2	95.0	91.0	72.9	82.6
w/o $\mathcal{L}_{unc}$	86.9	76.6	84.4	91.4	93.3	85.6	86.3
w/o $\mathcal{L}_k$	86.2	80.6	79.5	93.9	97.5	81.8	86.5
Ours	89.5	84.9	89.7	93.7	85.8	88.5	88.7

Table 7 Results on Office-31 with different scales of c_τ under the UniDA setting (H-score).

	Office-31 (10/10/11)						
	A2D	A2W	D2A	D2W	W2D	W2A	Avg
$0.1c_\tau$	84.6	81.7	83.2	94.0	97.5	**86.1**	87.9
$0.3c_\tau$	86.3	81.2	**82.6**	**95.0**	97.4	85.0	87.9
$0.5c_\tau$	88.1	**83.3**	81.7	94.4	97.0	85.4	88.3
$0.7c_\tau$	88.4	82.9	81.0	93.8	97.9	84.8	88.1
$0.8c_\tau$	**88.9**	83.0	81.1	94.5	98.3	85.2	**88.4**
$0.9c_\tau$	88.8	83.1	80.3	94.5	**99.2**	84.1	88.3

UniDA framework and show the results in Table 6. We can see that all losses contribute to the improvement of the results. In particular, among the three target-domain losses, $\mathcal{L}_{unk}$ has the largest impact on the final performance, which demonstrates that it is very important to reduce the inter-sample affinity between the unknown samples and the known ones.

To further show the effect of the proposed losses, we use t-SNE algorithm to visualise the features of target samples on Office-31 (D2W). As plotted in Fig. 12, without $\mathcal{L}_{unk}$ (left), the boundary between the unknown and the known samples is unclear. Without $\mathcal{L}_k$ (middle), samples belonging to a known class are not compact. However, the inter-sample affinity between the unknown and the known samples produced by the full version of our method (right), is much lower than that produced without $\mathcal{L}_{unk}$. And the inter-sample affinity in a known class produced by the full version of our method is much higher than that produced without $\mathcal{L}_k$. Such results demonstrate the main idea of the proposed method.

Sensitivity of scales for c_τ. Instead of using an automatic scheme, we set the parameter c_τ to 0.8 empirically. This is because changing c_τ has little influence on the performance. To verify this point, we test our method with different c_τ and show the results in Table 7.

Visual Explanations with Grad-CAM. In this section, we utilise the visualisation technique Grad-CAM in [41] to visualise the predictions and compare the Grad-CAM visualisations [41] for different methods in Fig. 13. To verify the validity of our method, we also visualise the previous methods including the source only model (second row) and DANCE [35] (third row) as well as OVANet [36] (fourth row) on their predictions. Obviously, we can observe that the semantic capabilities of our method (fifth row) are significantly stronger than OVANet [36] and DANCE [35]. We can also notice that our method concentrates on more relevant regions and the features of principal regions are accentuated, which verifies that our method indeed achieves an improvement for the critical parts in classification. The main reason is that our model learns discriminative information from each part and captures diverse relevant regions, while DANCE [35] and OVANet [36] are usually distracted by and even focus on some irrelevant area.

Performance of Using VGGNet as Backbone. Table 8 shows the quantitative comparison with the ODA setting on Office-31 using VGGNet [44] instead of ResNet50 as the backbone for feature extraction. According to the results, we demonstrate that our method is also effective with another backbone without changing any hyper-parameters.

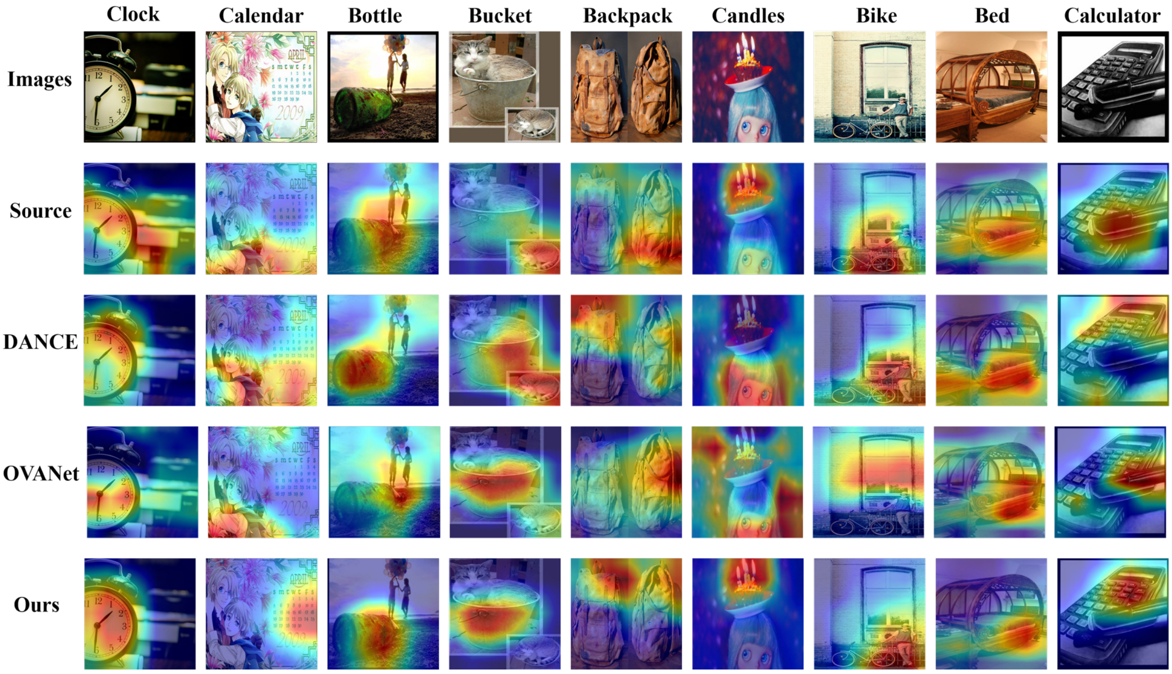


Fig. 13 Grad-CAM [41] visualisations of different methods on the sub-task R2A of OfficeHome. Generally, our method shows good concentration on known target samples and focuses on a variety of relevant regions.

Table 8 Results on Office-31 using the VGGNet [44] backbone with the ODA setting.

Method	Office-31 (10/10/11)						
	A2D	A2W	D2A	D2W	W2D	W2A	Avg
OSBP [38]	81.0	77.5	78.2	**95.0**	91.0	72.9	82.6
ROS [1]	79.0	81.0	78.1	94.4	**99.7**	74.1	84.4
OVANet [36]	**89.5**	**84.9**	89.7	93.7	85.8	88.5	88.7
Ours	89.5	84.6	**92.0**	94.5	91.5	**91.8**	**90.6**

5 Conclusions

In this paper, we propose a new framework to explore the inter-sample affinity for UniDA. Its core idea is to reduce the inter-sample affinity between the unknown and the known samples while increasing that within the known samples by estimating the knowability of each sample. Extensive experiments demonstrate that our method achieves the SOTA performance in various sub-tasks on four public datasets.

A limitation of our method is that it does not sufficiently utilise the inter-sample relationship within the set of unknown samples. Thus in the future work, we plan to extend our method to leverage this relationship for further boosting the performance with the UniDA setting. Moreover, since the proposed method assumes that the local affinity distributions of source and target samples within the same class are similar, we will explore the scenario where the distribution of samples in a known category is heterogeneous and differs between source and target domains in the future work.

Data Availability Statement

The datasets generated during and/or analysed during the current study are available in the Office-31 repository [Link], the OfficeHome repository [Link], the VisDA repository [Link], and the DomainNet repository [Link].

Declarations

The authors have no relevant financial or non-financial interests to disclose.

References

1. Bucci, S., Loghmani, M.R., Tommasi, T.: On the effectiveness of image rotation for open set domain adaptation. In: European Conference on Computer Vision. pp. 422–438. Springer (2020)
2. Cao, Z., Long, M., Wang, J., Jordan, M.I.: Partial transfer learning with selective adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2724–2732 (2018)
3. Cao, Z., Ma, L., Long, M., Wang, J.: Partial adversarial domain adaptation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 135–150 (2018)
4. Cao, Z., You, K., Long, M., Wang, J., Yang, Q.: Learning to transfer examples for partial domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2985–2994 (2019)

5. Chen, Y., Zhu, X., Li, W., Gong, S.: Semi-supervised learning under class distribution mismatch. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 3569–3576 (2020)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
7. Feng, Q., Kang, G., Fan, H., Yang, Y.: Attract or distract: Exploit the margin of open set. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7990–7999 (2019)
8. Fu, B., Cao, Z., Long, M., Wang, J.: Learning to detect open classes for universal domain adaptation. In: European Conference on Computer Vision. pp. 567–583. Springer (2020)
9. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International conference on machine learning. pp. 1180–1189. PMLR (2015)
10. Golan, I., El-Yaniv, R.: Deep anomaly detection using geometric transformations. Advances in neural information processing systems **31** (2018)
11. Gong, B., Grauman, K., Sha, F.: Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In: International Conference on Machine Learning. pp. 222–230. PMLR (2013)
12. Guo, L.Z., Zhang, Z.Y., Jiang, Y., Li, Y.F., Zhou, Z.H.: Safe deep semi-supervised learning for unseen-class unlabeled data. In: International Conference on Machine Learning. pp. 3897–3906. PMLR (2020)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
14. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint arXiv:1610.02136 (2016)
15. Hendrycks, D., Mazeika, M., Dietterich, T.: Deep anomaly detection with outlier exposure. arXiv preprint arXiv:1812.04606 (2018)
16. Hendrycks, D., Mazeika, M., Kadavath, S., Song, D.: Using self-supervised learning can improve model robustness and uncertainty. Advances in neural information processing systems **32** (2019)
17. Hsu, Y.C., Shen, Y., Jin, H., Kira, Z.: Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10951–10960 (2020)
18. Huang, Y., Dai, S., Nguyen, T., Baraniuk, R.G., Anandkumar, A.: Out-of-distribution detection using neural rendering generative models. arXiv preprint arXiv:1907.04572 (2019)
19. Kim, T., Ko, J., Choi, J., Yun, S.Y., et al.: Fine samples for learning with noisy labels. Advances in Neural Information Processing Systems **34**, 24137–24149 (2021)
20. Kundu, J.N., Venkat, N., Babu, R.V., et al.: Universal source-free domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4544–4553 (2020)
21. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. Advances in neural information processing systems **31** (2018)
22. Li, G., Kang, G., Zhu, Y., Wei, Y., Yang, Y.: Domain consensus clustering for universal domain adaptation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
23. Liang, J., Hu, D., Feng, J., He, R.: Dine: Domain adaptation from single and multiple black-box predictors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8003–8013 (2022)
24. Liang, J., Wang, Y., Hu, D., He, R., Feng, J.: A balanced and uncertainty-aware approach for partial domain adaptation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16. pp. 123–140. Springer (2020)
25. Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks. arXiv preprint arXiv:1706.02690 (2017)
26. Liu, H., Cao, Z., Long, M., Wang, J., Yang, Q.: Separate to adapt: Open set domain adaptation via progressive separation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2927–2936 (2019)
27. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. arXiv preprint arXiv:1602.04433 (2016)
28. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
29. Nalisnick, E.T., Matsukawa, A., Teh, Y.W., Lakshminarayanan, B.: Detecting out-of-distribution inputs to deep generative models using a test for typicality. (2019)
30. Panareda Busto, P., Gall, J.: Open set domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 754–763 (2017)
31. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32**, 8026–8037 (2019)
32. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1406–1415 (2019)
33. Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., Saenko, K.: Visda: The visual domain adaptation challenge. arXiv preprint arXiv:1710.06924 (2017)
34. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: European conference on computer vision. pp. 213–226. Springer (2010)
35. Saito, K., Kim, D., Sclaroff, S., Saenko, K.: Universal domain adaptation through self-supervision. In: Advances in Neural Information Processing Systems. p. 16282–16292 (2020)
36. Saito, K., Saenko, K.: Ovanet: One-vs-all network for universal domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9000–9009 (2021)
37. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3723–3732 (2018)
38. Saito, K., Yamamoto, S., Ushiku, Y., Harada, T.: Open set domain adaptation by backpropagation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 153–168 (2018)
39. Sastry, C.S., Oore, S.: Detecting out-of-distribution examples with gram matrices. In: International Conference on Machine Learning. pp. 8491–8501. PMLR (2020)
40. Sehwag, V., Chiang, M., Mittal, P.: Ssd: A unified framework for self-supervised outlier detection. arXiv preprint arXiv:2103.12051 (2021)

41. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
42. Serrà, J., Álvarez, D., Gómez, V., Slizovskaia, O., Núñez, J.F., Luque, J.: Input complexity and out-of-distribution detection with likelihood-based generative models. arXiv preprint arXiv:1909.11480 (2019)
43. Sharma, A., Kalluri, T., Chandraker, M.: Instance level affinity-based transfer for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5361–5371 (2021)
44. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
45. Tack, J., Mo, S., Jeong, J., Shin, J.: Csi: Novelty detection via contrastive learning on distributionally shifted instances. Advances in neural information processing systems **33**, 11839–11852 (2020)
46. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5018–5027 (2017)
47. Wang, S., Zhao, D., Zhang, C., Guo, Y., Zang, Q., Gu, Y., Li, Y., Jiao, L.: Cluster alignment with target knowledge mining for unsupervised domain adaptation semantic segmentation. IEEE Transactions on Image Processing **31**, 7403–7418 (2022)
48. Winkens, J., Bunel, R., Roy, A.G., Stanforth, R., Natarajan, V., Ledsam, J.R., MacWilliams, P., Kohli, P., Karthikesalingam, A., Kohl, S., et al.: Contrastive training for improved out-of-distribution detection. arXiv preprint arXiv:2007.05566 (2020)
49. You, K., Long, M., Cao, Z., Wang, J., Jordan, M.I.: Universal domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2720–2729 (2019)
50. Yu, Q., Aizawa, K.: Unsupervised out-of-distribution detection by maximum classifier discrepancy. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9518–9526 (2019)
51. Yu, Q., Ikami, D., Irie, G., Aizawa, K.: Multi-task curriculum framework for open-set semi-supervised learning. In: European Conference on Computer Vision. pp. 438–454. Springer (2020)
52. Zaeemzadeh, A., Joneidi, M., Rahnavard, N., Shah, M.: Iterative projection and matching: Finding structure-preserving representatives and its application to computer vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5414–5423 (2019)
53. Zhang, J., Ding, Z., Li, W., Ogunbona, P.: Importance weighted adversarial nets for partial domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8156–8164 (2018)
54. Zhao, Y., Cai, L., et al.: Reducing the covariate shift by mirror samples in cross domain alignment. Advances in Neural Information Processing Systems **34**, 9546–9558 (2021)
55. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European conference on computer vision (ECCV). pp. 289–305 (2018)