IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 29, 2019, accepted November 22, 2019, date of publication November 26, 2019,
date of current version December 10, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956103

Adversarial Learning and Interpolation Consistency for Unsupervised Domain Adaptation

XIN ZHAO AND SHENGSHENG WANG
College of Computer Science and Technology, Jilin University, Changchun 130012, China
Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

Corresponding author: Shengsheng Wang (wss@jlu.edu.cn)

This work was supported in part by the Science and Technology Development Project of Jilin Province, China, under Grant 20190302117GX and Grant 20180101334JC, and in part by the Innovation Capacity Construction Project of Jilin Province Development and Reform Commission under Grant 2019C053-3.

ABSTRACT Unsupervised domain adaptation (UDA) aims to learn a prediction model for the target domain given labeled source data and unlabeled target data. Impressive progress has been made by adversarial learning-based methods that align distributions across domains through deceiving a domain discriminator network. However, these methods only try to align two domains and neglect the boundaries between classes, which may lead to false alignment and poor generalization performance. In contrast, consistency-enforcing methods exploit the target data posterior distribution to make the target features far away from decision boundaries. Despite their efficacy, these approaches require additional intensity augmentation to align distributions when encountering datasets with large domain discrepancy. To solve the above problems, we propose a novel UDA method that unifies the adversarial learning-based method and consistency-enforcing method together to take both domain alignment and boundaries between classes into consideration. In addition to the supervised classification on the source domain and the adversarial domain adaptation, we introduce interpolation consistency into the UDA task. To be specific, we first construct robust and informative pseudo labels for target samples, and then we encourage the prediction at an interpolation of unlabeled target samples to be consistent with the interpolation of the pseudo labels of these samples. The extensive empirical results demonstrate that our method achieves state-of-the-art results on both digit classification and object recognition tasks.

INDEX TERMS Domain adaptation, transfer learning, deep learning, image classification.

I. INTRODUCTION

Deep learning approaches have achieved remarkable success in various computer vision tasks and applications. However, these achievements often rely on large-scale labeled datasets. In many cases, the collection and annotation of training data on novel domains are extremely expensive or sometimes impossible. Hence, there is a strong motivation to train a good classification model for a target domain by using readily-available annotated data from a source domain with a different distribution. However, this attractive transfer learning paradigm suffers from the data shift problem [1], which is a huge challenge for adapting classification models to the target domain. Learning a classifier under data shift between the labeled source domain and unlabeled target domain is known as unsupervised domain adaptation (UDA) [2].

Many UDA approaches directly align the marginal distributions across domains to bridge the domain gap [3]–[9]. Notably, approaches based on adversarial learning [3], [4] divide the base model into a feature extractor G and a task-specific classifier C, and add a domain discriminator D. The domain discriminator D takes the features extracted by G and predicts which domains the features come from. The feature extractor G is learned to extract domain-invariant feature representations by deceiving the domain discriminator. Domain alignment is expected when the adversarial training reaches an equilibrium. However, these approaches may fail to create discriminative features because they do not consider the decision boundary. The feature extractor can extract ambiguous target features near the decision boundary,

The associate editor coordinating the review of this manuscript and approving it for publication was Paolo Napoletano.

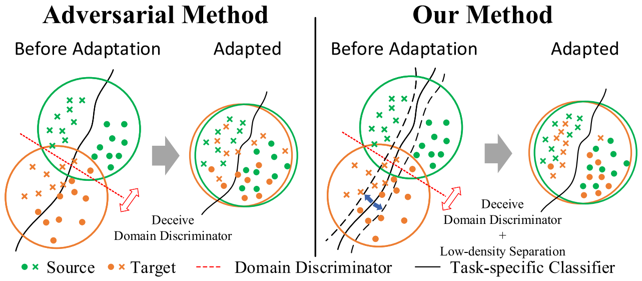


FIGURE 1. **Comparison of the adversarial method and our method. Left: The adversarial method tries to match different distributions by deceiving the domain discriminator. It ignores the decision boundaries between classes. Right: Our method tries to deceive the domain discriminator and make the features far away from the decision boundaries simultaneously.**

as it simply attempts to make the two domains similar (left side of Fig. 1).

In contrast, consistency-enforcing methods [10]–[12] exploit the target data posterior distributions to learn target discriminative features in UDA tasks. These methods encourage consistent predictions over augmented copies of the same target samples. This observation can be formalized as $p(\mathbf{y}|\tilde{\mathbf{x}}_1) \approx p(\mathbf{y}|\tilde{\mathbf{x}}_2)$, where $\tilde{\mathbf{x}}_1$ and $\tilde{\mathbf{x}}_2$ are augmented/perturbed versions of the unlabeled sample $\mathbf{x}$. These approaches move the decision boundary to low-density regions of the feature space by following the low-density separation assumption. Thus, the target features are far away from task-specific decision boundaries. Despite their efficacy, these approaches face a critical limitation. When applied to datasets with large domain discrepancy, they require additional intensity augmentation to sufficiently align the dataset distributions. This drawback makes these methods less efficient and less general.

We can find that the above two paradigms are complementary and suffer from the neglect of the other. Thus, we argue that to perform well on the target data, the adaptation model must take both domain alignment and decision boundaries between classes into account (right side of Fig. 1).

In this paper, we propose a novel UDA method that unifies the above two paradigms together to learn feature representations that are both domain-invariant and target discriminative. On the one hand, we align the two distributions across domains through the adversarial learning the same as DANN [4]. On the other hand, we refine the decision boundary using the consistency-enforcing method similarly to [10], [12]. However, random perturbations exploited by [12] are inefficient in high dimensions because only a part of perturbed inputs can move the decision boundary to low-density regions. To mitigate this problem, we introduce the interpolation consistency method [13] that is derived from the mixup [14] into the UDA task. Specifically, we first propose two techniques to produce robust and informative pseudo labels for unlabeled target samples. We average the predictions of two duplicate images differing only with respect to the adopted image augmentation to acquire a more accurate pseudo label. Then we apply a sharpening function to the average prediction to reduce the entropy of the label distribution, which can build a more informative common pseudo label for the two corresponding target samples. Second, we construct interpolations of the unlabeled target samples and interpolations of the corresponding pseudo labels using the mixup [14] operation. Then, we encourage the prediction at an interpolation of unlabeled samples to be consistent with the interpolation of the predictions at those samples. Notably, the interpolation has been proven to be a more efficient perturbation for consistency-based regularization [13]. We name the proposed method as ALIC, which is short for **A**dversarial **L**earning and **I**nterpolation **C**onsistency.

Experiments have proven our method yields state-of-the-art results on several standard datasets. Our contributions are summarized as follows:

- We argue that to perform well on the target data, the adaptation model must take both domain alignment and decision boundaries between classes into consideration. To this end, we propose a novel method for UDA which unifies the adversarial learning-based method and consistency-enforcing method together.
- We introduce interpolation consistency into the UDA task, which can move the decision boundary to low-density regions more efficiently. And we adopt two critical techniques to acquire more accurate and informative pseudo labels for unlabeled target samples to facilitate the success of interpolation consistency in the UDA task.
- We evaluate our method thoroughly by considering detailed comparisons against the state-of-the-art methods on several standard benchmark datasets. And we also conduct an empirical analysis using ablation study, feature visualization, distribution discrepancy, convergence performance and parameter sensitivity of our method.

II. RELATED WORKS

A. UNSUPERVISED DOMAIN ADAPTATION

Extensive UDA approaches have been proposed over the years, including both shallow methods and deep learning models. We focus primarily on deep learning approaches for UDA because they are more relevant to our method.

UDA approaches leverage different strategies to reduce the discrepancy between the source and target domains and can be divided into different groups. Inspired by the great achievement of GANs [15], adversarial learning has been applied to UDA and achieved impressive results [3], [4], [8], [16]–[20]. DANN [4] adds a subnetwork as domain discriminator after feature extractor layers to determine which domains the features come from, while the feature extractor is trained to fool the discriminator. In such an adversarial process, domain-agnostic features are learned to reduce the domain discrepancy. ADDA [8] outlines a generalized framework for adversarial adaptation which provided a simple and easy view to relate prior researches and introduces a novel UDA method that chooses the GAN loss and untied weight sharing.

Many domain adaptation approaches leverage a distance metric between the two domains to measure the domain discrepancy. Some methods [5], [21]–[23] utilize maximum mean discrepancy (MMD) to compare the differences between the source and target distributions. Reference [21] incorporates MMD into the last fully connected layer to learn transferable feature representations. Reference [5] embeds multi-kernel MMD into task-specific layers to further enhance the feature transferability. Reference [23] aligns the joint distributions of different domains using joint MMD criterion. In contrast, [7] tries to align the second-order statistics of the two distributions.

Several approaches are based on GANs, which reduce the domain discrepancy at the pixel level [24]–[30]. These methods directly translate source images to the target domain, and a predictor can then be trained on the transformed source images using the known source labels. Whereas GANs-based methods obtain impressive results on some simple scenarios, they behave badly when applied to complex datasets.

Entropy minimization is a successful strategy in semi-supervised learning. One way to enforce entropy minimization is to encourage the classifier outputs low-entropy predictions on unlabeled data [31]. "Pseudo Label" [32] is another way to achieve this implicitly by constructing hard labels form high-confidence predictions on unlabeled data. Due to its effectiveness, several UDA approaches [23], [25], [33], [34] have leveraged the entropy-loss to train deep neural networks.

Another popular paradigm in UDA is the consistency-enforcing methods that leverage the idea that a classifier should output the consistent predictions for an unlabeled target sample even after it has been perturbed. Several UDA methods have adopted this consistency strategy, as shown in [10], [33], [34].

B. MIXUP

Reference [14] have recently proposed a regularization method called mixup for supervised learning, achieving impressive performance in diverse tasks. Mixup regularized the neural network to have linear behavior between training samples, by enforcing that the network's output for interpolation of two inputs is close to the interpolation of the corresponding labels. Reference [35] improves mixup by performing interpolation in the hidden space representations. Furthermore, mixup has been applied to semi-supervised learning [13], [36]. Reference [13] trains a prediction model to penalize inconsistent predictions at interpolations between unlabeled training samples and [36] mixes labeled and unlabeled data together by using mixup.

III. METHOD

In the UDA problem, we are given n_s labeled examples of the source dataset $\mathcal{D}_s = \left\{\left(\mathbf{x}_i^s, \mathbf{y}_i^s\right)\right\}_{i=1}^{n_s}$ and n_t unlabeled examples of the target dataset $\mathcal{D}_t = \left\{\mathbf{x}_j^t\right\}_{j=1}^{n_t}$. The $P\left(\mathbf{x}_s, \mathbf{y}_s\right)$ and $Q\left(\mathbf{x}_t, \mathbf{y}_t\right)$ are joint distributions of the source and target domains respectively. The i.i.d. assumption is violated as $P \neq Q$. We assume that the two domains have an identical number of categories. The goal of UDA is to learn a classifier which gives accurate predictions on target test examples.

We unify the adversarial learning-based method and interpolation consistency together to take both domain alignment and decision boundaries between classes into consideration. Specifically, there are three losses in our method. In addition to the standard classification loss on labeled source data, we also have a domain adversarial loss on both source and target data and interpolation consistency loss on unlabeled target data. The domain adversarial loss is based on the adversarial domain adaptation method. As for the interpolation consistency loss, we first produce robust and informative pseudo labels using the prediction average and label sharpening techniques. Then, we apply mixup both to target samples and corresponding pseudo labels and encourage the prediction at the interpolation of these unlabeled target samples to be consistent with the interpolation of the predictions at these samples. The architecture of the proposed method is shown in the left side of Fig. 2.

In the rest of this section, we first briefly review the concept of adversarial domain adaptation in Section III-A. Second, we introduce interpolation consistency for domain adaptation in Section III-B. Finally, we summarize the overall objective of the proposed method in Section III-C.

A. ADVERSARIAL DOMAIN ADAPTATION

Adversarial domain adaptation methods, starting from Domain Adversarial Neural Network (DANN) [4], have delivered remarkable performance on UDA tasks by extracting domain-invariant features to bridge the gap between the two domains. They divide the base model f into a feature extractor G and a classifier C, and add a domain discriminator D. The domain discriminator D attempts to tell which domains the features come from while the feature extractor G is trained to fool the domain discriminator.

To learn domain-agnostic feature representations, domain discriminator D learns its parameters θ_d by minimizing the loss of the domain discriminator, while the feature extractor G learns its parameters θ_g by maximizing the loss of the domain discriminator D. The domain discriminator is optimized by:

$$\mathcal{L}_d\left(\theta_g, \theta_d\right) = \frac{1}{m} \sum_{\mathbf{x}_i \in B^s} \log D(G(\mathbf{x}_i)) + \frac{1}{m} \sum_{\mathbf{x}_i \in B_1^t} \log D(1 - G(\mathbf{x}_i)), \quad (1)$$

where B^s and B_1^t denote data batches of source and target domains respectively.

In addition, the parameters θ_c of classifier C and the parameters θ_g of feature extractor G are learned simultaneously by minimizing the classification loss of the source samples:

$$\mathcal{L}_y\left(\theta_g, \theta_c\right) = \frac{1}{m} \sum_{\mathbf{x}_i \in B^s} \mathbf{CE}(f(\mathbf{x}), \mathbf{y}), \quad (2)$$

where $\mathbf{CE}$ is the cross-entropy loss, $f = F(G(\mathbf{x}))$.

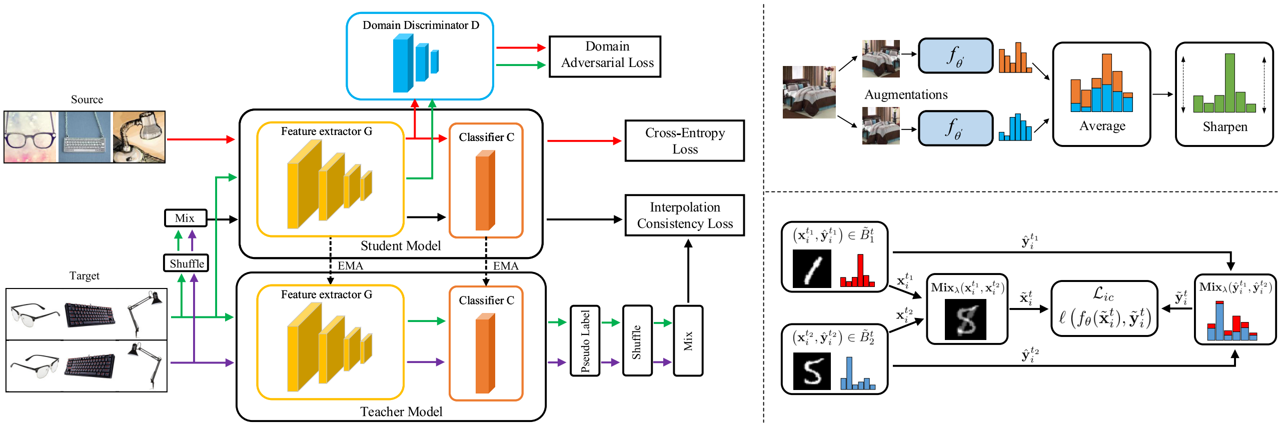

Domain Discriminator D
Domain Adversarial Loss
Source
Feature extractor G
Classifier C
Cross-Entropy Loss
Mix
Shuffle
Student Model
Interpolation Consistency Loss
EMA
EMA
Target
Feature extractor G
Classifier C
Pseudo Label
Shuffle
Mix
Teacher Model
Augmentations
$f_{\theta'}$
$f_{\theta'}$
Average
Sharpen
$(\mathbf{x}_i^{t_1}, \hat{\mathbf{y}}_i^{t_1}) \in \tilde{B}_1^t$
$(\mathbf{x}_i^{t_2}, \hat{\mathbf{y}}_i^{t_2}) \in \tilde{B}_2^t$
$\hat{\mathbf{y}}_i^{t_1}$
$\hat{\mathbf{y}}_i^{t_2}$
$\mathbf{x}_i^{t_1}$
$\mathbf{x}_i^{t_2}$
$\mathrm{Mix}_\lambda(\mathbf{x}_i^{t_1}, \mathbf{x}_i^{t_2})$
$\tilde{\mathbf{x}}_i^t$
$\mathcal{L}_{ic}$
$\ell\left(f_\theta(\tilde{\mathbf{x}}_i^t), \tilde{\mathbf{y}}_i^t\right)$
$\tilde{\mathbf{y}}_i^t$
$\mathrm{Mix}_\lambda(\hat{\mathbf{y}}_i^{t_1}, \hat{\mathbf{y}}_i^{t_2})$

FIGURE 2. **An illustration of the proposed ALIC. Left: The architecture of the proposed ALIC. The network includes feature extractor G, classifier C, domain discriminator D and with associated parameters $\theta_g, \theta_c, \theta_d$. The EMA means the parameters θ' of the teacher model is the exponential moving average of the parameters θ of the student model($\theta = \{\theta_g, \theta_c\}$). The whole model is optimized by minimizing the cross-entropy loss plus the domain adversarial loss and interpolation consistency loss in an end-to-end manner. Right-top: Diagram of prediction average and label sharpening. Two duplicate images differing only with respect to the adopted image augmentation are fed through the teacher model $f_{\theta'}$. Then, the average of the two predictions is "sharpened" by adjusting the distribution's temperature. Right-bottom: an illustration of the computation process of the interpolation consistency loss. Mixup operation: $\mathrm{Mix}_\lambda(a, b) = \lambda \cdot a + (1 - \lambda) \cdot b$. For more details, please refer to the text.**

Formally, the ultimate goal of DANN is to optimize the following objective:

$$\min_{\theta_g,\theta_c} \max_{\theta_d} \mathcal{L} = \mathcal{L}_y\left(\theta_g, \theta_c\right) + \lambda_d \mathcal{L}_d\left(\theta_g, \theta_d\right), \tag{3}$$

where λ_d is a trade-off parameter between the two objectives.

B. INTERPOLATION CONSISTENCY FOR DOMAIN ADAPTATION

1) PREDICTION AVERAGE

Since we have no access to the labels of target data, we need to produce pseudo labels for corresponding target samples using the model's predictions. These pseudo labels are later used in the mixup operation.

We apply standard data augmentations (i.e. random horizontal flips and crops) on $\mathcal{D}_t$. There are two different target batches (B_1^t and B_2^t), which contain duplicate pairs of samples differing only with respect to the adopted augmentation.

Let $B_1^t = \left\{\mathbf{x}_1^{t_1}, \ldots, \mathbf{x}_m^{t_1}\right\}$ and $B_2^t = \left\{\mathbf{x}_1^{t_2}, \ldots, \mathbf{x}_m^{t_2}\right\}$ be two batches of the augmented target samples. To produce a more robust pseudo label, we compute the average of the model's predictions across two samples by

$$\overline{p}_b = \frac{1}{2}\left(f_{\theta'}\left(\mathbf{x}_b^{t_1}\right) + f_{\theta'}\left(\mathbf{x}_b^{t_2}\right)\right), \tag{4}$$

where $\mathbf{x}_b^{t_1} \in B_1^t$ and $\mathbf{x}_b^{t_2} \in B_2^t$ are samples from the two augmented target batches, θ' is a moving average of θ ($\theta = \left\{\theta_g, \theta_c\right\}$).

2) LABEL SHARPENING

In generating a pseudo label, we perform one additional step inspired by the success of entropy minimization in UDA (discussed in section II). Given the average prediction over the two augmented target samples $\overline{p}_b$, we apply a sharpening function to reduce the entropy of the label distribution. In practice, we use the common approach of adjusting the "temperature" of this categorical distribution [37] as the sharpening function, which is defined as the operation

$$Sharpen(p, T)_i := p_i^{\frac{1}{T}} \Big/ \sum_{j=1}^{C} p_j^{\frac{1}{T}} \tag{5}$$

where p is some input categorical distribution, C is the number of the classes and T is a hyperparameter. As $T \to 0$, the output of $Sharpen(p, T)$ will approach a Dirac ("one-hot") distribution. Since we will later use $\hat{\mathbf{y}}_b = Sharpen\left(\overline{p}_b, T\right)$ as the pseudo label for the two augmented target samples, lowering the temperature encourages the model to make lower-entropy predictions. It is worth noting that $\hat{\mathbf{y}}_b$ is the pseudo label of both $\mathbf{x}_b^{t_1}$ and $\mathbf{x}_b^{t_2}$. An illustration of the prediction average and label sharpening is shown in the right-top of Fig. 2.

3) MIXUP

Mixup, first proposed in [14], is a powerful regularization method in supervised learning which enforces the classification model to change linearly in between samples. In a nutshell, mixup constructs virtual input-target vectors

$$\begin{aligned} \tilde{\mathbf{x}} &= \lambda \mathbf{x}_i + (1 - \lambda)\mathbf{x}_j, \\ \tilde{\mathbf{y}} &= \lambda \mathbf{y}_i + (1 - \lambda)\mathbf{y}_j, \end{aligned} \tag{6}$$

where $(\mathbf{x}_i, \mathbf{y}_i)$ and $\left(\mathbf{x}_j, \mathbf{y}_j\right)$ are two input-target vectors sampled randomly from the training set, and $\lambda \in [0, 1]$.

Here we extend mixup to UDA setting to refine the target decision boundary. First, we collect two batches of target samples and corresponding pseudo labels into $\hat{B}_1^t = \left(\left(\mathbf{x}_b^{t_1}, \hat{\mathbf{y}}_b\right); b \in (1, \ldots, m)\right)$ and $\hat{B}_2^t = \left(\left(\mathbf{x}_b^{t_2}, \hat{\mathbf{y}}_b\right); b \in (1, \ldots, m)\right)$. Then, we shuffle the two target batches $\hat{B}_1^t$ and $\hat{B}_2^t$. The shuffled batches are

Algorithm 1 Interpolation Consistency Loss Computation

```
Input: Two target batches $B_1^t = \{\mathbf{x}_1^{t_1}, \ldots, \mathbf{x}_m^{t_1}\}$ and $B_2^t = \{\mathbf{x}_1^{t_2}, \ldots, \mathbf{x}_m^{t_2}\}$ (same images with different image augmentations), Student network $f_\theta$, Teacher network $f_{\theta'}$ ($\theta'$ equal to the moving average of $\theta$), Sharpening temperature parameter $T$, Beta distribution parameter $\alpha$.
for $b = 1$ to $m$ do
    $\bar{p}_b = \frac{1}{2}\left(f_{\theta'}\left(\mathbf{x}_b^{t_1}\right) + f_{\theta'}\left(\mathbf{x}_b^{t_2}\right)\right)$
    $\hat{\mathbf{y}}_b = \text{Sharpen}\left(\bar{p}_b, T\right)$
end for
$\hat{B}_1^t = \left(\left(\mathbf{x}_b^{t_1}, \hat{\mathbf{y}}_b\right); b \in (1, \ldots, m)\right)$
$\hat{B}_2^t = \left(\left(\mathbf{x}_b^{t_2}, \hat{\mathbf{y}}_b\right); b \in (1, \ldots, m)\right)$
$\tilde{B}_1^t = \mathit{Shuffle}(\hat{B}_1^t) = \left(\left(\mathbf{x}_i^{t_1}, \hat{\mathbf{y}}_i^{t_1}\right); i \in (1, \ldots, m)\right)$
$\tilde{B}_2^t = \mathit{Shuffle}(\hat{B}_2^t) = \left(\left(\mathbf{x}_i^{t_2}, \hat{\mathbf{y}}_i^{t_2}\right); i \in (1, \ldots, m)\right)$
Sample $\lambda \sim \text{Beta}(\alpha, \alpha)$
for $i = 1$ to $m$ do
    $\tilde{\mathbf{x}}_i^t = \lambda \mathbf{x}_i^{t_1} + (1-\lambda)\mathbf{x}_i^{t_2}$
    $\tilde{\mathbf{y}}_i^t = \lambda \hat{\mathbf{y}}_i^{t_1} + (1-\lambda)\hat{\mathbf{y}}_i^{t_2}$
end for
$\tilde{B}^t = \left(\left(\tilde{\mathbf{x}}_i^t, \tilde{\mathbf{y}}_i^t\right); i \in (1, \ldots, m)\right)$
$\mathcal{L}_{ic}(\theta_g, \theta_c) = \frac{1}{m}\sum_{(\tilde{\mathbf{x}}_i^t, \tilde{\mathbf{y}}_i^t) \in \tilde{B}^t} \ell\left(f_\theta(\tilde{\mathbf{x}}_i^t), \tilde{\mathbf{y}}_i^t\right)$
return $\mathcal{L}_{ic}$
```

formalized as: $\tilde{B}_1^t = \left(\left(\mathbf{x}_i^{t_1}, \hat{\mathbf{y}}_i^{t_1}\right); i \in (1, \ldots, m)\right)$ and $\tilde{B}_2^t = \left(\left(\mathbf{x}_i^{t_2}, \hat{\mathbf{y}}_i^{t_2}\right); i \in (1, \ldots, m)\right)$.

Now, we can constructs the following interpolations of training samples using mixup:

$$\begin{aligned}\tilde{\mathbf{x}}_i^t &= \lambda \mathbf{x}_i^{t_1} + (1-\lambda)\mathbf{x}_i^{t_2},\\ \tilde{\mathbf{y}}_i^t &= \lambda \hat{\mathbf{y}}_i^{t_1} + (1-\lambda)\hat{\mathbf{y}}_i^{t_2},\end{aligned} \tag{7}$$

where $\lambda \sim \text{Beta}(\alpha, \alpha)$, for $\alpha \in (0, \infty)$. Our goal is to train the student model f_θ to predict the fake label $\tilde{\mathbf{y}}_i^t$ at the interpolation point $\tilde{\mathbf{x}}_i^t$. The discrepancy between the prediction $f_\theta(\tilde{\mathbf{x}}_i^t)$ and $\tilde{\mathbf{y}}^t$ is measured by the mean squared loss in our experiment. We formalize our interpolation consistency loss as follows:

$$\mathcal{L}_{ic}(\theta_g, \theta_c) = \frac{1}{m}\sum_{(\tilde{\mathbf{x}}_i^t, \tilde{\mathbf{y}}_i^t) \in \tilde{B}^t} \ell\left(f_\theta(\tilde{\mathbf{x}}_i^t), \tilde{\mathbf{y}}_i^t\right), \tag{8}$$

where $\tilde{B}^t = \left(\left(\tilde{\mathbf{x}}_i^t, \tilde{\mathbf{y}}_i^t\right); i \in (1, \ldots, m)\right)$. An illustration of interpolation consistency is shown in the right-bottom of Fig. 2. The full algorithm of computing the interpolation consistency loss is provided in algorithm 1.

C. OVERALL OBJECTIVES

As discussed in section I, domain adversarial methods and consistency-enforcing methods are complementary to each other. So we unify domain adversarial method and interpolation consistency together. Combing the losses (1), (2) and (8) together, we have the following training objective:

$$\min_{\theta_g, \theta_c} \max_{\theta_d} \mathcal{L} = \mathcal{L}_y\left(\theta_g, \theta_c\right) + \lambda_d \mathcal{L}_d\left(\theta_g, \theta_d\right) + \lambda_c \mathcal{L}_{ic}\left(\theta_g, \theta_c\right), \tag{9}$$

where λ_d and λ_c are weights that trade-off among the three objectives.

D. DISCUSSION

In this section, we show the differences between our method and two relevant methods DANN [4] and SE [10]. (1) Both DANN and our method try to align the feature distributions to reduce the domain discrepancy by using adversarial learning, but only considering domain alignment for UDA is far from enough. Our method not only considers domain alignment but also takes decision boundaries between classes into account, which can learn feature representations that are both domain-invariant and target discriminative. (2) SE is a consistency-enforcing method, which pushes the decision boundary away from data points by penalizing inconsistent predictions over randomly perturbed copies of the same target samples. Different from SE, we replace randomly perturbations with the interpolation perturbations following the interpolation consistency [13], which can refine the decision boundaries more efficiently. To facilitate the success of interpolation consistency in the UDA task, we also adopt two key techniques to acquire more accurate and informative pseudo labels for unlabeled target samples. Additional, Our method alleviates the dependency of consistency-forcing methods on additional intensity augmentation by exploiting the adversarial domain adaptation to align the feature distributions.

IV. EXPERIMENTS

In this section, we first illustrate the datasets, baseline methods, and implementation details. Then, we show extensive empirical results and further analysis.

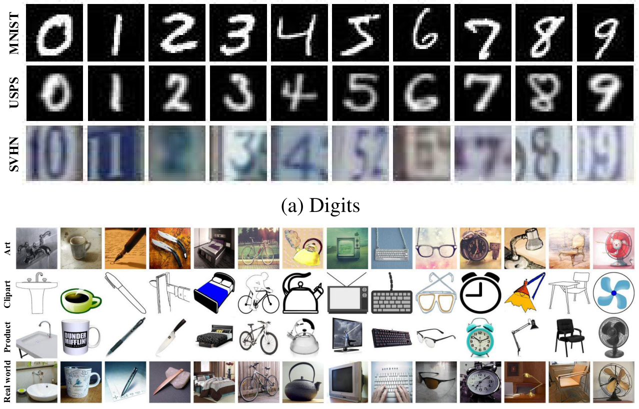
A. DATASETS

Digits We investigate three digits datasets: **MNIST** [38], **USPS** [39] and **SVHN** [40]. The three datasets all contain digits of 10 classes ranging from 0 to 9. In particular, MNIST and USPS contain 28 × 28 and 16 × 16 grey images respectively; SVHN consists of 32 × 32 color images which might contain more than one digit in each image. We conduct experiment on three transfer tasks: MNIST to USPS (**M** → **U**), USPS to MNIST (**U** → **M**) and SVHN to MNIST (**S** → **M**). In our experiment, we use all training data and report results on standard test sets. Fig. 3(a) shows image samples from these three digits datasets.

Office-Home [41] is a very challenging dataset for UDA task, which consists of 15,500 samples in total from 65 categories of everyday objects in office and home scenes (Fig. 3(b)). There are four domains: Art (**Ar**), Clipart (**Cl**), Product (**Pr**) and Real-World (**Rw**). The images of these domains have significantly different appearances and backgrounds.

B. BASELINE METHODS

We unify domain adversarial method and consistency-enforcing method together to improve the performance of the adaptation model. Apparently, the adversarial method



(a) Digits

(b) Office-Home

FIGURE 3. Samples from datasets used. (a) Samples from MNIST, USPS, and SVHN datasets. (b) Samples from four domains of Office-Home dataset.

DANN [4] and the consistency-enforcing method SE [10] can be regarded as the main baselines of our approach. To further verify the efficacy of our approach, we also compare it with a variety of state-of-the-art deep UDA models.

All baseline methods are listed as follows: (1) **Source Only** trains a classification model only on labeled source data to predict target test samples directly, which can serve as an empirical lower bound of target performance. (2) **DAN** [5] learns a transfer model by embedding multi-kernel MMD into several task-specific layers. (3) **JAN** [23] utilizes joint MMD criterion to align joint distributions of different domains. (4) **DANN** [4] enables domain adversarial learning by training the feature extractor to deceive a domain discriminator which tries to classify features of different domains. (5) **ADDA** [8] designs a generalized framework for UDA and proposes a novel adaptation model using GAN loss and united weight sharing. (6) **SBADA-GAN** [25] jointly optimizes bi-directional image transformations to learn a robust and general adaptation model. (7) **I2I-Adapt** [30] learns transferable feature representations by unifying cycle-consistency, image reconstruction and domain adversarial learning together. (8) **UNIT** [29] proposes a novel image translation framework based on the shared-latent space assumption, and this framework can also be applied to domain adaptation. (9) **GTA** [26] utilizes a generator-discriminator pair to match the feature distributions across domains in the learned feature space. (10) **CyCADA** [42] adapts between domains on both pixel level and feature level. (11) **SE** [10] explores the use of self-ensembling for UDA. (12) **CDAN** [18] conditions the domain discriminator on classifier predictions to capture the multimodal data structures. (13) **SWD** [43] aligns feature distributions between domains by using the Wasserstein metric to measure the discrepancy of two task-specific classifiers. For a fair comparison, we re-implement DANN using the same network architecture and image augmentation as our method, and the results of other approaches are taken from the corresponding papers.

C. IMPLEMENTATION DETAIL

1) NETWORK ARCHITECTURES

We adopt the same base network following CDAN [18] in the digits experiments. For discriminator, we also use the same architecture with CDAN, x → 500 → 500 → 1. Only random image cropping is adopted in this setting. For the Office-Home dataset, we employ a 50 layer ResNet [44] network pre-trained on ImageNet [45]. Following RTN [22] and DANN [4] , a bottleneck layer fcb with 256 units is added after the $fc7$ layer for safer transfer representation learning. For discriminator, we use the same architecture with DANN, x → 1024 → 1024 → 1. We adopt image random flipping and cropping following JAN [23].

2) PARAMETERS

Our method is implemented by **Pytorch**. We use mini-batch SGD with a momentum of 0.9. The learning rate is adjusted by a decay strategy proposed by [4]: $lr_p = \frac{lr_0}{(1+\omega p)^{\phi}}$, $lr_0 = 0.01$, $\omega = 10$, $\phi = 0.75$, and p is changed from 0 to 1 as the training goes on. The learning rate of the pre-trained layers is divided by 10. We set the batch size to 64 for each domain in the digits experiment. As for Office-Home dataset, the batch size is selected as 36 due to the memory limitation of GPU. We adopt the same schedule strategy as [4] to gradually change the value of λ_d by computing $\lambda_d = \frac{2}{1+\exp(-\delta p)} - 1$, where $\delta = 10$. We set $\lambda_c = K \times \lambda_d$ to focus more on the interpolation consistency as the training proceeds, and K is selected as 30 for all datasets. The Beta distribution parameter α is fixed as 1 following [13] and the decay coefficient of the teacher model is fixed as 0.99 following [10]. We set $T = 0.5$ throughout all experiments. We follow standard evaluation protocols for UDA as [3], [5].

D. RESULTS

We present our experimental results and compare with various UDA approaches on both Digits and Office-Home datasets. For a fair comparison, the results of other approaches are taken from their original papers. The results obtained on the digits datasets are provided in Table 1. All adaptation methods yield impressive improvement over the source-only network which is trained by only using the

TABLE 1. Accuracy (%) on the Digit datasets for UDA. † indicates our implementation of DANN [4].

Method	M → U	U → M	S → M	Avg
Source Only	83.4	72.0	80.3	78.6
DANN† [4]	93.5	96.1	87.8	92.5
ADDA [8]	89.4	90.1	86.3	88.6
SBADA-GAN [25]	97.6	95.0	76.1	89.6
I2I-Adapt [30]	95.1	92.2	92.1	93.1
UNIT [29]	96.0	93.6	90.5	93.5
GTA [26]	95.3	90.8	92.4	92.8
CyCADA [42]	95.6	96.5	90.4	94.2
CDAN [18]	96.5	97.1	89.2	94.3
SE [10]	**98.1**	97.3	98.6	98.0
SWD [43]	**98.1**	97.1	98.9	98.0
ALIC (ours)	97.8	**99.2**	**99.5**	**98.8**

TABLE 2. Accuracy (%) on the Office-Home dataset [41] for UDA. All methods are based on the ResNet-50 model.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
Source Only	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [5]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN [4]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN [23]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
SE [10]	48.8	61.8	72.8	54.1	63.2	65.1	50.6	**49.2**	72.3	66.1	55.9	78.7	61.5
CDAN [18]	**50.6**	65.9	73.4	55.7	62.7	64.2	51.8	49.1	74.5	68.2	**56.9**	80.7	62.8
ALIC (ours)	45.8	**68.5**	**75.1**	**57.4**	**69.2**	**70.1**	**55.1**	48.2	**77.1**	**68.5**	54.8	**81.8**	**64.1**

source data. Our method shows competitive performance in all three transfer tasks. Specifically, our method significantly outperforms DANN by 4.3%, 3.1% and 11.7% on M→U, U→M and S→M, respectively. On average, we achieve an improvement of 0.8% over the state-of-the-art methods SE and SWD.

We also conduct experiments on the challenging Office-Home dataset. Table 2 shows that our method outperforms all comparison methods. Specifically, our method substantially outperforms DANN and SE by 6.5% and 2.6% respectively. Our method also boosts the accuracy of state-of-the-art method CDAN by 1.3%. The success of our method could indicate that taking both domain alignment and decision boundaries into consideration is beneficial for UDA models.

From Table 1 and 2, we can get several observations. (1) ALIC achieves better performance than domain alignment approaches (e.g. DAN, DANN, and JAN). This confirms that taking boundaries between classes into consideration is beneficial for the UDA model. (2) ALIC achieves better accuracy than SE (+2.6%) in the challenging Office-Home dataset. It proves that domain alignment can alleviate the dependency of consistency-enforcing methods on additional intensity augmentation.

E. ANALYSIS

1) ABLATION STUDY

To investigate how adversarial domain adaptation and interpolation consistency benefit the adaptation performance, we remove the domain adversarial loss and the interpolation consistency loss from the overall objective (9) respectively, and these two experimental settings are denoted as ALIC (w/o $\mathcal{L}_d$) and ALIC (w/o $\mathcal{L}_{ic}$). Table 3 shows that both losses are important for our method and when one of them is removed, the mean accuracy drops 17.2% and 6.3% respectively. We further investigate the effect of the prediction average and label sharpening. ALIC (w/o PA) uses a single target sample prediction rather than the average prediction of two augmented target samples and ALIC (w/o LS) removes the label sharpening from our model (i.e. $T = 1$ in (5)). As shown in Table 3, both prediction average and label sharpening are useful for our method and when one of them is removed, the mean accuracy drops 2.9% and 3.1% respectively. Lastly, to verify the superiority of the interpolation consistency compared to the random perturbations based-method [10], [11], we replace the interpolation consistency loss with the mean square loss over the two augmented target batches differing only with respect to the adopted augmentation directly and this setting is denoted as ALIC (random). Table 3 depicts that ALIC outperforms ALIC (random) by 3.5% on average, validating the efficacy of the interpolation consistency for moving the decision boundary to low-density regions.

TABLE 3. Ablation experiments on the Digit datasets under different settings.

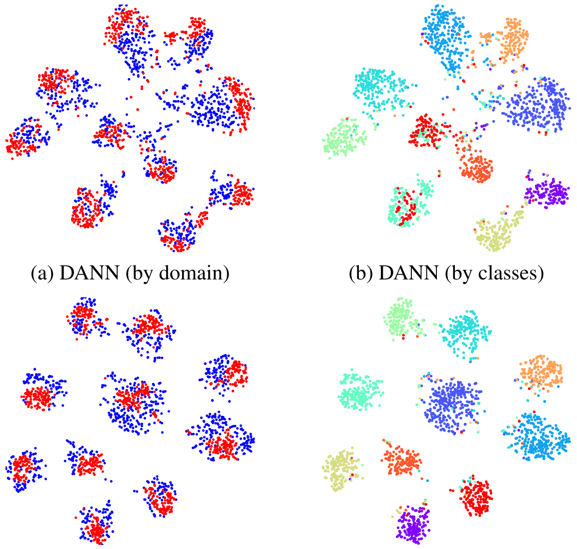
Method	M→U	U→M	S→M	Avg
Source Only	83.4	72.0	80.3	78.6
ALIC (w/o $\mathcal{L}_d$)	87.4	74.0	83.5	81.6
ALIC (w/o $\mathcal{L}_{ic}$)	93.5	96.1	87.8	92.5
ALIC (w/o PA)	94.8	97.3	95.6	95.9
ALIC (w/o LS)	97.2	98.8	91.2	95.7
ALIC (random)	96.6	90.8	98.5	95.3
ALIC	**97.8**	**99.2**	**99.5**	**98.8**

2) FEATURE VISUALIZATION

We visualize the network activations from feature extractors of DANN and ALIC on the adaptation task S → M by t-SNE [46] in Fig.4. For features of DANN, two domains are aligned together, however, there are still many ambiguous features lying between different category clusters. In contrast, for features of ALIC, the shared categories across domains are well aligned while different categories are well distinguished, which leads to better adaptation accuracy. The superior results suggest that our ALIC is able to learn features that are both domain invariant and target discriminative.

3) DISTRIBUTION DISCREPANCY

$\mathcal{A}$-distance can be used to measure distribution discrepancy [47]. The empirical $\mathcal{A}$-distance is defined as $d_{\mathcal{A}} = 2(1 - 2\epsilon)$, where ϵ denotes the test error of a classifier trained to discriminate the features of different domains. We exploit a kernel SVM as the classifier. A smaller $d_{\mathcal{A}}$ means a smaller distribution discrepancy. Table. 4 demonstrates $d_{\mathcal{A}}$ on task S → M with features of Source Only, DANN and our ALIC. From this table, we can see that our ALIC significantly reduces the $\mathcal{A}$-distance compared with Source Only model, implying that our method is helpful for the domain alignment. However, when compared to DANN, ALIC shows smaller improvement for $\mathcal{A}$-distance, but boosts the performance by a


(a) DANN (by domain)

(b) DANN (by classes)

(c) ALIC (by domain)

(d) ALIC (by classes)

FIGURE 4. Feature visualization for embedding of digit datasets for adapting SVHN to MNIST using t-SNE algorithm. Source and target samples are denoted as blue and red points in the first column. Each class is encoded by a color in the second column. (a), (b) learned features for DANN. (c), (d) learned features for our method ALIC.

TABLE 4. Distribution discrepancy of different approaches.

Method	Source Only	DANN [4]	ALIC (ours)
$\mathcal{A}$-distance	1.48	0.78	0.71

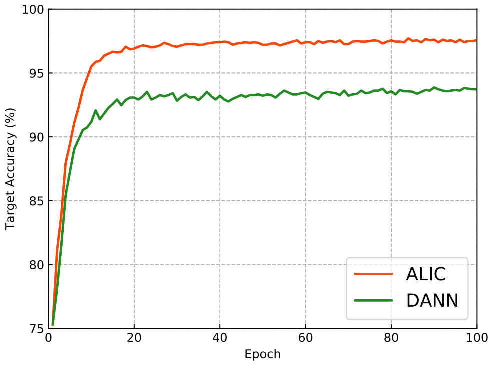


FIGURE 5. Test accuracy with the number of training epochs.

large margin, which demonstrates that only taking the domain alignment into consideration for UDA is far from enough.

4) CONVERGENCE PERFORMANCE

We demonstrate the convergence performances of ALIC and DANN in Fig. 5. The target accuracy curves of the two approaches on task $M \rightarrow U$ are plotted. ALIC shows similar convergence rate with DANN but better performance.

5) PARAMETER SENSITIVITY

We utilize the parameter K to determine λ_c which controls the importance of enforcing interpolation consistency.

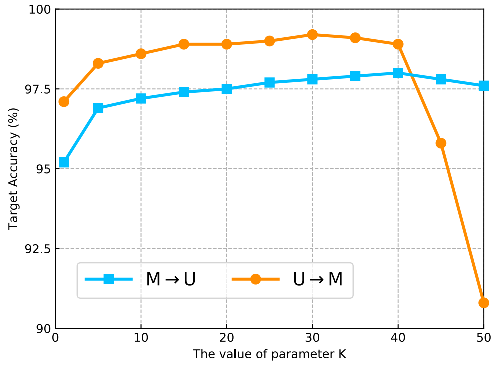


FIGURE 6. The relationship between accuracy and parameter *K*.

We conduct case studies to investigate the sensitivity of K by choosing it in the range of $\{1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$ on tasks $M \rightarrow U$ and $U \rightarrow M$ in Fig. 6. As the value of K gets larger, the accuracy steadily increases before decreasing. The accuracy is stable when $K \in \{5, 10, 15, 20, 25, 30, 35, 40\}$, indicating that our approach is robust with a wide range of K.

V. CONCLUSION

We propose a novel method called ALIC for UDA which considers both domain alignment and decision boundaries between classes. The method unifies adversarial learning based-method and consistency-enforcing method together to learn feature representations that are both domain-invariant and target discriminative. Specifically, we propose prediction average and label sharpening to produce robust and informative pseudo labels for unlabeled target samples and introduce interpolation consistency into the UDA task to refine decision boundaries more efficiently. Experiments on several standard benchmark datasets verify the efficacy of our method. Further, the analysis is provided in terms of ablation study, feature visualization, distribution discrepancy, convergence performance and parameter sensitivity for better insight about the proposed method.

REFERENCES

[1] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf, *Covariate Shift and Local Learning by Distribution Matching*. Cambridge, MA, USA: MIT Press, 2009, pp. 131–160.

[2] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[3] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.

[4] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, May 2015.

[5] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[6] P. O. Pinheiro, "Unsupervised domain adaptation with similarity learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8004–8013.

[7] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 443–450.

[8] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7167–7176.

[9] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2272–2281.

[10] G. French, M. Mackiewicz, and M. Fisher, "Self-ensembling for visual domain adaptation," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–20.

[11] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1163–1171.

[12] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1195–1204.

[13] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," Mar. 2019, *arXiv:1903.03825*. [Online]. Available: https://arxiv.org/abs/1903.03825

[14] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, Montreal, QC, Canada, 2014, pp. 2672–2680.

[16] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Proc. 22nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[17] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5419–5428.

[18] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1640–1650.

[19] Y. Zou, Z. Yu, B. V. K. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 289–305.

[20] C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, and J. Huang, "Progressive feature alignment for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 627–636.

[21] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," Dec. 2014, *arXiv:1412.3474*. [Online]. Available: https://arxiv.org/abs/1412.3474

[22] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 136–144.

[23] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 2208–2217.

[24] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3722–3731.

[25] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo, "From source to target and back: Symmetric bi-directional adaptive GAN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8099–8108.

[26] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8503–8512.

[27] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2107–2116.

[28] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," in *Proc. ICLR*, 2016, pp. 1–14.

[29] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 700–708.

[30] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4500–4509.

[31] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. 17th Int. Conf. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, 2004, pp. 529–536.

[32] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. ICML Workshop Challenges Represent. Learn.*, 2013, pp. 1–6.

[33] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 2988–2997.

[34] R. Shu, H. H. Bui, H. Narui, and S. Ermon, "A dirt-t approach to unsupervised domain adaptation," in *Proc. 6th Int. Conf. Learn. Represent.*, 2018, pp. 1–19.

[35] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6438–6447.

[36] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, "MixMatch: A holistic approach to semi-supervised learning," May 2019, *arXiv:1905.02249*. [Online]. Available: https://arxiv.org/abs/1905.02249

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, MA, USA: MIT Press, 2016.

[38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[39] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994.

[40] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011, pp. 1–9.

[41] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5018–5027.

[42] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1994–2003.

[43] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 10285–10295.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[45] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[46] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[47] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, nos. 1–2, pp. 151–175, May 2010.

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

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