Pattern Recognition 146 (2024) 110023

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

# Inter-domain mixup for semi-supervised domain adaptation

Jichang Li [a,b], Guanbin Li [a,*], Yizhou Yu [b]

[a] School of Computer Science and Engineering, Research Institute of Sun Yat-sen University in Shenzhen, Sun Yat-sen University, Guangzhou, China
[b] Department of Computer Science, The University of Hong Kong, Hong Kong

ARTICLE INFO

*Keywords:*
Semi-supervised domain adaptation
Inter-domain mixup
Neighborhood expansion

ABSTRACT

Semi-supervised domain adaptation (SSDA) aims to bridge source and target domain distributions, with a small number of target labels available, achieving better classification performance than unsupervised domain adaptation (UDA). However, existing SSDA work fails to make full use of label information from both source and target domains for feature alignment across domains, resulting in label mismatch in the label space during model testing. This paper presents a novel SSDA approach, Inter-domain Mixup with Neighborhood Expansion (IDMNE), to tackle this issue. Firstly, we introduce a cross-domain feature alignment strategy, Inter-domain Mixup, that incorporates label information into model adaptation. Specifically, we employ sample-level and manifold-level data mixing to generate compatible training samples. These newly established samples, combined with reliable and actual label information, display diversity and compatibility across domains, while such extra supervision thus facilitates cross-domain feature alignment and mitigates label mismatch. Additionally, we utilize Neighborhood Expansion to leverage high-confidence pseudo-labeled samples in the target domain, diversifying the label information of the target domain and thereby further increasing the performance of the adaptation model. Accordingly, the proposed approach outperforms existing state-of-the-art methods, achieving significant accuracy improvements on popular SSDA benchmarks, including DomainNet, Office-Home, and Office-31.

## 1. Introduction

Domain adaptation (DA) aims to first train an adaptation model from label-rich datasets (a.k.a source domain) and then transfer the learned knowledge to a new but label-scarce dataset (a.k.a target domain) of different distribution so as to avoid relying largely on the human-annotated in-distribution data. Nowadays, application scenarios of DA include semantic segmentation [1], object detection [2], person re-identification [3], and so on. Unfortunately, a direct application of such models trained on the source domain dataset to the target domain would cause severe performance degradation due to different data distributions in these two domains. The commonly defined DA setting, unsupervised domain adaptation (UDA), witnesses great progress in the reduction of domain shift [4]. Nevertheless, compared with UDA, semi-supervised domain adaptation (SSDA), where a small number of target labels are available, achieves much better performance on the target domain. This is because supervision on a few labeled target domain samples, as well as a large number of labeled source domain samples, is already capable of bridging partial distribution discrepancies across domains [5,6].

Previous approaches for the SSDA problem extract domain-invariant features mainly by minimizing cross-domain discrepancy measures [7], relying on image style transfer [8] or adversarial training [5,6,9]. They can achieve domain alignment so that, in theory, or in hypothesis, the adapted classifier is prone to obtain good classification performance in the target domain. However, these strategies to enforce domain-level feature alignment may fail to generate discriminative target features for two main reasons. First, the source domain has a much larger number of labeled samples than the target domain when we perform supervised learning using labeled samples across domains. Thus, features of labeled target domain samples do not have the same level of diversity as those of source domain samples [6,10], impairing their discriminability. In addition, the majority of previous strategies performed to cross-domain feature alignment neglect label information from either the source or target domain for adaptation and the model consequently cannot align sample features from both domains according to their class labels [11]. As a result, as shown in Fig. 1, domain-invariant yet non-discriminative features generated from the model are used to align different categories, thereby giving rise to cross-domain label mismatch in the label space [12]. To alleviate this issue, existing label-free feature alignment strategies have to impose more crafted constraints on the target domain [6,7,13]; nevertheless, unsupervised learning based

* Corresponding author.
*E-mail addresses:* csjcli@connect.hku.hk (J. Li), liguanbin@mail.sysu.edu.cn (G. Li), yizhouy@acm.org (Y. Yu).

https://doi.org/10.1016/j.patcog.2023.110023
Received 6 June 2023; Received in revised form 18 August 2023; Accepted 28 September 2023
Available online 5 October 2023

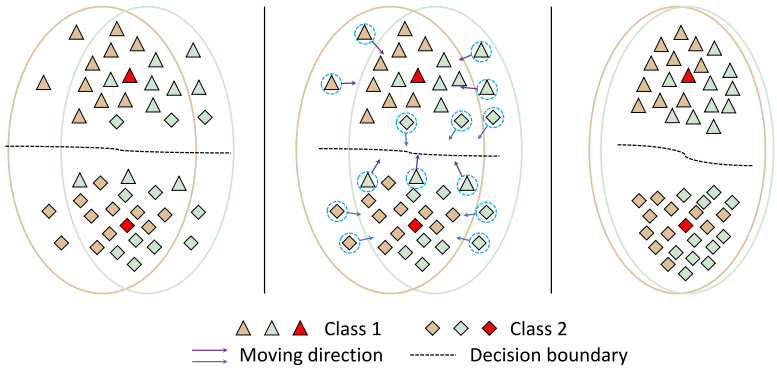



**Fig. 1.** A conceptual description of our basic idea. Sample points in brown, green, and red represent source domain data, target domain data, and class prototypes, respectively. Arrows in purple indicate that sample points move towards the prototype of Class 1, while blue arrows illustrate that the prototype of Class 2 attracts samples from the corresponding class towards itself. **Left:** Previous label-free strategies to enforce domain-level feature alignment fail to generate discriminative target features, thereby giving rise to cross-domain label mismatch in label space. **Middle:** Our approach incorporates label information into adaptation, and thus, the model can align class-wise sample features from both domains with the aid of their class labels. **Right:** The proposed approach enables the model to produce domain-invariant and discriminative features and thus enhance the performance of the model.

regularization is not necessarily beneficial to label mismatch rectification, but on the contrary, may make it worse. For example, entropy minimization in [5], or self-training in [6], seeks to learn knowledge from the model predictions themselves, which is risky due to label noise accumulation [14] and poor model calibration [15]. Hence, recent advances, such as [8,16], have proposed label-aware alignment strategies for cross-domain feature distributions. which considers the ingratiation of label information into adaptation, thereby effectively mitigating the aforementioned issues. However, these alignment strategies overlook the potential to delve deeper into the provided labeled source and target domains to excavate more definitive and authentic supervised label information. Such exploration of a broader cross-domain search scope could better bridge distribution discrepancies between domains, thereby encouraging the model to generate domain-invariant yet discriminative features on the target domain.

This paper proposes a novel label-aware cross-domain feature alignment strategy, namely Inter-domain Mixup, where label information is incorporated into model adaptation. Specifically, Inter-domain Mixup conducts sample-level data mixing and manifold-level data mixing between paired labeled samples from the source and target domains. In detail, sample-level data mixing directly mixes the source images and target images and their corresponding labels through linear combinations of existing labeled samples; on the other hand, manifold-level data mixing creates virtual samples with convex combinations of the features (outputs at the penultimate layer of the model) along with their labels from the source-target sample pairs from labeled data. Thus, many virtual training samples with reliable label information are created. These newly established samples along with their labels connect both domains and are both diverse and complementary to both domains. We, hence, perform supervised learning using these labeled virtual samples so as to mitigate the discrepancy between the source and target domains. As a result, the generated features are agnostic to the distribution discrepancies between the two domains, and meanwhile, exhibit class discriminability.

As introduced above, Inter-domain Mixup makes full use of label information from the labeled samples to enforce label-aware feature alignment across domains. However, there is only a very small amount of labeled data in the target domain, which seriously hinders the potential of cross-domain fusion because the diversity of samples in the target domain is not reflected in data mixing. As the model's generalization ability continues to improve during the training process, for unlabeled samples highly correlated to labeled ones, their predicted labels may become increasingly reliable. Like label propagation for semi-supervised learning [17], to better leverage unlabeled samples in the target domain, we introduce Neighborhood Expansion to transfer existing label information to unlabeled samples. Specifically, we perform label propagation via pseudo labeling to progressively produce pseudo class labels for unlabeled target domain samples that have high-confidence model predictions for the corresponding classes. In the meantime, two schemes, i.e., Self-Regularization and Pairwise Approaching, are presented to reduce the uncertainty of model predictions at unlabeled samples in the target domain, which benefits the model in producing low-entropy and high-confidence predictions.

We call this SSDA framework **I**nter-**d**omain **M**ixup with **N**eighborhood **E**xpansion (**IDMNE**). In a nutshell, the main contributions of our proposed method can be summarized as follows:

- Inter-domain Mixup, a novel cross-domain feature alignment strategy, is proposed to conduct sample-level and manifold-level data mixing for source-target sample pairs from labeled data, which not only facilitates cross-domain feature alignment but also alleviates label mismatch simultaneously to address cross-domain distribution discrepancies and thus achieve a considerable performance gain for model adaptation.
- Neighborhood Expansion is proposed to leverage massive high-confidence pseudo-labeled target domain samples to diversify the label information of the target domain. Moreover, Neighborhood Expansion includes Self-Regularization and Pairwise Approaching, which reduce the uncertainty of model predictions at unlabeled target domain samples in order to make them more confident.
- Inter-domain Mixup and Neighborhood Expansion are integrated into an adaptation framework, and numerous experiments show our proposed method can outperform existing state-of-the-art approaches and achieve considerable accuracy improvement on three commonly used benchmark datasets such as `DomainNet` [18], `Office-Home` [19] and `Office-31` [20].

## 2. Related work

### 2.1. Domain adaptation

Deep neural networks (DNNs) do well in learning discriminative representations for input data by resorting to a considerable amount of labeled data, which is extremely expensive and time-consuming to obtain. Recently, a vast number of domain adaptation techniques [21–24] attempted to design good adaptation models in order to train with the source domain data with rich labels and the target domain data with scarce labels to realize the accurate recognition of the target domain data.

In general, DA requires addressing huge domain shifts by learning to align cross-domain feature representations. Mainstream DA methods consider learning domain-invariant knowledge by constructing various statistical measures to represent domain discrepancy and then decreasing it, such as Maximum Mean Discrepancy (MMD) [25] and its modified versions [26–28]. For instance, Long et al. [26] introduced a deep adaptation network to minimize MMD over multiple domain-specific feature representation layers so as to learn more transferable features. Also, JMMD proposed in [27], inherited from conventional MMD, was employed to enforce cross-domain joint distribution alignment on domain-specific layers, while CMD used in [28] performed order-wise matching in higher-order feature distributions.

Adversarial training, which is employed to confuse the discriminator in a two-player min–max manner to learn cross-domain feature alignment, is another effective way for aligning feature distributions across domains [4,6,29,30]. In particular, Ganin et al. in [4] introduced a gradient reversal layer into the adaptation framework so that a classifier and a discriminator can be responsible for two tasks, namely, object classification and domain classification. The former trains the classifier over source domain data while the latter learned domain-invariant feature representations across domains by fooling the discriminator. Also, Tang et al. [29] proposed discriminative adversarial learning to carry out domain alignment at both the feature level and category level.

Furthermore, several recent work [31,32] also consider image style transfer across domains to bridge visual domain differences as DNN is sensitive to the style of image inputs as pointed out in [33]. For instance, Kim et al. [31] introduced a style transfer algorithm to stylize the source domain to adapt the target domain so as to diversify the texture of synthetic images. Also, Adversarial Style Mining proposed in [32] explored complex styles for an unseen target domain to enhance the adaptation performance in a data-scarce scenario. What's even more, other studies [34,35] have focused on addressing negative transfer to improve the adaptation capability of models for the target domain. Specifically, Lu et al. in [34] proposed weighted correlation embedding learning to specifically address what should be transferred for a given task, thereby avoiding negative transfer caused by distribution outliers. In addition, Lu et al. in [35] introduced guided discrimination and correlation subspace learning for cross-domain image classification, which accounts for domain-invariant, category-discriminative, and correlation-based learning of data. Last but not least, Lu et al. in [36] also put forward a method called cross-domain structure learning (CDSL) for recognizing visual data in the target domain. CDSL incorporates global distribution alignment and local discriminative structure preservation to extract common underlying features between domains.

In practice, domain adaptation has been developed for many new settings in different application scenarios, such as heterogeneous domain adaptation [37], open set domain adaptation [38], partial domain adaptation [39], etc. In this paper, we focus on the tasks of domain adaptation related to semi-supervised learning due to its potential superiority over the commonly defined DA setting, i.e., unsupervised domain adaptation.

### 2.2. Semi-supervised domain adaptation

Tremendous progress has been made in cross-domain feature alignment by recent DA approaches [5,6,40,41] in order to reduce distribution discrepancies between both domains to some extent. However, commonly used DA techniques, such as decreasing cross-domain statistical discrepancy measures, domain adversarial learning, and image style transfer, etc., can only ensure that the model produces domain-invariant features, but as label information is not incorporated in constraining model training, the discriminability of generated features cannot be guaranteed, thus is of considerable significance in matching class-wise distributions [42]. Therefore, several recent DA methods [12,30] proposed to impose constraints on the target domain to take into consideration the class-aware information so as to alleviate label mismatch across domains, e.g., assigning pseudo-labels for unlabeled samples in the target domain.

In contrast to the above practice, Saito et al. in [5] directly assumed that the target domain has a small fraction of ground-truth labels (typically one-shot or three-shot per class), which achieves significant performance gains through extra supervision on the target domain. However, Saito et al. in [5] also demonstrated that a direct application of traditional UDA techniques to the SSDA problem does harm its effectiveness. Therefore, they proposed Minimax Entropy which optimizes in an adversarial scheme to minimize the distance between the class prototypes and neighboring unlabeled target domain samples so as to achieve domain alignment. This relied heavily on a few target labels to establish correlations among samples from both domains. Similar to [5], Qin et al. [9] also introduced an adversarial learning based method called Contradictory Structure Learning to enforce a target-classifier and a source-classifier so as to learn well-clustered target features and well-scattered source features, respectively. In this case, this method could better align the target distribution with the source distribution. Li et al. [6] proposed a cross-domain adaptive clustering algorithm to achieve cluster-wise feature alignment across domains, in which adversarial learning was also adopted. In addition, APE proposed in [7] attempted to adopt an attraction scheme to align global feature distributions across domains and then take on a perturbation scheme and an exploration scheme to optimize intra-domain discrepancy in the target domain. Compared to the above feature-level adaptation, Luo et al. in [8] designed a Relaxed-cGAN to produce new image-label pairs with class-wise conditional semantic information for pixel-level adaptation through image transfer, where the information of source images is ignored.

To sum up, despite the success of previous label-free cross-domain feature alignment strategies to address the SSDA tasks, most of them focus on aligning features at the domain level and ignore the label attribution of samples. In this case, the generated features cannot be aligned according to their class labels, perhaps giving rise to label mismatch in the label space. Hence, imposing crafted constraints on the target domain is crucial for these strategies. Instead, our proposed Inter-domain Mixup takes advantage of the label information of labeled samples to learn class-wise feature alignment across domains by incorporating them into adapting the model. This ensures samples of diverse classes are aligned correctly, especially on the target domain.

### 2.3. Data Mixup

Data Mixup [43] refers to performing convex interpolation on a pair of labeled samples to generate augmented samples for model training. It is designed as a regularizer and augments the smoothness of learned features for supervised learning. Verma et al. [44] extended it to produce more continuous hidden representations during training. Recently, Mixup has confirmed its effectiveness in the field of semi-supervised learning (SSL) [45,46]. For example, Mixmatch augmented labeled and unlabeled samples for training to smooth the model's manifolds [45]. Also, Berthelot et al. [46] enhanced Mixmatch and then proposed ReMixMatch for training the SSL model to achieve a better classification performance. As well, Wang et al. [47] used Mixup to deal with the problem of semi-supervised 3D Medical Image Detection and obtained a substantial improvement in performance by mixing medical images at the image level and object level. Afterwards, several studies [48,49] found the potential of applying Mixup in domain adaptation. For instance, Wu et al. [48] proposed two mixup regularizers at the category and domain levels to instruct the classifier to enforce the consistency of in-between sample predictions and to enrich feature-space intrinsic structures. Also, Na et al. presented in [49] augmented diverse intermediate domains between source-target sample pairs using a fixed ratio-based mixup regularizer, which successfully bridged domain spaces so as to alleviate domain discrepancy.

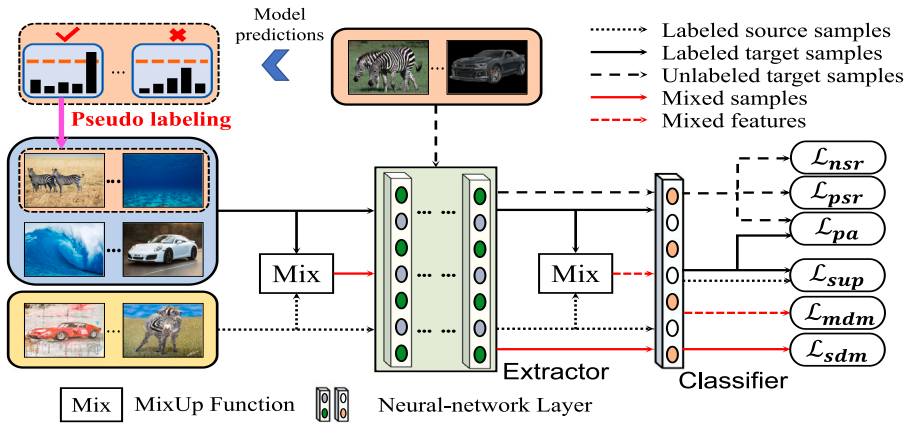



**Fig. 2.** An overview of our proposed Inter-domain Mixup with Neighborhood Expansion for semi-supervised domain adaptation. We use arrows with different line styles to represent data flow, where the black arrow denotes labeled source or target domain data, and the red arrow indicates mixed sample or mixed feature. Our model includes an extractor for feature generation and a classifier for object classification. Also, we train our model with six loss terms in which $\mathcal{L}_{sup}$ is for supervision over labeled data from both domains; $\mathcal{L}_{sdm}$ and $\mathcal{L}_{mdm}$ are for Inter-domain Mixup to perform cross-domain class-wise feature alignment; and the remaining $\mathcal{L}_{psr}$, $\mathcal{L}_{nsr}$ and $\mathcal{L}_{pa}$ are for Neighborhood Expansion to make unlabeled target domain data more confident. To further leverage unlabeled samples in the target domain, we also employ pseudo labeling to assign pseudo-labels to unlabeled target domain samples with high probability scores and merge the selected pseudo-labeled target domain samples into the labeled target domain set.

In this paper, we use Data Mixup to establish Inter-domain Mixup, a novel strategy to perform cross-domain feature alignment for the SSDA task. The most similar work to our method may be [50–52], but there are obvious differences. Firstly, our model behaves linearly within source-target labeled sample pairs whose labels are available randomly. Instead, [50] requires pairing each target class with a unique and dedicated source class, i.e., one-to-one pairing from the target to source classes. In addition, the limitation by the absence of target labels makes these works namely [50–52] an unsupervised Mixup approach that lacks actual but diversified supervised label information. This unsupervised scheme might inadvertently introduce noisy label information, leading to more serious label mismatch during domain alignment. Moreover, the features generated in [50–52] lack associated real label information, thus limiting the classifier's discriminative efficacy. In contrast, the actual but virtual sample-label pairs constructed by our proposed method are fed into the classifier for training, thereby enhancing the classifier's discriminatory capacity. Also, [51] has been proposed to effectively address the task of semantic segmentation, whereas our approach is focused on enhancing the performance of image classification. Most importantly, [50,51] only enforce data mixing at the sample level but ours conducts convex interpolation at both sample and manifold levels. Our proposed method encourages the model to explore more cross-domain searching ranges so as to better bridge distribution discrepancies between domains. Although [52] takes feature-level Mixup into account, pursues the approximation of mixed features to mixup input features through the MSE loss. This scheme yields a narrower cross-domain search scope, thereby compromising the effective bridging of distribution discrepancies across domains.

## 3. The proposed method

In this section, we first present the problem formulation and notations in the context of semi-supervised domain adaptation (SSDA), and then elaborate on our proposed method, i.e., Inter-domain Mixup with Neighborhood Expansion (IDMNE). An overview of the proposed method is illustrated in Fig. 2.

### 3.1. Problem formulation and notation

In the context of SSDA, we are given two sets of labeled samples in the source and target domains respectively, and a set of unlabeled samples in the target domain. They can be denoted as $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$, $D_l = \{(x_i^l, y_i^l)\}_{i=1}^{N_l}$ and $D_u = \{(x_i^u,)\}_{i=1}^{N_u}$, respectively, where the size of $D_l$, i.e., $N_l$, is much smaller than $N_s$ and $N_u$. Specifically, each sample $x_i^s$ ($x_i^l$) in the labeled sample set $D_s$ ($D_l$) is accompanied with a given label indexed by $y_i^s$ ($y_i^l$). However, for an unlabeled target domain sample $x_i^u$ from $D_u$, we have no access to its associated label. Provided with the aforementioned information, the goal of this work is to learn an adaptation model that enables accurate classification of target domain samples during the testing phase.

Recent SSDA work [5–8,13] has proved that a prototypical classifier is beneficial to feature alignment across domains. Following such studies, we also construct a model with parameters $\theta$, which consists of a feature extractor $\mathcal{F}$ and a prototypical classifier $\mathcal{G}$. The extractor is a deep neural network followed by an $\ell_2$ normalization layer, while the classifier comprises an unbiased linear layer. According to [5–7], the weights of such a prototypical classifier can be represented as class prototypes, i.e., $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_k, \dots, \mathbf{w}_K]$, where $k = 1, 2, \dots, K$ indicates the class index. In this case, a normalized feature $\frac{\mathcal{F}(x)}{\|\mathcal{F}(x)\|}$ of a data point $x$ is fed into the classifier, meaning that the feature has been mapped into a spherical feature space [5,7]. Then, the distances between this point and the class prototypes $\{\mathbf{w}_k\}_{k=1}^K$ can be represented as the probabilistic prediction outputs of the classifier, i.e.,

$$\mathbf{p} = p(x) = \sigma(\mathcal{G}(\mathcal{F}(x))) = \sigma(\frac{1}{T}\frac{\mathbf{W}^\top \mathcal{F}(x)}{\|\mathcal{F}(x)\|}), \tag{1}$$

where $\sigma(\cdot)$ represents a softmax function and $T$ indicates a temperature parameter. Larger $p_k(x)$, where $p_k(x)$ is the $k$th component of $p(x)$, means a higher correlation between the data point $x$ and the prototype corresponding to Class $k$, i.e., $\mathbf{w}_k$. To build a good model for adaptation, we have to minimize the distances between the class prototypes and corresponding samples from both source and target domains, indirectly strengthening correlations between both domains and thus achieving cross-domain feature alignment.

In this paper, we perform supervised model training over all labeled samples across domains to address cross-domain distribution discrepancies using a standard cross-entropy loss as follows,

$$\mathcal{L}_{sup}(\theta; D_s, D_l) = -\frac{1}{N_s + N_l} \sum_{(x_i, y_i) \in D_s \cup D_l} p_y(y_i) \log(p(x_i)), \tag{2}$$

where $p_y(\cdot)$ is a function to create a one-hot label probability vector corresponding to the index of a class label.

### 3.2. Inter-domain Mixup

Similar to other DA tasks [5–7], we need to consider data from the source and target domains in the context of probability distributions $P_s$

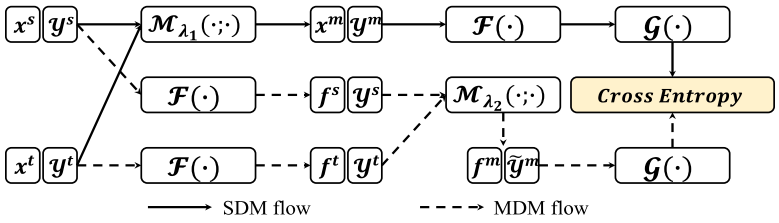



**Fig. 3.** A flow diagram of Inter-domain Mixup. A solid line represents a flow of sample-level data mixing (SDM), while a dashed line indicates a flow of manifold-level data mixing (MDM). $\mathcal{M}_{\lambda_1}(\cdot;\cdot)$ and $\mathcal{M}_{\lambda_2}(\cdot;\cdot)$ are two mixup functions where $\lambda_1$ and $\lambda_2$ are two different mixup ratios. For the SDM flow, we obtain a mixup sample $(x^m, y^m)$ by mixing a source-target sample pair containing a labeled source domain sample $(x^s, y^s)$ and a labeled target domain sample $(x^t, y^t)$ through a linearly convex interpolation. For the MDM flow, two feature representations $f^s$ and $f^t$ along with their original labels $y^s$ and $y^t$ are mixed to generate an augmented feature $f^m$ and its associated label $\tilde{y}^m$. Afterwards, extra supervision over these two types of mixup points, i.e., $(x^m, y^m)$ and $(f^m, \tilde{y}^m)$, is performed via a standard cross-entropy loss function.

and $\mathcal{P}_t$, where $\mathcal{P}_s \neq \mathcal{P}_t$. Therefore, our adaptation model needs to reduce domain shifts in order to align feature distributions across domains.

As partial label information is available in the target domain, we propose to make full use of the label information in the target domain to achieve domain adaptation. Specifically, we propose a novel cross-domain feature alignment strategy called Inter-Domain Mixup, where source domain samples and target domain samples, as well as their affiliated labels, are integrated into the feature alignment process with Mixup operations. It acts as data augmentation and can produce additional reliably labeled samples for model training. Also, in the inter-domain mixup operation, each generated sample can be regarded as a bridge between the source and target domains, playing an indispensable role in closing the gaps between the two domains. Hence, extra supervision from these intermediate and complementary data not only prevents the model from overfitting the source domain dataset but also improves feature discriminability. Our proposed Inter-Domain Mixup conducts sample-level data mixing (**SDM**) and manifold-level data mixing (**MDM**) on source-target labeled sample pairs. A flow diagram of Inter-domain Mixup is given in Fig. 3.

**SDM** is achieved with vanilla Mixup [43] to produce mixed image $x^m$ and its corresponding mixed label $y^m$ through a convex interpolation of a source-target sample pair as follows:

$$\begin{aligned} x^m &= \mathcal{M}_{\lambda_1}(x^s, x^t) = \lambda_1 x^s + (1-\lambda_1)x^t, \\ y^m &= \mathcal{M}_{\lambda_1}(y^s, y^t) = \lambda_1 p_y(y^s) + (1-\lambda_1)p_y(y^t), \end{aligned} \tag{3}$$

where $\lambda_1 \sim Beta(\alpha)$ is a mixup ratio and $\alpha$ is a constant scalar of the beta distribution. As well, $(x^s, y^s)$ and $(x^t, y^t)$ are labeled samples from $D_s$ and $D_l$ respectively. Note that $y^m$ here is not a class index but a label probability vector.

Meanwhile, $(x^s, y^s)$ and $(x^t, y^t)$ are fed into the feature extractor and the feature representations $f^s = \mathcal{F}(x^s)$ and $f^t = \mathcal{F}(x^t)$ are obtained. According to Manifold Mixup used in [44], we carry out **MDM** that mixes the source feature $f^s$ and the target feature $f^t$ in the latent feature space by means of a convex combination:

$$\begin{aligned} f^m &= \mathcal{M}_{\lambda_2}(f^s, f^t) = \lambda_2 f^s + (1-\lambda_2)f^t, \\ \tilde{y}^m &= \mathcal{M}_{\lambda_2}(y^s, y^t) = \lambda_2 p_y(y^s) + (1-\lambda_2)p_y(y^t), \end{aligned} \tag{4}$$

where $\lambda_2 \sim Beta(\alpha)$ is another mixup ratio.

Compared with existing labeled samples from both domains, the mixed samples or features associated with mixed labels obtained by SDM and MDM enrich label information and create a denser data distribution in-between the two domains. Finally, we impose extra supervision on the mixed samples using the following loss,

$$\mathcal{L}_{IDM}(\theta; D_s, D_l) = \mathcal{L}_{sdm}(\theta; D_s, D_l) + \mathcal{L}_{mdm}(\theta; D_s, D_l), \tag{5}$$

$$\mathcal{L}_{sdm}(\theta; D_s, D_l) = -\frac{1}{N_{pair}} \sum_{i=1}^{N_{pair}} y_i^m \log\left(p(x_i^m)\right), \tag{6}$$

$$\mathcal{L}_{mdm}(\theta; D_s, D_l) = -\frac{1}{N_{pair}} \sum_{i=1}^{N_{pair}} \tilde{y}_i^m \log\left(\sigma(\mathcal{G}(f_i^m))\right), \tag{7}$$

where $N_{pair}$ indicates the number of source-target sample pairs constructed by one-to-one pairing from labeled samples in $D_s$ and $D_l$.

As discussed in Section 4.4, our mixup regularizer improves model calibration, thereby increasing the accuracy of pseudo-labels of unlabeled target domain samples provided by Neighborhood Expansion. In addition, Mixup also has the effect of denoising, attenuating the negative impact of incorrect pseudo-labels during training. In general, we generate augmented samples by mixing labeled source domain samples with originally labeled target domain samples or pseudo-labeled target domain samples. As a result, Inter-domain Mixup ensures that at least a portion of the label information for such an augmented sample is reliable.

### 3.3. Neighborhood expansion

Inter-domain Mixup benefits the model by creating a large number of valuable labeled samples. However, only a limited portion of target labels are available, i.e., one-shot or three-shot per class, which significantly restrains the diversity of mixup samples and the representativeness of the dataset. Several recent work [6,7,13] have proved that pseudo labeling is very helpful to remedy this issue. Like such work, we further propose Neighborhood Expansion to make full use of the unlabeled samples in the target domain. Concretely, by pseudo labeling, we assign a pseudo-label with the highest predicted probability to each unlabeled sample in the target domain at the beginning of each training epoch, as long as its confidence is greater than a certain threshold. A large number of pseudo-labeled samples in the target domain provides more diversified representation learning information to improve the generalization ability of the classifier within the target domain.

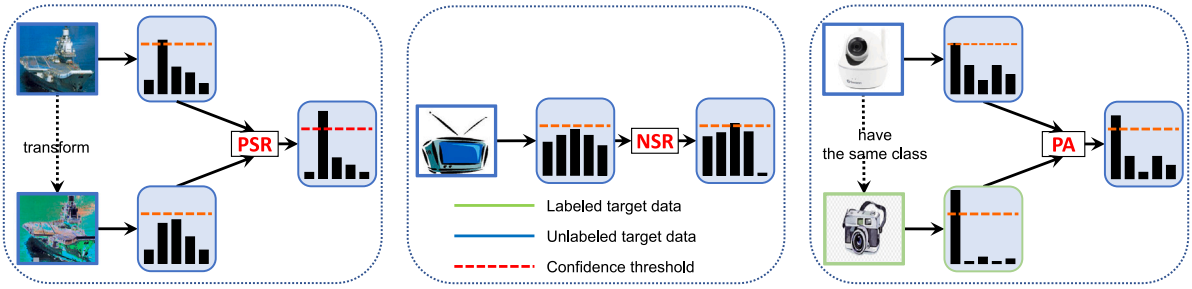
Technically, we pre-define a confidence threshold $\tau$, and an unlabeled target domain sample is assigned a pseudo-label $\hat{y}_i^u = \text{argmax}\,(p(x_i^u))$ when the probability (confidence) score of its predicted class label is larger than $\tau$. Therefore, we select pseudo-labeled target domain samples from $D_u$, merge them into the labeled target domain set $D_l$, and finally form a new target domain set $D_l'$. Here, it is essential to observe that these pseudo-labeled target samples are not excluded from $D_u$. Furthermore, $D_l'$ is used not only in equations of this section, but also in Eq. (6) and Eq. (7) of Section 3.2.

To make model predictions at unlabeled target domain samples more confident, we introduce **Self-Regularization** and **Pairwise Approaching** to reduce the uncertainty of the predictions, achieving the goal of this by minimizing the following loss,

$$\mathcal{L}_{NE}(\theta; D_l', D_u) = \mathcal{L}_{psr}(\theta; D_u) + \mathcal{L}_{nsr}(\theta; D_u) + \mathcal{L}_{pa}(\theta; D_u, D_l'). \tag{8}$$

#### 3.3.1. Self-Regularization

Self-Regularization allows the network to learn knowledge from a sample itself. In our proposed method, Self-Regularization includes positive self-regularization learning (**PSR**) and negative self-regularization learning (**NSR**). We employ PSR and NSR to handle unlabeled target domain samples whose predicted probability scores are above and below the confidence threshold $\tau$ respectively. Specifically, different from




(a) PSR (b) NSR (c) PA

**Fig. 4.** Illustrations of three schemes applied in Neighborhood Expansion to encourage low-entropy and high-confidence predictions for unlabeled target domain samples. (a) Positive self-regularization learning (PSR) introduces self-training to augment the model's robustness. (b) Negative self-regularization learning (NSR) is to raise the predictive probabilities of each class except for the class corresponding to the lowest predicted probability scores. (c) Pairwise Approaching (PA) aims to drive high-confidence unlabeled target domain samples towards labeled data of the same class in the target domain. Note that PSR and PA handle samples from $\mathcal{D}_u$ whose confidence scores of their predicted class labels are above the confidence threshold $\tau$, while NSR is of importance for unlabeled target domain samples with confidence scores lower than $\tau$.

traditional self-training techniques [53–56], our **PSR** only operates on unlabeled target domain samples that achieve high predicted probabilities since they are more likely to be assigned with correct pseudo-labels. In addition, similar to [57], the proposed PSR expects an augmented image to have the same model output as its original version. We enforce positive self-regularization learning with a supervised cross-entropy loss as follows,

$$\mathcal{L}_{psr}(\theta;\mathcal{D}_u) = -\frac{1}{\sum_{x_i^u\in\mathcal{U}}\mu_i}\sum_{x_i^u\in\mathcal{U}}\mu_i\cdot p_y(\hat{y}_i^u)\log(p(x_i^u+\delta)), \tag{9}$$

where $\hat{y}_i^u = \text{argmax}\,(p(x_i^u))$ indicates the pseudo-label of a sample $x_i^u$, and $p(x_i^u)$ represents the prediction at an unlabeled image $x_i^u$. Also, $p(x_i^u+\delta)$ represents the prediction at a transformed image $x_i^u+\delta$, where we use RandAugment [58] to add a perturbation $\delta$ to the original image $x_i^u$. Moreover, $\mu_i = \mathbb{1}\{\max(p(x_i^u)) \geq \tau\}$ indicates the probabilistic confidence score of the predicted label of a sample $x_i^u$ should be larger than $\tau$, and $\mathbb{1}\{\cdot\}$ is an indicator function. As illustrated in Fig. 4(a), PSR can strengthen the robustness of the model so as to implicitly achieve higher confidence at an unlabeled sample in the target domain.

For unlabeled target domain samples whose probability (confidence) scores associated with their predicted class labels are below the confidence threshold $\tau$, assigning them hard pseudo-labels could easily confuse the model with incorrect label information. As training proceeds, the classifier would gradually fit the assigned noisy pseudo-labels, giving rise to performance degradation. Therefore, as like [59], we here intend to employ **NSR** to assign a hard "complementary" label to an unlabeled sample with a low confidence score in the target domain. Concretely, a complementary label is determined by the index of the minimum component of the sample's predicted probability vector. In this case, the complementary label means that the sample has the maximum probability of not belonging to the corresponding class. To this end, we optimize the model via an NSR loss, i.e., $\mathcal{L}_{nsr}$, to make the probabilities of the predicted complementary class labels of unlabeled target domain samples farther away from 1 but closer to 0 as follows:

$$\mathcal{L}_{nsr}(\theta;\mathcal{D}_u) = -\frac{1}{\sum_{x_i^u\in\mathcal{U}}\mu_i'}\sum_{x_i^u\in\mathcal{U}}\mu_i'\cdot p_y(\bar{y}_i^u)\log(1-p(x_i^u)), \tag{10}$$

where $\bar{y}_i^u = \text{argmax}\,(1-p(x_i^u))$ and $\mu_i' = \mathbb{1}\{\max(p(x_i^u)) < \tau\}$ indicates the highest predicted probability score of $x_i^u$ is less than $\tau$. Note that NSR differs from [59] since a complementary label in [59] is selected from all class labels at random except for the predicted class label corresponding to the highest confidence score of a sample. As shown in Fig. 4(b), after model optimization with the NSR loss, the probability score of the predicted complementary class label of a sample has been almost decreased to 0, but the predicted probabilities of other classes are increased.

**Algorithm 1:** Pseudo-code of IDMNE.

**Input:** Labeled samples $\mathcal{D}_s$, labeled samples $\mathcal{D}_l$, unlabeled samples $\mathcal{D}_u$, confidence threshold $\tau$, number of training epochs $\mathcal{T}$
**Output:** Optimal model parameters $\theta$
**for** $epoch = 1, 2, \ldots, \mathcal{T}$ **do**
  **for** $x \in \mathcal{D}_u$ **do**
    $\hat{y} = \text{argmax}\,(p(x))$;
    $\mathcal{D}_l' \leftarrow \mathcal{D}_l' \cup \{(x, \hat{y}) | \max(p(x)) \geq \tau\}$;
  $\mathcal{D}_l' \leftarrow \mathcal{D}_l \cup \mathcal{D}_l'$;
  Randomly sample mini-batches $\mathcal{B}_s \subset \mathcal{D}_s, \mathcal{B}_l \subset \mathcal{D}_l, \mathcal{B}_l' \subset \mathcal{D}_l' and \mathcal{B}_u \subset \mathcal{D}_u$;
  Update model parameters $\theta$ by applying SGD with the overall loss, $\mathcal{L} = \mathcal{L}_{sup}(\theta;\mathcal{B}_s,\mathcal{B}_l) + \beta\mathcal{L}_{IDM}(\theta;\mathcal{B}_s,\mathcal{B}_l') + \gamma\mathcal{L}_{NE}(\theta;\mathcal{B}_l',\mathcal{B}_u)$.

### 3.3.2. Pairwise Approaching

Pairwise Approaching (**PA**) is introduced to make feature representations in the target domain more compact by driving unlabeled samples closer to those labeled ones. During the training process, model predictions at more and more unlabeled target domain samples have a confidence level exceeding the threshold $\tau$. Given this observation, we draw on the idea of contrastive learning [60] and introduce binary cross-entropy as a loss term to make those unlabeled samples with high confidence approach those labeled target domain samples of the same category to learn more compact features. The loss for Pairwise Approaching is formulated as follows:

$$\begin{aligned}\mathcal{L}_{pa}(\theta;\mathcal{D}_u,\mathcal{D}_l') = \\ -\frac{1}{\sum_{x_i^u\in\mathcal{U}}\mu_i}\sum_{x_i^u\in\mathcal{D}_u}\sum_{x_j^l\in\mathcal{D}_l'}\mu_i\cdot[\nu_{ij}\log(\mathbf{p}_i^\top\mathbf{p}_j) \\ +(1-\nu_{ij})\log(1-\mathbf{p}_i^\top\mathbf{p}_j)],\end{aligned} \tag{11}$$

where $\mathbf{p}_i = p(x_i^l)$ and $\mathbf{p}_j = p(x_j^u)$ represent the predictions at a labeled (or pseudo-labeled) target image $x_i^l$ and an unlabeled target image $x_j^u$, respectively. Also, $\nu_{ij} = \mathbb{1}\{\hat{y}_i^u = y_j^l\}$ denotes a pairwise label that indicates whether $x_i^u$ and $x_j^l$ have the same class label. As shown in Fig. 4(c), this loss enables the selected unlabeled target domain data to obtain higher predicted probability scores so that the model produces predictions with lower entropy but higher confidence for them.

### 3.4. Overall loss function

The overall training procedure of our proposed method is described in Algorithm 1. At the beginning of each epoch, we first use the pseudo labeling scheme to assign pseudo-labels to a subset of unlabeled samples in $\mathcal{D}_u$ whose confidence score of the predicted class label is

larger than the threshold $\tau$. Then, we update $\mathcal{D}_l$ to include the pseudo-labeled target domain samples and form $\mathcal{D}'_l$. Next, four mini-batches $\mathcal{B}_s$, $\mathcal{B}_l$, $\mathcal{B}'_l$ and $\mathcal{B}_u$ are assembled by means of random sampling from $\mathcal{D}_s$, $\mathcal{D}_l$, $\mathcal{D}'_l$ and $\mathcal{D}_u$. Finally, the overall loss function for model training can be formulated as follows,

$$\mathcal{L} = \mathcal{L}_{sup}(\theta; \mathcal{B}_s, \mathcal{B}_l) + \beta \mathcal{L}_{IDM}(\theta; \mathcal{B}_s, \mathcal{B}'_l) + \gamma \mathcal{L}_{NE}(\theta; \mathcal{B}'_l, \mathcal{B}_u), \tag{12}$$

where $\beta$ and $\gamma$ are two hyper-parameters that trade-offs the losses. In all experiments, we set $\beta = 1.0$ and $\gamma = 0.1$, respectively. In order to mitigate the influence exerted by under-confident unlabeled samples on the model, we have discovered that configuring a smaller $\gamma$ proves advantageous for the acquisition of a more optimal adaptation model.

## 4. Experiments

To demonstrate the advantages of the proposed IDMNE method, we have conducted extensive experiments on multiple widely-used benchmarks, namely `DomainNet` [18], `Office-Home` [19] and `Office-31` [20]. We first briefly introduce the experimental setups, such as the details of evaluation datasets and the corresponding evaluation protocols. Then, we perform comprehensive comparisons to verify the superiority of our IDMNE over existing state-of-the-art methods. Finally, we have performed detailed ablation studies to demonstrate the contribution of each component within IDMNE. Note that we implemented the proposed method using PyTorch,[1] a popular platform for deep learning, and run all experiments on an NVIDIA GeForce 1080Ti GTX GPU.

### 4.1. Datasets

We evaluate the proposed approach on three widely used SSDA benchmark datasets: `DomainNet` [18], `Office-Home` [19] and `Office-31` [20]. Following prior algorithms, the number of labeled target domain samples is set to 1 shot or 3 shots per class.

`DomainNet` is a standard benchmark dataset for multi-source domain adaptation with a large scale of 0.6 million images. `DomainNet` has 6 domains and each domain consists of 345 categories. Following [5], we adopt only 4 domains, namely Real: R, Clipart: C, Painting: P, and Sketch: S, and 126 categories to validate the proposed framework for adaptation. As in [5], we also choose seven adaptation scenarios for performance comparisons.

`Office-Home` is another popular SSDA benchmark dataset used to evaluate the proposed method with several challenging adaptation scenarios. This dataset involves approximately 65 classes. There are 4 domains, including Real: R, Clipart: C, Art: A, and Product: P. To be fair, we follow previous SSDA work [5–7] to carry out 12 adaptation scenarios on this dataset.

`Office-31` is a small-scale benchmark dataset used for SSDA evaluation. This dataset has three domains, including DSLR: W, Webcam: W, and Amazon: A, and 31 classes.

### 4.2. Experimental protocols

Following [5,6,13,61], we adopt AlexNet [62] and ResNet-34 [63] as network backbones for evaluation on `DomainNet` while using AlexNet, VGG-16 [64] and ResNet-34 on `Office-Home`. We only use AlexNet on `Office-31`. To achieve a fair comparison, we first initialize the networks with pre-trained weights from ImageNet [65] and replace the last layer with a prototypical classifier. This prototypical classifier has an unbiased linear layer, which is initialized using random parameters and a temperature $T = 0.05$. We perform model training using Stochastic Gradient Descent (SGD) with a momentum of 0.9 and a weight decay of $5 \times 10^{-4}$. Moreover, the initial learning rate is set to $\eta_0 = 0.001$ and is updated by following the rule in [5] as model training iterates, i.e., $\eta_t = \frac{\eta_0}{(1+0.0001\times t)^{0.75}}$, where $\eta_t$ denotes the learning rate at $t$th iteration.

At the start of each iteration, we randomly sample four mini-batches $\mathcal{B}_s \subset \mathcal{D}_s$, $\mathcal{B}_l \subset \mathcal{D}_l$, $\mathcal{B}'_l \subset \mathcal{D}'_l$ and $\mathcal{B}_u \subset \mathcal{D}_u$, where $\mathcal{D}'_l$ contains labeled samples from $\mathcal{D}_l$ as well as pseudo-labeled samples obtained from $\mathcal{D}_u$. The mini-batch size of $\mathcal{B}_s$, $\mathcal{B}_l$, $\mathcal{B}'_l$ and $\mathcal{B}_u$ are set to 24, 24, 24 and 48, respectively when ResNet-34 or VGG-16 is used (32, 32, 32 and 64 for AlexNet). Before being fed into the network, the size of the input images is first resized to $256 \times 256$, then the images are augmented using the random horizontal flip and random crop ($224 \times 224$ for VGG-16 and ResNet-34, and $227 \times 227$ for AlexNet). We finally subtract the per-pixel image mean of the dataset from all images. In addition, the model is trained with 150 epochs (7500 iterations) on `DomainNet`, 100 epochs (2500 iterations) on `Office – Home`, and 100 epochs (1500 iterations) on `Office – 31`, i.e., $\mathcal{T} = 150$ for `DomainNet` and $\mathcal{T} = 100$ for `Office-Home` and `Office-31`. As well, $\tau$ and $\alpha$ are set to 0.95 and 2.0, respectively.

**Setting the Hyper-parameters** $\alpha$, $\beta$, $\gamma$, and $\tau$. We set the value of $\alpha$ to 1.0, which is commonly used in Mixup operations [43,44]. For $\tau$, we decided on a value of 0.95, following the common practice in pseudo-labeling techniques [57]. To determine suitable values for $\beta$ and $\gamma$, we conducted hyper-parameter selection using the "$R \rightarrow S$" case on the `DomainNet` dataset, using ResNet-34 and the 3-shot setup. We then applied these chosen settings to other benchmark datasets and adaptation scenarios, ensuring the reproducibility and simplicity of our proposed algorithm. For the details of choosing of $\beta$ and $\gamma$, similarly to MME [5], we first selected three labeled examples as the validation set for the target domain. We utilized these validation examples to choose those hyper-parameters through a grid search. During this process, we conducted the experiment by fixing the value of $\beta$ to adjust $\gamma$, and then fixing $\gamma$ to adjust $\beta$. Finally, we chose the values of $\beta = 1.0$ and $\gamma = 0.1$ based on the highest validation accuracy achieved.

**Averaged Cluster Centroid Distance.** We propose the averaged Cluster Centroid Distance (ACCD) as a metric, inspired by prior works [6], to evaluate the effectiveness of our Inter-domain Mixup approach in achieving cross-domain feature alignment. ACCD quantifies the distance between the feature clusters of the source and target domains for all classes in the dataset. Specifically, ACCD is calculated as $d^e_{avg} = \textbf{average}(\{d^e_1, d^e_2, \ldots, d^e_k, \ldots, d^e_K\})$, where $\textbf{average}(\cdot)$ computes the average of the given inputs, and $K$ represents the number of classes in the dataset. Here, $d^e_k$ refers to the pairwise Euclidean distances between the centroids of the feature clusters from the source and target domains for class $k$ at epoch $e$, each of which should be normalized by the initial distance $d^0_k$, obtained from the initial model with pre-trained weights on ImageNet without any fine-tuning. In general, a smaller ACCD indicates better feature alignment between the source and target domain clusters.

### 4.3. Comparison with the state of the arts

We perform experimental comparisons between the proposed method and existing state-of-the-art SSDA algorithms, including **S+T** [5], **DANN** [4], **MME** [5], **UODA** [9], **Meta-MME** [66], **BiAT** [67], **APE** [7], **PAC** [68], **ELP** [61], **DECOTA** [13], **Relaxed-cGAN** [8], **CDAC** [6], and **UODAv2** [69]. The results of **S+T** and **DANN** are borrowed from [5]. **S+T** trains the model using a cross-entropy loss over the labeled source and target domain data only. **DANN** is modified from [4], and the model is trained with labeled source domain data, unlabeled target domain data, and a small amount of labeled target domain data. In addition, **MME**, **Meta-MME**, **UODA** and **CDAC** are adversarial learning based approaches while **APE** and **Relaxed-cGAN** mainly focus on minimizing cross-domain discrepancy measures and rely on image style transfer, while **UODAv2** is the extension of **UODA**. Furthermore, to address the SSDA problem, **Meta-MME**, **BiAT**, **PAC**

[1] https://pytorch.org/

**Table 1**
Comparison results (%) on 4 domains of DomainNet under the 3-shot setting using AlexNet. (Mean accuracy and 95% confidence interval over five trails).

| Method | R→C | R→P | P→C | C→S | S→P | R→S | P→R | Mean |
|---|---|---|---|---|---|---|---|---|
| S+T [5] | 47.1 | 45.0 | 44.9 | 36.4 | 38.4 | 33.3 | 58.7 | 43.4 |
| DANN [5] | 46.1 | 43.8 | 41.0 | 36.5 | 38.9 | 33.4 | 57.3 | 42.4 |
| MME [5] | 55.6 | 49.0 | 51.7 | 39.4 | 43.0 | 37.9 | 60.7 | 48.2 |
| Meta-MME [66] | 56.4 | 50.2 | 51.9 | 39.6 | 43.7 | 38.7 | 60.7 | 48.8 |
| BiAT [67] | 58.6 | 50.6 | 52.0 | 41.9 | 42.1 | 42.0 | 58.8 | 49.4 |
| APE [7] | 54.6 | 50.5 | 52.1 | 42.6 | 42.2 | 38.7 | 61.4 | 48.9 |
| PAC [68] | 61.7 | 56.9 | 59.8 | 52.9 | 43.9 | 48.2 | 59.7 | 54.7 |
| Relaxed-cGAN [8] | 56.8 | 51.8 | 52.0 | 44.1 | 44.2 | 42.8 | 61.1 | 50.5 |
| CDAC [6] | 61.4 | 57.5 | 58.9 | 50.7 | 51.7 | 46.7 | 66.8 | 56.2 |
| IDMNE (Ours) | **63.17 ± 0.22** | **58.96 ± 0.30** | **61.48 ± 0.21** | **54.88 ± 0.77** | **53.58 ± 0.42** | **48.52 ± 0.44** | **67.49 ± 0.22** | **58.30** |

**Table 2**
Comparison results (%) on 4 domains of DomainNet under the 1-shot and 3-shot settings using ResNet-34. (Mean accuracy and 95% confidence interval over five trails).

| Method | R→C | | R→P | | P→C | | C→S | | S→P | | R→S | | P→R | | Mean | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1-shot | 3-shot | 1-shot | 3-shot | 1-shot | 3-shot | 1-shot | 3-shot | 1-shot | 3-shot | 1-shot | 3-shot | 1-shot | 3-shot | 1-shot | 3-shot |
| S+T [5] | 55.6 | 60.0 | 60.6 | 62.2 | 56.8 | 59.4 | 50.8 | 55.0 | 56.0 | 59.5 | 46.3 | 50.1 | 71.8 | 73.9 | 56.9 | 60.0 |
| DANN [5] | 58.2 | 59.8 | 61.4 | 62.8 | 56.3 | 59.6 | 52.8 | 55.4 | 57.4 | 59.9 | 52.2 | 54.9 | 70.3 | 72.2 | 58.4 | 60.7 |
| MME [5] | 70.0 | 72.2 | 67.7 | 69.7 | 69.0 | 71.7 | 56.3 | 61.8 | 64.8 | 66.8 | 61.0 | 61.9 | 76.1 | 78.5 | 66.4 | 68.9 |
| UODA [9] | 72.7 | 75.4 | 70.3 | 71.5 | 69.8 | 73.2 | 60.5 | 64.1 | 66.4 | 69.4 | 62.7 | 64.2 | 77.3 | 80.8 | 68.5 | 71.2 |
| BiAT [67] | 73.0 | 74.9 | 68.0 | 68.8 | 71.6 | 74.6 | 57.9 | 61.5 | 63.9 | 67.5 | 58.5 | 62.1 | 77.0 | 78.6 | 67.1 | 69.7 |
| APE [7] | 70.4 | 76.6 | 70.8 | 72.1 | 72.9 | 76.7 | 56.7 | 63.1 | 64.5 | 66.1 | 63.0 | 67.8 | 76.6 | 79.4 | 67.6 | 71.7 |
| PAC [68] | 74.9 | 78.6 | 73.0 | 74.3 | 72.6 | 76.0 | 65.8 | 69.6 | 67.9 | 69.4 | 68.7 | 70.2 | 76.7 | 79.3 | 71.4 | 73.9 |
| ELP [61] | 72.8 | 74.9 | 70.8 | 72.1 | 72.0 | 74.4 | 59.6 | 64.3 | 66.7 | 69.7 | 63.3 | 64.9 | 77.8 | 81.0 | 69.0 | 71.6 |
| DECOTA [13] | 79.1 | 80.4 | 74.9 | 75.2 | 76.9 | 78.7 | 65.1 | 68.6 | 72.0 | 72.7 | 69.7 | 71.9 | 79.6 | 81.5 | 73.9 | 75.6 |
| CDAC [6] | 77.4 | 79.6 | 74.2 | 75.1 | 75.5 | 79.3 | 67.6 | 69.9 | 71.0 | 73.4 | 69.2 | 72.5 | 80.4 | 81.9 | 73.6 | 76.0 |
| UODAv2 [69] | 77.0 | 79.4 | 75.4 | 76.7 | 75.5 | 78.3 | 66.5 | 70.2 | 72.1 | 74.2 | 70.9 | 72.1 | 79.7 | 82.3 | 73.9 | 76.2 |
| IDMNE (Ours) | **79.56 ± 0.21** | **80.81 ± 0.21** | **75.95 ± 0.30** | **76.88 ± 0.11** | **79.43 ± 0.17** | **80.29 ± 0.13** | **71.69 ± 0.73** | **72.22 ± 0.35** | **75.35 ± 0.48** | **75.39 ± 0.12** | **73.48 ± 0.64** | **73.92 ± 0.28** | **82.14 ± 0.67** | **82.80 ± 0.06** | **76.80** | **77.47** |

**Table 3**
Comparison results (%) on 4 domains of Office-Home under the 3-shot setting using AlexNet. (Mean accuracy and 95% confidence interval over five trails).

| Method | R→C | R→P | R→A | P→R | P→C | P→A | A→P | A→C | A→R | C→R | C→A | C→P | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S+T [5] | 44.6 | 66.7 | 47.7 | 57.8 | 44.4 | 36.1 | 57.6 | 38.8 | 57.0 | 54.3 | 37.5 | 57.9 | 50.0 |
| DANN [5] | 47.2 | 66.7 | 46.6 | 58.1 | 44.4 | 36.1 | 57.2 | 39.8 | 56.6 | 54.3 | 38.6 | 57.9 | 50.3 |
| ENT [5] | 44.9 | 70.4 | 47.1 | 60.3 | 41.2 | 34.6 | 60.7 | 37.8 | 60.5 | 58.0 | 31.8 | 63.4 | 50.9 |
| MME [5] | 51.2 | 73.0 | 50.3 | 61.6 | 47.2 | 40.7 | 63.9 | 43.8 | 61.4 | 59.9 | 44.7 | 64.7 | 55.2 |
| Meta-MME [66] | 50.3 | – | – | – | 48.3 | 40.3 | – | 44.5 | – | – | 44.5 | – | – |
| BiAT [67] | – | – | – | – | – | – | – | – | – | – | – | – | 56.4 |
| APE [7] | 51.9 | 74.6 | 51.2 | 61.6 | 47.9 | 42.1 | 65.5 | 44.5 | 60.9 | 58.1 | 44.3 | 64.8 | 55.6 |
| PAC [68] | **58.9** | 72.4 | 47.5 | 61.9 | **53.2** | 39.6 | 63.8 | **49.9** | 60.0 | 54.5 | 36.3 | 64.8 | 55.2 |
| CDAC [6] | 54.9 | 75.8 | 51.8 | 64.3 | 51.3 | **43.6** | 65.1 | 47.5 | **63.1** | 63.0 | **44.9** | 65.6 | 56.8 |
| IDMNE (Ours) | 55.91 ± 0.38 | **76.91 ± 0.56** | **51.98 ± 0.25** | **65.79 ± 0.22** | 53.00 ± 0.72 | 43.48 ± 0.69 | **66.28 ± 0.31** | 47.20 ± 0.78 | 61.10 ± 0.42 | **63.69 ± 0.58** | 42.83 ± 0.16 | **65.97 ± 0.42** | **57.84** |

**Table 4**
Comparison results (%) on 4 domains of Office-Home under the 1-shot and 3-shot settings using VGG-16. (Mean accuracy and 95% confidence interval over five trails).

| Method | R→C | R→P | R→A | P→R | P→C | P→A | A→P | A→C | A→R | C→R | C→A | C→P | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1-shot | | | | | | | | | | | | | |
| S+T [5] | 39.5 | 75.3 | 61.2 | 71.6 | 37.0 | 52.0 | 63.6 | 37.5 | 69.5 | 64.5 | 51.4 | 65.9 | 57.4 |
| DANN [5] | 52.0 | 75.7 | 62.7 | 72.7 | 45.9 | 51.3 | 64.3 | 44.4 | 68.9 | 64.2 | 52.3 | 65.3 | 60.0 |
| ENT [5] | 23.7 | 77.5 | 64.0 | 74.6 | 21.3 | 44.6 | 66.0 | 22.4 | 70.6 | 62.1 | 25.1 | 67.7 | 51.6 |
| MME [5] | 49.1 | 78.7 | 65.1 | 74.4 | 46.2 | 56.0 | 68.6 | 45.8 | 72.2 | 68.0 | 57.5 | 71.3 | 62.7 |
| UODA [9] | 49.6 | 79.8 | 66.1 | 75.4 | 45.5 | 58.8 | 72.5 | 43.3 | **73.3** | 70.5 | 59.3 | 72.1 | 63.9 |
| ELP [61] | 49.2 | 79.7 | 65.5 | 75.3 | 46.7 | 56.3 | 69.0 | 46.1 | 72.4 | 68.2 | **67.4** | 71.6 | 63.1 |
| DECOTA [13] | 47.2 | 80.3 | 64.6 | 75.5 | 47.2 | 56.6 | 71.1 | 42.5 | 73.1 | 71.0 | 57.8 | 72.9 | 63.3 |
| UODAv2 [69] | 51.6 | 80.9 | 66.9 | 75.9 | 49.7 | **60.5** | 71.0 | 44.9 | 73.2 | 70.6 | 58.7 | 72.8 | 64.7 |
| IDMNE (Ours) | **52.61 ± 0.92** | **81.75 ± 0.98** | **67.51 ± 0.23** | **77.27 ± 0.06** | **50.67 ± 0.45** | 59.70 ± 0.74 | **73.71 ± 0.77** | **49.62 ± 0.06** | 72.64 ± 0.41 | **71.42 ± 0.18** | 62.52 ± 0.53 | **76.17 ± 1.70** | **66.30** |
| 3-shot | | | | | | | | | | | | | |
| S+T [5] | 49.6 | 78.6 | 63.6 | 72.7 | 47.2 | 55.9 | 69.4 | 47.5 | 73.4 | 69.7 | 56.2 | 70.4 | 62.9 |
| DANN [5] | 56.1 | 77.9 | 63.7 | 73.6 | 52.4 | 56.3 | 69.5 | 50.0 | 72.3 | 68.7 | 56.4 | 69.8 | 63.9 |
| ENT [5] | 48.3 | 81.6 | 65.5 | 76.6 | 46.8 | 56.9 | 73.0 | 44.8 | 75.3 | 72.9 | 59.1 | 77.0 | 64.8 |
| MME [5] | 56.9 | 82.9 | 65.7 | 76.7 | 53.6 | 59.2 | 75.7 | 54.9 | 75.3 | 72.9 | 61.1 | 76.3 | 67.6 |
| UODA [9] | 57.6 | 83.6 | 67.5 | 77.7 | 54.9 | 61.0 | 77.7 | 55.4 | **76.7** | 73.8 | 61.9 | 78.4 | 68.9 |
| APE [7] | 56.0 | 81.0 | 65.2 | 73.7 | 51.4 | 59.3 | 75.0 | 54.4 | 73.7 | 71.4 | 61.7 | 75.1 | 66.5 |
| ELP [61] | 57.1 | 83.2 | 67.0 | 76.3 | 53.9 | 59.3 | 75.9 | 55.1 | 76.3 | 73.3 | 61.9 | 76.1 | 68.0 |
| DECOTA [13] | 59.9 | 83.9 | 67.7 | 77.3 | 57.7 | 60.7 | 78.0 | 54.9 | 76.0 | 74.3 | 63.2 | 78.4 | 69.3 |
| UODAv2 [69] | 59.3 | 83.6 | 68.0 | **78.3** | 56.8 | 61.8 | **78.6** | 55.7 | 75.3 | 74.0 | 63.3 | 78.9 | 69.5 |
| IDMNE (Ours) | **60.21 ± 0.29** | **84.42 ± 0.59** | **69.33 ± 0.38** | 77.92 ± 0.44 | **59.15 ± 1.06** | **62.63 ± 0.81** | 77.68 ± 1.16 | **58.24 ± 0.15** | 76.68 ± 0.20 | **74.89 ± 0.49** | **64.56 ± 0.41** | **79.27 ± 0.32** | **70.41** |

and **DECOTA** adapt previous techniques, such as meta-learning [70], VAT [71], FixMatch [57], Co-training [72], and so on. Tables 1–6 have listed the mean accuracy and 95% confidence interval over five trials in each experiment of different adaptation scenarios. The results demonstrate that across different datasets using various network backbones under either 1-shot or 3-shot setups, the average performance for all adaptation scenarios achieves higher than all existing state-of-the-art algorithms, illustrating the superiority of the proposed approach in handling the SSDA task. Additionally, it can be observed that, in the majority of individual adaptation cases, the results show that our average accuracy with the lower bound still achieves the best. This shows that the proposed method can obtain statistically significant performance improvement over most of the existing best-performing methods.

**Results on DomainNet.** Comparison results between our method and previous SSDA algorithms on DomainNet are shown in Tables 1–2. On this dataset, we carry out experiments under the 3-shot setting using AlexNet and ResNet-34, and under the 1-shot setting using ResNet-34. The average performance of the proposed method exceeds that of previous algorithms by large margins under all settings. This demonstrates that our method can perform well in diverse adaptation scenarios defined on DomainNet. Specifically, as shown in Table 1, our algorithm improves the mean accuracy achieved by the existing best-performing algorithm, i.e., **CDAC**, by 2.10%, under the 3-shot

**Table 5**
Comparison results (%) on 4 domains of Office-Home under the 3-shot setting using ResNet-34. (Mean accuracy and 95% confidence interval over five trails).

| Method | R→C | R→P | R→A | P→R | P→C | P→A | A→P | A→C | A→R | C→R | C→A | C→P | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S+T [5] | 55.7 | 80.8 | 67.8 | 73.1 | 53.8 | 63.5 | 73.1 | 54.0 | 74.2 | 68.3 | 57.6 | 72.3 | 66.2 |
| DANN [5] | 57.3 | 75.5 | 65.2 | 69.2 | 51.8 | 56.6 | 68.3 | 54.7 | 73.8 | 67.1 | 55.1 | 67.5 | 63.5 |
| ENT [5] | 62.6 | 85.7 | 70.2 | 79.9 | 60.5 | 63.9 | 79.5 | 61.3 | 79.1 | 76.4 | 64.7 | 79.1 | 71.9 |
| MME [5] | 64.6 | 85.5 | 71.3 | 80.1 | 64.6 | 65.5 | 79.0 | 63.6 | 79.7 | 76.6 | 67.2 | 79.3 | 73.1 |
| Meta-MME [66] | 65.2 | – | – | – | 64.5 | 66.7 | – | 63.3 | – | – | 67.5 | – | – |
| APE [7] | 66.4 | 86.2 | 73.4 | 82.0 | 65.2 | 66.1 | 81.1 | 63.9 | 80.2 | 76.8 | 66.6 | 79.9 | 74.0 |
| Relaxed-cGAN [8] | 68.4 | 85.5 | 73.8 | 81.2 | **68.1** | 67.9 | 80.1 | 64.3 | 80.1 | 77.5 | 66.3 | 78.3 | 74.2 |
| DECOTA [13] | 70.4 | 87.7 | 74.0 | 82.1 | 68.0 | **69.9** | 81.8 | 64.0 | 80.5 | 79.0 | 68.0 | **83.2** | 75.7 |
| CDAC [6] | 67.8 | 85.6 | 72.2 | 81.9 | 67.0 | 67.5 | 80.3 | 65.9 | **80.6** | **80.2** | 67.4 | 81.4 | 74.2 |
| IDMNE (Ours) | **71.73** ± **0.56** | **88.09** ± **0.33** | **75.16** ± **0.17** | **82.68** ± **0.26** | 67.58 ± 0.23 | 68.98 ± 0.28 | **82.41** ± **0.32** | **66.39** ± **0.68** | 79.32 ± 0.18 | 79.49 ± 0.52 | **69.10** ± **0.79** | 83.08 ± 0.77 | **76.17** |

**Table 6**
Comparison results (%) on Office-31 under the 3-shot setting using AlexNet. (Mean accuracy and 95% confidence interval over five trails).

| Method | W→A | D→A | Mean |
|---|---|---|---|
| S+T [5] | 61.2 | 62.4 | 61.8 |
| DANN [5] | 64.4 | 65.2 | 64.8 |
| ENT [5] | 64.0 | 66.2 | 65.1 |
| MME [5] | 67.3 | 67.8 | 67.6 |
| BiAT [67] | 68.2 | 68.5 | 68.4 |
| APE [7] | 67.6 | 69.0 | 68.3 |
| CDAC [6] | 70.1 | 70.0 | 70.0 |
| IDMNE (Ours) | **71.03** ± **0.34** | **71.32** ±**0.47** | **71.18** |

setting using AlexNet. Moreover, Table 2 also shows that the proposed algorithm achieves the highest mean accuracy and exceeds **UODAv2** by 2.90% and 1.27%, respectively, under the 1-shot and 3-shot settings using ResNet-34.

**Results on Office-Home.** To further validate the effectiveness of IDMNE, we compare our method with existing algorithms on the smaller Office-Home benchmark. For fair comparisons, we conduct experiments only under the 3-shot setting when AlexNet or ResNet-34 is the backbone network but report the experimental results under both 1-shot and 3-shot settings when VGG-16 is used as the backbone. Comparison results on Office-Home are shown in Tables 3–5, indicating that the proposed method again achieves the best average performance in all adaptation cases under all settings. In particular, Table 4 shows that in comparison to the highest accuracy achieved by previous algorithms, the average performance gain of our proposed method is respectively 1.60% and 0.91% under the 1-shot and 3-shot settings when VGG-16 is the backbone. Furthermore, our method also performs well on Office-Home with 1.04% and 0.47% performance boosts when AlexNet and ResNet-34 work as the backbone networks, respectively.

**Results on Office-31.** We also evaluate IDMNE using the adaptation tasks defined on Office-31. For a fair comparison with previous algorithms, only AlexNet is adopted as the backbone in the experiments on Office-31. Table 6 shows the comparison result under the 3-shot setting. The proposed method achieves a mean accuracy of 70.9%, which boosts the previous best-performing algorithm, **CDAC**, by 0.9%, suggesting that our algorithm also works well on small and relatively simple datasets.

### 4.4. Analysis

In this section, we first evaluate the effect of our proposed IDMNE in two aspects, including Inter-domain Mixup and Neighborhood Expansion. Then, further analysis is performed to validate the impact of several important factors w.r.t our approach.

**Ablation Study on Inter-domain Mixup.** Inter-domain Mixup involves two loss functions, i.e., $\mathcal{L}_{sdm}$ and $\mathcal{L}_{mdm}$, to achieve cross-domain feature alignment. We conduct variants of Inter-domain Mixup to verify the efficacy of each loss function. We first design a baseline experiment, namely "Baseline1", where the model is trained only with $\mathcal{L}_{sup}$. Then, adding $\mathcal{L}_{sdm}$ and $\mathcal{L}_{mdm}$ in turn to "Baseline1" forms "Baseline1+SDM", "Baseline1+MDM" and "Baseline1+SDM+MDM". As shown in Table 7, "Baseline1+SDM+MDM" improves "Baseline1" by 15.3% on average while the performance gain of "Baseline1+SDM" and "Baseline1+MDM" over "Baseline1" reaches 13.1% and 14.0%, respectively. This means that both $\mathcal{L}_{sdm}$ and $\mathcal{L}_{mdm}$ improve the performance of "Baseline1", and incorporating both into "Baseline1" achieves a greater performance improvement, demonstrating the complementarity of the two.

**Ablation Study on Neighborhood Expansion.** Neighborhood Expansion has three loss functions, including $\mathcal{L}_{psr}$, $\mathcal{L}_{nsr}$ and $\mathcal{L}_{pa}$, which are used to further optimize the model trained with $\mathcal{L}_{sup} + \mathcal{L}_{sdm} + \mathcal{L}_{mdm}$ (Called "Baseline2"). We compare diverse loss functions in Neighborhood Expansion in order to verify the validity of each loss term. For simplification, we design three variants, namely "Baseline2+PSR", "Baseline2+PSR+NSR" and "Baseline2+PSR+NSR+PA", to gradually add $\mathcal{L}_{psr}$, $\mathcal{L}_{nsr}$ and $\mathcal{L}_{pa}$ to the loss function for "Baseline2". As shown in Table 8, in comparison to the performance of "Baseline2", the gain of each variant in mean accuracy reaches 0.9%, 1.6% and 2.2%, respectively, suggesting that both Self-Regularization and Pairwise Approaching are of significance for the model's performance.

**Hyper-parameter Sensitivity to Confidence Threshold** $\tau$. $\tau$ is a hyper-parameter that determines the assignment of pseudo-labels to unlabeled target samples. We study the impact of this parameter on the overall performance of our model. As shown in Fig. 5(a)–(d), more pseudo-labels are assigned as $\tau$ decreases, but we can observe an accuracy drop of pseudo-labels at the same time, indicating the adverse effect of the noisy pseudo-labels on the model training. We set $\tau = 0.95$ according to its best performance on the validation set.

**Effect of the Proposed Complementary Label Selection Scheme in NSR.** To evaluate the effectiveness of our proposed complementary label selection scheme, we compared it to the scheme outlined in [59]. In their strategy, the complementary label is randomly chosen from all possible class labels, except the one with the highest confidence. Fig. 6(a) shows that their scheme, referred to as "IDMNE w/ NSR (random)", does not offer a significant performance advantage over our proposed scheme, here denoted as "IDMNE w/ NSR (minimum)". In fact, it may even be less effective than the cases indicated by "IDMNE w/o NSR" where **NSR** is not employed. This outcome can be attributed to the fact that **NSR** is only applied to unlabeled target samples with a predicted confidence below a predefined threshold. As depicted in Fig. 5, the pseudo-labels assigned to these samples tend to be unreliable and may contain noisy labels. Randomly selecting class labels, excluding the one with the highest confidence score, increases the likelihood of selecting a true label as a complementary label. This elevates the risk of propagating incorrect information during NSR, which has an adverse effect on the training of the adaptation model and diminishes its overall performance.

**Necessity of Pseudo Labeling.** We compare IDMNE with "IDMNE w/o pseudo labeling", which refers to that the model is trained without using any supervision from pseudo-labels. As shown in Fig. 6(b), as training progresses, the accuracy of IDMNE and "IDMNE w/o pseudo labeling" gradually improves. However, the accuracy of IDMNE improves faster. At the end, the accuracy of IDMNE surpasses that of the latter by approximately 3%. This confirms the crucial necessity of pseudo-labels for our proposed method.

**Model Calibration with Mixup.** According to [15], deep neural networks (DNNs) tend to possess poor model calibration, resulting in

**Table 7**
Ablation study of Inter-domain Mixup on DomainNet under the 3-shot setting using ResNet-34.

| Method | R→C | R→P | P→C | C→S | S→P | R→S | P→R | Mean |
|---|---|---|---|---|---|---|---|---|
| Baseline1 | 60.0 | 62.2 | 59.4 | 55.0 | 59.5 | 50.1 | 73.9 | 60.0 |
| Baseline1+SDM | 74.7 | 72.9 | 75.5 | 69.1 | 70.8 | 68.9 | 79.8 | 73.1 |
| Baseline1+MDM | 76.9 | 73.1 | 76.8 | 69.3 | 71.3 | 69.6 | 80.9 | 74.0 |
| Baseline1+SDM+MDM | 77.7 | 75.3 | 78.3 | 70.7 | 72.4 | 71.7 | 81.0 | 75.3 |

**Table 8**
Ablation study of Neighborhood Expansion on DomainNet under the 3-shot setting using ResNet-34.

| Method | R→C | R→P | P→C | C→S | S→P | R→S | P→R | Mean |
|---|---|---|---|---|---|---|---|---|
| Baseline2 | 77.7 | 75.3 | 78.3 | 70.7 | 72.4 | 71.7 | 81.0 | 75.3 |
| Baseline2+PSR | 78.5 | 76.0 | 79.2 | 72.0 | 73.1 | 72.2 | 82.2 | 76.2 |
| Baseline2+PSR+NSR | 79.8 | 76.7 | 79.8 | 71.6 | 74.4 | 73.3 | 82.9 | 76.9 |
| Baseline2+PSR+NSR+PA (i.e., IDMNE) | 80.7 | 77.0 | 80.6 | 72.1 | 75.2 | 74.2 | 82.7 | 77.5 |

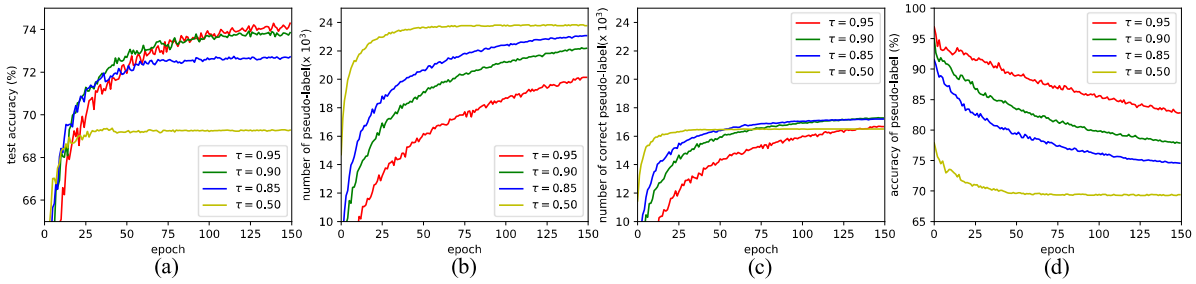



**Fig. 5.** Hyper-parameter sensitivity to confidence threshold $\tau$. We show the evolution of (a) the test accuracy in the target domain w.r.t different setting of $\tau$, (b) the number of pseudo-labels involved in samples from $D_u$ with maximum class probability prediction larger than $\tau$, (c) the number of correct pseudo-labels, and (d) the correction accuracy of pseudo-labels by comparing (b) with (c), while varying the confidence threshold $\tau$. Various colors denote different values with respect to $\tau$. We carry out these experiments on DomainNet in the adaptation scenario "$R \rightarrow S$" under the 3-shot setting using ResNet-34.

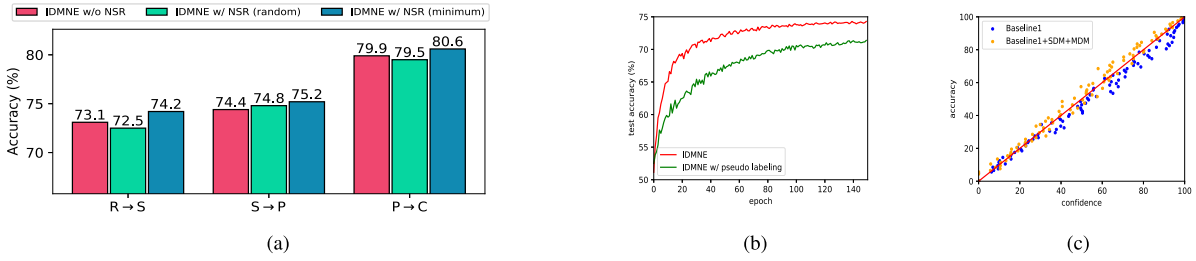
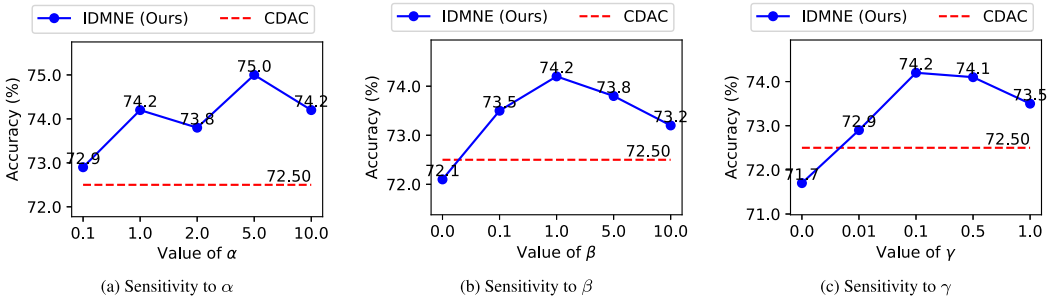



**Fig. 6.** (a) Comparison results of "IDMNE w/o NSR", "IDMNE w/ NSR (random)" and "IDMNE w/ NSR (minimum)". (b) Comparison results of IDMNE and "IDMNE w/o pseudo labeling". (c) Calibration results of "Baseline1+SDM+MDM" (with Mixup) and "Baseline1" (without Mixup) while displaying with a scatterplot between the accuracy and the average confidence per bin (100 bins in total). The red line indicates where the accuracy matches the confidence. We conducted experiments on DomainNet, specifically in the adaptation cases "$R \rightarrow S$", "$S \rightarrow P$, and "$P \rightarrow C$" for (a), as well as in the scenario "$R \rightarrow S$" for both (b) and (c). All experiments were performed under the 3-shot setting using the ResNet-34 architecture. (**Best viewed zoomed in.**).

the model generating high probability (confidence) scores for class labels that are actually incorrect. This overconfidence could give rise to the result that the model accuracy on a sample set is lower than its average predicted confidence score on the same set. Thus, overconfidence would do harm to the pseudo labeling scheme since these model predictions with high confidence are likely to produce many noisy pseudo-labels for selected unlabeled target domain samples, thereby causing negative effects in model optimization during training. A well-calibrated model, therefore, is in need. Several recent work [73,74] has demonstrated that Mixup can play a significant role in improving the calibration of DNNs. Here, we conduct experiments to explore the effect of Mixup on model calibration. For simplicity, we only choose "Baseline1" (without Mixup) and "Baseline1+SMD+MDM"(with Mixup) for comparison, both of which have previously been considered in Section 4.4. As shown in Fig. 6(c), Mixup makes the "Baseline1+SMD+MDM" model better calibrated as more sample points lie in or above the red line where the accuracy matches the confidence. Specifically, on samples at a certain confidence level, the accuracy of "Baseline1+SMD+MDM" is higher than or at least on par with that of "Baseline1".

**Hyper-parameter Sensitivity to $\alpha$, $\beta$, and $\gamma$.** We conducted case studies to investigate the sensitivity of the hyper-parameters $\alpha$, $\beta$, and $\gamma$. The results of the sensitivity analysis are displayed in Fig. 7(a)–(c). We found that in Fig. 7(a), employing the default value of $\alpha = 1.0$, as commonly referenced in MixUp operations [43,44], does not yield optimal performance. This indicates that adjusting the value of $\alpha$ can lead to improved results. Notably, even when using the default $\alpha$ value, our method consistently outperforms the state-of-the-art SSDA baseline, **CDAC** [6], which highlights its robustness to changes in $\alpha$ within this adaptation scenario. Figs. 7(b)–(c) showcase the impact of the hyper-parameters $\beta$ and $\gamma$ on test accuracy. Initially, increasing the values of $\beta$ and $\gamma$ significantly improves test accuracy. However, further increases gradually diminish accuracy. Nevertheless, the final test accuracy remains superior to that of the **CDAC** baseline. Overall, our method demonstrates low sensitivity to changes in these two hyper-parameters across a wide range. Notably, the lowest test accuracy is attained when setting both $\beta$ and $\gamma$ to 0. This may result from the excluding of the loss terms associated with Inter-domain Mixup and Neighborhood Expansion from the model training process.

**Feature Distribution and its Visualization.** To gain deeper insights into the effects of each component within Inter-domain Mixup on feature distribution alignment across both domains, we employ visualization techniques and quantitative measures. Specifically, we utilize




(a) Sensitivity to $\alpha$ (b) Sensitivity to $\beta$ (c) Sensitivity to $\gamma$

**Fig. 7.** Sensitivity to the hyper-parameters $\alpha$, $\beta$, and $\gamma$. The experiments are conducted in the adaptation task of "$R \rightarrow S$" on `DomainNet` with a 3-shot setup using the ResNet-34 backbone.

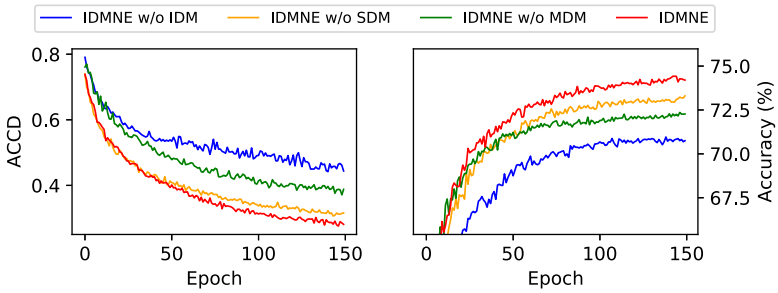



**Fig. 8.** The variations of Averaged Cluster Centroid Distance of "IDMNE w/o SDM", "IDMNE w/o MDM", and "IDMNE w/o IDM" and IDMNE (our full model) using t-SNE. The experiment is performed on the "$R \rightarrow S$" adaptation task of `DomainNet`, using the ResNet-34 backbone and the 3-shot setup.

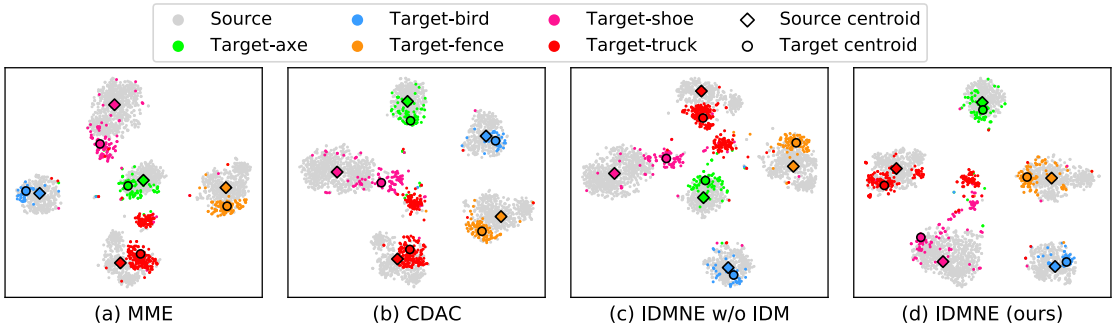



**Fig. 9.** Feature visualization of **MME**, **CDAC**, "IDMNE w/o IDM" and IDMNE (our full model) using t-SNE. The visualization is performed on the "$R \rightarrow S$" adaptation task of `DomainNet`, using the ResNet-34 backbone and the 3-shot setup. We randomly select five representative classes with distinct bright colors for their demonstration, namely "Axe" (green), "Bir" (blue), "Fence" (orange), "Shoe" (pink), and "Truck" (red). Additionally, grey data points correspond to source samples, while the cluster centroids of various classes on both the source and target domains are represented with "square" and "circle" markers. .

the Averaged Cluster Centroid Distance (ACCD) method, as elaborated in Section 4.2, to quantify the impact of Inter-domain Mixup for domain alignment. Additionally, we conduct feature visualization to provide further support for our analysis. The corresponding results are presented in Fig. 8 and 9.

Firstly, ACCD measures the distance between feature clusters of the source and target domains for all classes in the dataset, where smaller ACCDs indicate greater alignment of feature clusters between source and target domains. To assess this, we compared our full model, IDMNE, with three variants: "IDMNE w/o SDM", "IDMNE w/o MDM", and "IDMNE w/o ID". These variants represent degraded versions of IDMNE by removing sample-level data mixing (denoted by **SDM**), manifold-level data mixing (denoted by **MDM**), or both (denoted by **IDM**) from the Inter-domain Mixup. As shown in Fig. 8, all four models consistently exhibited a gradual decrease in ACCDs during model training, indicating source and target clusters across all classes become closer in the feature space. Notably, "IDMNE w/o SDM", "IDMNE w/o MDM", and IDMNE achieved better feature alignment than "IDMNE w/o IDM", with IDMNE reaching the minimum ACCD value. This suggests that the proposed Inter-domain Mixup, containing both **SDM** and **MDM**, whether used individually or in combination (**IDM**), contributes to feature alignment across domains, ultimately ensuring the superior classification performance of IDMNE.

Furthermore, we performed feature visualization using t-SNE for IDMNE, its variant "IDMNE w/o IDM", and two comparison methods, **MME** [5] and **CDAC** [6]. The results, as illustrated in Fig. 9, clearly demonstrate that IDMNE achieves more compact and aligned feature distributions for samples across both domains, surpassing the performance of **MME**, **CDAC**, and "IDMNE w/o IDM". Additionally, in comparison to "IDMNE w/o IDM", IDMNE exhibits a closer proximity of feature cluster centroids for both domains, highlighting the effectiveness of Inter-domain Mixup in promoting cross-domain feature fusion and achieving superior fusion performance.

## 5. Conclusion

In this paper, we propose Inter-domain Mixup with Neighborhood Expansion (IDMNE) for semi-supervised domain adaptation of image recognition. IDMNE consists of a well-designed cross-domain feature

alignment strategy called Inter-domain Mixup, and a practical auxiliary scheme called Neighborhood Expansion. Specifically, Inter-domain Mixup conducts sample-level and manifold-level data mixing of source-target sample pairs from labeled data. Incorporating augmented training samples and their label information reduces cross-domain distribution discrepancies by facilitating cross-domain feature alignment and alleviating label mismatch simultaneously. On the other hand, Neighborhood Expansion leverages massive high-confidence pseudo-labeled samples to diversify the label information of the target domain. Self-Regularization and Pairwise Approaching are also included in Neighborhood Expansion for reducing the uncertainty of model predictions at unlabeled target domain samples. This enables the model to produce higher probability (confidence) scores for predicted class labels corresponding to those unlabeled samples of the target domain. Finally, Inter-domain Mixup and Neighborhood Expansion are integrated into an adaptation framework, and extensive experiments demonstrate that our proposed method achieves considerable accuracy improvement over existing state-of-the-art algorithms on three commonly used benchmark datasets, i.e., `DomainNet`, `Office-Home` and `Office-31`.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation (No. 2020B1515020048), in part by the National Natural Science Foundation of China (No. 62322608, No. 61976250), in part by the Shenzhen Science and Technology Program (NO. JCYJ20220530141211024).

## References

[1] G.-H. Wang, B.-B. Gao, C. Wang, How to reduce change detection to semantic segmentation, Pattern Recognit. 138 (2023) 109384.
[2] C. Ma, X. Pan, Q. Ye, F. Tang, W. Dong, C. Xu, CrossRectify: Leveraging disagreement for semi-supervised object detection, Pattern Recognit. 137 (2023) 109280.
[3] A. Khatun, S. Denman, S. Sridharan, C. Fookes, Pose-driven attention-guided image generation for person re-identification, Pattern Recognit. 137 (2023) 109246.
[4] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: International Conference on Machine Learning, PMLR, 2015, pp. 1180–1189.
[5] K. Saito, D. Kim, S. Sclaroff, T. Darrell, K. Saenko, Semi-supervised domain adaptation via minimax entropy, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2019, pp. 8050–8058.
[6] J. Li, G. Li, Y. Shi, Y. Yu, Cross-domain adaptive clustering for semi-supervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.
[7] T. Kim, C. Kim, Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation, in: Proceedings of the European Conference on Computer Vision, (ECCV), Springer, 2020, pp. 591–607.
[8] Q. Luo, Z. Liu, L. Hong, C. Li, K. Yang, L. Wang, F. Zhou, G. Li, Z. Li, J. Zhu, Relaxed conditional image transfer for semi-supervised domain adaptation, 2021, arXiv preprint arXiv:2101.01400.
[9] C. Qin, L. Wang, Q. Ma, Y. Yin, H. Wang, Y. Fu, Contradictory structure learning for semi-supervised domain adaptation, in: Proceedings of the 2021 SIAM International Conference on Data Mining, (SDM), SIAM, 2021, pp. 576–584.
[10] H. Li, Y. Chen, D. Tao, Z. Yu, G. Qi, Attribute-aligned domain-invariant feature learning for unsupervised domain adaptation person re-identification, IEEE Trans. Inf. Forensics Secur. 16 (2020) 1480–1494.
[11] G. Kang, L. Jiang, Y. Yang, A.G. Hauptmann, Contrastive adaptation network for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4893–4902.
[12] Z. Deng, Y. Luo, J. Zhu, Cluster alignment with a teacher for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2019, pp. 9944–9953.
[13] L. Yang, Y. Wang, M. Gao, A. Shrivastava, K.Q. Weinberger, W.-L. Chao, S.-N. Lim, Deep co-training with task decomposition for semi-supervised domain adaptation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 8906–8916.
[14] C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, J. Huang, Progressive feature alignment for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 627–636.
[15] C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, On calibration of modern neural networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1321–1330.
[16] C.-X. Ren, Y.-W. Luo, D.-Q. Dai, BuresNet: Conditional bures metric for transferable representation learning, IEEE Trans. Pattern Anal. Mach. Intell. 45 (4) (2022) 4198–4213.
[17] A. Iscen, G. Tolias, Y. Avrithis, O. Chum, Label propagation for deep semi-supervised learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5070–5079.
[18] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, B. Wang, Moment matching for multi-source domain adaptation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1406–1415.
[19] H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), IEEE, 2017, pp. 5385–5394.
[20] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: European Conference on Computer Vision, Springer, 2010, pp. 213–226.
[21] J. Yang, R. Xu, R. Li, X. Qi, X. Shen, G. Li, L. Lin, An adversarial perturbation oriented domain adaptation approach for semantic segmentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, 34, (07) 2020, pp. 12613–12620.
[22] Z. Zhang, W. Chen, H. Cheng, Z. Li, S. Li, L. Lin, G. Li, Divide and contrast: source-free domain adaptation via adaptive contrastive learning, Advances in Neural Information Processing Systems 35 (2022) 5137–5149.
[23] X. Xiong, S. Li, G. Li, Unpaired image-to-image translation based domain adaptation for polyp segmentation, in: 2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI), IEEE, 2023, pp. 1–5.
[24] R. Xu, G. Li, J. Yang, L. Lin, Larger norm more transferable: an adaptive feature norm approach for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1426–1435.
[25] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, IEEE Trans. Neural Netw. 22 (2) (2010) 199–210.
[26] M. Long, Y. Cao, J. Wang, M. Jordan, Learning transferable features with deep adaptation networks, in: International Conference on Machine Learning, PMLR, 2015, pp. 97–105.
[27] M. Long, H. Zhu, J. Wang, M.I. Jordan, Deep transfer learning with joint adaptation networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 2208–2217.
[28] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, S. Saminger-Platz, Central moment discrepancy (cmd) for domain-invariant representation learning, in: International Conference on Learning Representations, (ICLR), 2017.
[29] H. Tang, K. Jia, Discriminative adversarial domain adaptation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, (04) 2020, pp. 5940–5947.
[30] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, W. Zhang, Adversarial domain adaptation with domain mixup, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, (04) IEEE, 2020, pp. 6502–6509, http://dx.doi.org/10.1609/aaai.v34i04.6123.
[31] M. Kim, H. Byun, Learning texture invariant representation for domain adaptation of semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2020, pp. 12975–12984.
[32] Y. Luo, P. Liu, T. Guan, J. Yu, Y. Yang, Adversarial style mining for one-shot unsupervised domain adaptation, Adv. Neural Inf. Process. Syst. 33 (2020).
[33] H. Nam, H.-E. Kim, Batch-instance normalization for adaptively style-invariant neural networks, Adv. Neural Inf. Process. Syst. 31 (2018) 2558–2567.
[34] Y. Lu, Q. Zhu, B. Zhang, Z. Lai, X. Li, Weighted correlation embedding learning for domain adaptation, IEEE Trans. Image Process. 31 (2022) 5303–5316, http://dx.doi.org/10.1109/TIP.2022.3193758.
[35] Y. Lu, W.K. Wong, B. Zeng, Z. Lai, X. Li, Guided discrimination and correlation subspace learning for domain adaptation, IEEE Trans. Image Process. 32 (2023) 2017–2032, http://dx.doi.org/10.1109/TIP.2023.3261758.
[36] Y. Lu, X. Luo, J. Wen, Z. Lai, X. Li, Cross-domain structure learning for visual data recognition, Pattern Recognit. 134 (2023) 109127.
[37] W. Li, L. Duan, D. Xu, I.W. Tsang, Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation, IEEE Trans. Pattern Anal. Mach. Intell. 36 (6) (2014) 1134–1148.
[38] J. Zhuang, Z. Chen, P. Wei, G. Li, L. Lin, Discovering implicit classes achieves open set domain adaptation, in: 2022 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2022, pp. 01–06.

[39] S. Li, C.H. Liu, Q. Lin, Q. Wen, L. Su, G. Huang, Z. Ding, Deep residual correction network for partial domain adaptation, IEEE Trans. Pattern Anal. Mach. Intell. 43 (7) (2021) 2329–2344, http://dx.doi.org/10.1109/TPAMI.2020.2964173.
[40] J. Li, G. Li, Y. Yu, Adaptive betweenness clustering for semi-supervised domain adaptation, IEEE Transactions on Image Processing (2023).
[41] D. Huang, J. Li, W. Chen, J. Huang, Z. Chai, G. Li, Divide and adapt: active domain adaptation via customized learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7651–7660.
[42] F. You, H. Su, J. Li, L. Zhu, K. Lu, Y. Yang, Learning a weighted classifier for conditional domain adaptation, Knowl.-Based Syst. 215 (2021) 106774.
[43] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, Mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations, 2018.
[44] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, Y. Bengio, Manifold mixup: Better representations by interpolating hidden states, in: Proceedings of the 36th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 6438–6447.
[45] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C.A. Raffel, MixMatch: A holistic approach to semi-supervised learning, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), 32, Curran Associates, Inc., 2019,
[46] D. Berthelot, N. Carlini, E.D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, C. Raffel, ReMixMatch: Semi-supervised learning with distribution matching and augmentation anchoring, in: International Conference on Learning Representations, 2020.
[47] D. Wang, Y. Zhang, K. Zhang, L. Wang, FocalMix: Semi-supervised learning for 3D medical image detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2020.
[48] Y. Wu, D. Inkpen, A. El-Roby, Dual mixup regularized learning for adversarial domain adaptation, in: A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), Proceedings of the European Conference on Computer Vision, (ECCV), Springer International Publishing, Cham, 2020, pp. 540–555.
[49] J. Na, H. Jung, H.J. Chang, W. Hwang, Fixbi: Bridging domain spaces for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1094–1103.
[50] X. Li, H. Xiong, H. An, C. Xu, D. Dou, Xmixup: Efficient transfer learning with auxiliary samples by cross-domain mixup, CoRR (2020) arXiv:2007.10252.
[51] S. Chen, X. Jia, J. He, Y. Shi, J. Liu, Semi-supervised domain adaptation based on dual-level domain mixing for semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11018–11027.
[52] S. Yan, H. Song, N. Li, L. Zou, L. Ren, Improve unsupervised domain adaptation with mixup training, 2020, arXiv preprint arXiv:2001.00677.
[53] J. Li, G. Li, F. Liu, Y. Yu, Neighborhood collective estimation for noisy label identification and correction, in: European Conference on Computer Vision, Springer, 2022, pp. 128–145.
[54] S. Wu, J. Li, C. Liu, Z. Yu, H.-S. Wong, Mutual learning of complementary networks via residual correction for improving semi-supervised classification, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 6500–6509.
[55] S. Wu, G. Deng, J. Li, R. Li, Z. Yu, H.-S. Wong, Enhancing TripleGAN for semi-supervised conditional instance synthesis and classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10091–10100.
[56] J. Li, S. Wu, C. Liu, Z. Yu, H.-S. Wong, Semi-supervised deep coupled ensemble learning with classification landmark exploration, IEEE Transactions on Image Processing 29 (2019) 538–550.
[57] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C.A. Raffel, E.D. Cubuk, A. Kurakin, C.-L. Li, FixMatch: Simplifying semi-supervised learning with consistency and confidence, in: Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 596–608.
[58] D.E. Cubuk, B. Zoph, J. Shlens, Q. Le, RandAugment - practical automated data augmentation with a reduced search space, in: NeurIPS, 2020.
[59] Y. Kim, J. Yim, J. Yun, J. Kim, Nlnl: Negative learning for noisy labels, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2019, pp. 101–110.
[60] J. Zbontar, L. Jing, I. Misra, Y. LeCun, S. Deny, Barlow twins: Self-supervised learning via redundancy reduction, 2021, arXiv preprint arXiv:2103.03230.
[61] Z. Huang, K. Sheng, W. Dong, X. Mei, C. Ma, F. Huang, D. Zhou, C. Xu, Effective label propagation for discriminative semi-supervised domain adaptation, CoRR (2020) arXiv:2012.02621.
[62] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012).
[63] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2016, pp. 770–778, http://dx.doi.org/10.1109/CVPR.2016.90.
[64] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, (ICLR), 2015.
[65] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet classification with deep convolutional neural networks, in: NIPS, 2012.
[66] D. Li, T. Hospedales, Online meta-learning for multi-source and semi-supervised domain adaptation, in: European Conference on Computer Vision, Springer, 2020, pp. 382–403.
[67] P. Jiang, A. Wu, Y. Han, Y. Shao, B. Li, Bidirectional adversarial training for semi-supervised domain adaptation, in: Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence, IJCAI-PRICAI-20, 2020.
[68] S. Mishra, K. Saenko, V. Saligrama, Surprisingly simple semi-supervised domain adaptation with pretraining and consistency, in: NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications, 2021.
[69] C. Qin, L. Wang, Q. Ma, Y. Yin, H. Wang, Y. Fu, Semi-supervised domain adaptive structure learning, IEEE Trans. Image Process. 31 (2022) 7179–7190.
[70] J. Schmidhuber, Learning to control fast-weight memories: an alternative to dynamic recurrent networks, Neural Comput. (1992) 131–139.
[71] T. Miyato, S.-i. Maeda, M. Koyama, S. Ishii, Virtual adversarial training: a regularization method for supervised and semi-supervised learning, IEEE Trans. Pattern Anal. Mach. Intell. (2018) 1979–1993.
[72] M. Chen, Q.K. Weinberger, Y. Chen, Automatic feature decomposition for single view co-training, in: ICML, 2011, pp. 953–960.
[73] S. Thulasidasan, G. Chennupati, J.A. Bilmes, T. Bhattacharya, S. Michalak, On mixup training: Improved calibration and predictive uncertainty for deep neural networks, in: Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019.
[74] J. Maroñas, D. Ramos, R. Paredes, Improving calibration in mixup-trained deep neural networks through confidence-based loss functions, CoRR (2020) arXiv:2003.09946.

**Jichang Li** is currently pursuing the Ph.D. degree in the Department of Computer Science, The University of Hong Kong. In 2020, he received the M.Eng. degree from the School of Computer Science and Technology, South China University of Technology. His current research interests include computer vision and deep learning.

**Guanbin Li** (M'15) is currently an associate professor in the School of Computer Science and Engineering, Sun Yat-sen University. He received his Ph.D. degree from The University of Hong Kong in 2016. His current research interests include computer vision, image processing, and deep learning. He is a recipient of ICCV 2019 Best Paper Nomination Award. He has authorized and co-authorized on more than 100 papers in top-tier academic journals and conferences. He serves as an area chair for the conference of VISAPP. He has been serving as a reviewer for numerous academic journals and conferences such as TPAMI, IJCV, TIP, TMM, TCyb, CVPR, ICCV, ECCV and NeurIPS.

**Yizhou Yu** (M'10, SM'12, F'19) received the Ph.D. degree from University of California at Berkeley in 2000. He is a professor at The University of Hong Kong, and was a faculty member at University of Illinois at Urbana-Champaign between 2000 and 2012. He is a recipient of 2002 US National Science Foundation CAREER Award and ACCV 2018 Best Application Paper Award. Prof Yu has served on the editorial board of IET Computer Vision, The Visual Computer, and IEEE Transactions on Visualization and Computer Graphics. He has also served on the program committee of many leading international conferences, including CVPR, ICCV, and SIGGRAPH. His current research interests include computer vision, deep learning, AI for medicine, and geometric computing.