Adversarial-Learned Loss for Domain Adaptation

Minghao Chen, Shuai Zhao, Haifeng Liu, Deng Cai*
State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, Hangzhou, China
Fabu Inc., Hangzhou, China
Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou, China
{minghaochen01, zhaoshuaimcc}@gmail.com, {haifengliu, dcai}@zju.edu.cn

Abstract

Recently, remarkable progress has been made in learning transferable representation across domains. Previous works in domain adaptation are majorly based on two techniques: domain-adversarial learning and self-training. However, domain-adversarial learning only aligns feature distributions between domains but does not consider whether the target features are discriminative. On the other hand, self-training utilizes the model predictions to enhance the discrimination of target features, but it is unable to explicitly align domain distributions. In order to combine the strengths of these two methods, we propose a novel method called Adversarial-Learned Loss for Domain Adaptation (ALDA). We first analyze the pseudo-label method, a typical self-training method. Nevertheless, there is a gap between pseudo-labels and the ground truth, which can cause incorrect training. Thus we introduce the confusion matrix, which is learned through an adversarial manner in ALDA, to reduce the gap and align the feature distributions. Finally, a new loss function is auto-constructed from the learned confusion matrix, which serves as the loss for unlabeled target samples. Our ALDA outperforms state-of-the-art approaches in four standard domain adaptation datasets. Our code is available at https://github.com/ZJULearning/ALDA.

Introduction

In recent years, deep learning has made impressive progress in the classification task. The success of deep neural networks is based on the large scale datasets with a tremendous amount of labeled samples (Deng et al. 2009). However, in many practical situations, a large number of labeled samples are inaccessible. The deep neural networks pre-trained on existing datasets cannot generalize well on the new data with different appearance characteristics. Essentially, the difference in data distribution between domains makes it difficult to transfer knowledge from the source to target domains. This transferring problem is known as *domain shift* (Torralba and Efros 2011).

Unsupervised domain adaptation (UDA) tackles the above *domain shift* problem while transferring the model from a labeled source domain to an unlabeled target domain. The common idea of UDA is to make features extracted by neural networks similar between domains (Long et al. 2015; Ganin et al. 2016). In particular, the domain-adversarial learning methods (Ganin et al. 2016; Tzeng et al. 2017) train a domain discriminator to distinguish whether the feature is from the source domain or target domain. To fool the discriminator, the feature generator has to output similar source and target feature distributions. However, it is challenging for this type of UDA methods to learn discriminative features on the target domain (Saito et al. 2018; Xie et al. 2018). That is because they overlook whether the aligned target features can be discriminated by the classifier.

Recently, self-training based methods (French, Mackiewicz, and Fisher 2018; Zou et al. 2018; Chen, Xue, and Cai 2019) become another solution for UDA and achieve state-of-the-art performance on multiple tasks. A typical way of self-training is to generate pseudo-labels corresponding to large prediction probability of target samples and train the model with these pseudo-labels. In this way, the features

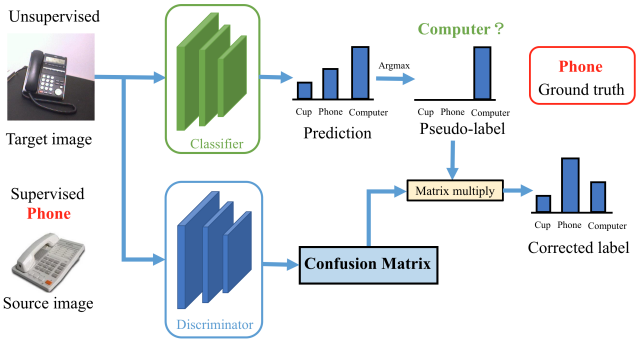


Figure 1: The illustration of proposed adversarial-learned loss (ALDA). There is a gap between pseudo-label predicted by the model and the ground truth which is unavailable on the target domain. We employ a discriminator network to produce a confusion matrix to correct the pseudo-label, which then serves as the training label for the target sample.

*Corresponding author

contributing to the target classification are enhanced. However, the alignment between the source and target feature distributions is implicit and has no theoretical guarantee. With unmatched target features, self-training based methods can lead to a drop of performance in the case of shallow networks (Zou et al. 2018; Saito et al. 2019).

In conclusion, domain-adversarial learning is able to align the feature distributions with a theoretical guarantee, while self-training can learn discriminative target features. It is ideal to have a method to combine the advantages of these two types of methods. To achieve this goal, we first analyze the loss function of self-training with pseudo-labels (Zou et al. 2018) on the unlabeled target domain. Previous works in learning from noisy labels (Sukhbaatar and Fergus 2014; Zhang and Sabuncu 2018) proposed accounting for noisy labels with a confusion matrix. Following their analyzing approach, we reveal that the loss function using pseudo-labels (Zou et al. 2018) differs from the loss function learned with the ground truth by a confusion matrix. Concretely, the commonly used cross entropy loss becomes:

$$\begin{aligned}\mathcal{L}_T(x) &= \sum_{k=1}^{K} -p(y=k|x)\log p(\hat{y}=k|x) \\ &= \sum_{k=1}^{K}\sum_{l=1}^{K} -p(y=k|\hat{y}=l,x)p(\hat{y}=l|x)\log p(\hat{y}=k|x),\end{aligned}$$

where K represents the number of categories, y is the ground truth label for the sample x, $\hat{y}$ is the model prediction, i.e., pseudo-labels, and $p(y=k|\hat{y}=l,x)$ is the (k,l)-th component of the confusion matrix.

If the confusion matrix can be estimated correctly, we can minimize the noise in pseudo-labels and boost the training of target samples. In this paper, we propose a novel method called Adversarial-learned Loss for Domain Adaptation (ALDA). As illustrated in Fig. 1, we generate the confusion matrix with a discriminator network. After multiplying with the confusion matrix, the pseudo-label vector turns into a corrected label vector, which serves as the training label on the target domain. As there is no direct way to optimize the confusion matrix, we learn it with *noise-correcting domain discrimination*. Specifically, the domain discriminator has to produce different corrected labels for different domains, while the feature generator aims to confuse the domain discriminator. The adversarial process finally leads to a proper confusion matrix on the target domain.

The main contributions of this paper are as follows:

- We analyze the noise in pseudo-labels with the confusion matrix, and propose our Adversarial-learned Loss for Domain Adaptation (ALDA) method, which uses adversarial learning to estimate the confusion matrix.
- We theoretically prove that ALDA can align the feature distributions between domains and correct the target prediction of the classifier. In this way, ALDA takes the strengths of domain-adversarial learning and self-training based methods.
- ALDA can outperform state-of-the-art methods on four standard unsupervised domain adaptation datasets.

Related Work

Unsupervised Domain Adaptation. With the success of deep learning, unsupervised domain adaptation (UDA) (Tzeng et al. 2014; Long et al. 2015; 2017b; Ganin et al. 2016) has been embedded into deep neural networks to transfer the knowledge between the labeled source domain and unlabeled target domain. It has been revealed that the accuracy of the classifier on the target domain is bounded by the accuracy of the source and the domain discrepancy (Ben-David et al. 2010). Therefore, the major line of the current UDA study is to align the distributions between the source and target domains. The distribution divergence between domains can be measured by Maximum Mean Discrepancy (MMD) (Tzeng et al. 2014; Long et al. 2015) or second-order statistics (Sun and Saenko 2016).

Domain-adversarial Methods. The domain-adversarial learning-based methods (Ganin et al. 2016; Tzeng et al. 2017) utilize a domain discriminator to represent the domain discrepancy. These methods play a minimax game: the discriminator is trained to distinguish the feature come from the source or target sample while the feature generator has to confuse the discriminator. However, due to practical issues, e.g., mode collapse (Che et al. 2017), domain-adversarial learning cannot match the multi-modal distributions. Recently, together with the prediction of classifier (Long et al. 2017a; Hong et al. 2018), the discriminator can match the distributions of each category, which significantly enhances the final classification results.

Self-training Methods. Semi-supervised learning (Lee 2013; Grandvalet and Bengio 2004; Tarvainen and Valpola 2017) is a similar task with domain adaptation, which also deals with labeled and unlabeled samples. With the data "manifold" assumption, some methods train the model based on the prediction of itself to smooth the decision boundary around the data. In particular, (Grandvalet and Bengio 2004) minimizes the prediction entropy as a regularizer for unlabeled samples. Pseudo-label method (Lee 2013) selects high-confidence predictions as training target for unlabeled samples. Mean Teacher method (Tarvainen and Valpola 2017) sets the exponential moving average of the model as the teacher model and lets the prediction of the teacher model guide the original model.

Recently, many works apply the above self-training based methods to unsupervised domain adaptation (Zou et al. 2018; Chen, Xue, and Cai 2019; French, Mackiewicz, and Fisher 2018). These UDA methods implicitly encourage the class-wise feature alignment between domains and achieve surprisingly good results on multiple UDA tasks.

Methods

Preliminaries

For unsupervised domain adaptation, we have a labeled source domain $\mathcal{D}_S = \{(x_s^i, y_s^i)\}_{i=1}^{n_s}$ and a unlabeled target domain $\mathcal{D}_T = \{x_t^j\}_{j=1}^{n_t}$. We train a generator network G to extract the high-level feature from the data x_s or x_t, and a classifier network C to finish the K-class classification task on the feature space. The classifier C outputs probability

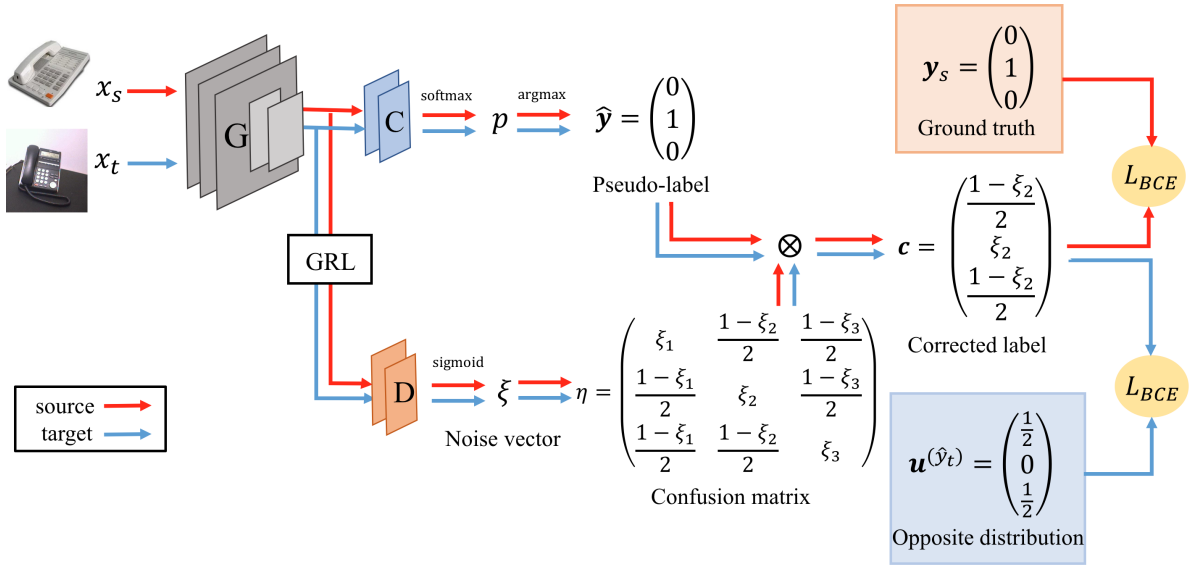


Figure 2: The illustration of noise-correcting domain discrimination ($K = 3$). The confusion matrix η is class-wise uniform with the vector ξ generated by the discriminator D. The corrected pseudo-label $\mathbf{c}$ is generated by multiplying the confusion matrix η and the pseudo-label vector $\hat{\mathbf{y}}$. For the source sample, the target of $\mathbf{c}$ is the ground truth $\mathbf{y_s}$, and the target is the opposite distribution for the target sample. The generator G is designed to confuse the above targets. Therefore, we add a gradient reverse layer (GRL) (Ganin et al. 2016) to achieve the minimax optimization.

vectors $\mathbf{p}_s, \mathbf{p}_t \in \mathbb{R}^K$, indicating the prediction probability of x_s, x_t respectively.

In this paper, we consider providing a proper loss function on the target domain. Theoretically, the ideal loss function is the loss with the ground truth y_t:

$$\mathcal{L}_T(x_t, \mathcal{L}) = \sum_{k=1}^{K} p(y_t = k|x_t)\mathcal{L}(\mathbf{p}_t, k), \tag{1}$$

where $\mathcal{L}$ is a basic loss function, e.g., cross entropy (CE), mean absolute error (MAE).

However, the target ground truth y_t is unavailable in the UDA setting. Pseudo-label method (Lee 2013; Zou et al. 2018) substitutes y_t with the model prediction: $\hat{y}_t = \text{argmax}_k\ \mathbf{p}_t^k$, if $\max_k\ \mathbf{p}_t^k > \delta$, where δ is a threshold. As mentioned in the introduction, we analyze the difference between the ideal loss and the loss with pseudo-labels:

$$\mathcal{L}_T(x_t, \mathcal{L}) = \sum_{k=1}^{K} p(y_t = k|x_t)\mathcal{L}(\mathbf{p}_t, k) \tag{2}$$

$$= \sum_{k=1}^{K}\sum_{l=1}^{K} p(y_t = k|\hat{y}_t = l, x_t)p(\hat{y}_t = l|x_t)\mathcal{L}(\mathbf{p}_t, k) \tag{3}$$

$$= \sum_{k=1}^{K}\sum_{l=1}^{K} \eta_{kl}^{(x_t)} p(\hat{y}_t = l|x_t)\mathcal{L}(\mathbf{p}_t, k), \tag{4}$$

where $\eta^{(x_t)}$ is the confusion matrix. The confusion matrix is unknown on the unlabeled target domain. For brevity, we define $\mathbf{c}_k^{(x_t)} = \sum_l \eta_{kl}^{(x_t)} p(\hat{y}_t = l|x_t)$ and name $\mathbf{c}^{(x_t)}$ as the corrected label vector.

In previous works studying noisy labels (Zhang and Sabuncu 2018), it is commonly assumed that the confusion matrix is conditionally independent of inputs x_t and *uniform* with noise rate α. The unhinged loss has been proved to be robust to the *uniform* noise (van Rooyen, Menon, and Williamson 2015; Ghosh, Kumar, and Sastry 2017),

$$\mathcal{L}_{unh}(\mathbf{p}, k) = 1 - \mathbf{p}_k. \tag{5}$$

However, these assumptions cannot hold in the case of pseudo-labels, which makes the problem more intractable.

Adversarial-Learned Loss

The general idea of our method is that if we can adequately estimate the noise matrix $\eta_{kl}^{(x_t)}$, the noise in pseudo-labels will be corrected and we can approximately optimize the ideal loss function on the target domain.

Firstly, to simplify the noisy label problem, we assume that the noise is class-wise uniform with vector $\xi^{(x_t)}$.

Definition 1. Noise is *class-wise uniform* with vector $\xi^{(x_t)} \in \mathbb{R}^K$, if $\eta_{kl}^{(x_t)} = \xi_k^{(x_t)}$ for $k = l$, and $\eta_{kl}^{(x_t)} = \frac{1-\xi_l^{(x_t)}}{K-1}$ for $k \neq l$.

In this work, we propose to use an extra neural network, called noise-correcting domain discriminator, to learn the vector $\xi^{(x_t)}$.

Noise-correcting Domain Discrimination

As shown in Fig. 2, the noise-correcting domain discriminator D is a multi-layer neural network, which takes the deep

feature $G(x)$ as the input and outputs a multi-class score vector $D(G(x)) \in \mathbb{R}^K$. After a sigmoid layer, the discriminator produces the noise vector $\xi^{(x)} = \sigma(D(G(x)))$. Each component of $\xi^{(x)}$ denotes the probability that the pseudo label is the same as the correct label: $\xi_k^{(x)} = p(y = k|\hat{y} = k, x)$.

We adopt the idea of the domain-adversarial learning (Ganin et al. 2016) that makes the discriminator and the generator play a minimax game. Instead of letting the discriminator perform a domain classification task, we let the discriminator generate different noise vectors for the source and target domains. As illustrated in Fig. 2, for the source feature $G(x_s)$, the discriminator aims to minimize the discrepancy between the corrected label vector $\mathbf{c}^{(x_s)}$ and the ground truth $\mathbf{y}_s (= \text{one\_hot}(y_s))$. The adversarial loss for the source data is:

$$\mathcal{L}_{Adv}(x_s, y_s) = \mathcal{L}_{BCE}(\mathbf{c}^{(x_s)}, \mathbf{y}_s) \tag{6}$$

$$= \sum_k -\mathbf{y}_{sk} \log \mathbf{c}_k^{(x_s)} - (1 - \mathbf{y}_{sk}) \log(1 - \mathbf{c}_k^{(x_s)}). \tag{7}$$

As for the target feature $G(x_t)$, the discriminator do the opposite way. The discriminator will correct pseudo-labels to the opposite distribution $\mathbf{u}^{(\hat{y}_t)} \in \mathbb{R}^K$, in which $\mathbf{u}_k^{(\hat{y}_t)} = 0$ for $k = \hat{y}_t$ and $\mathbf{u}_k^{(\hat{y}_t)} = \frac{1}{(K-1)}$ for $k \neq \hat{y}_t$. The adversarial loss for the target data is:

$$\mathcal{L}_{Adv}(x_t) = \mathcal{L}_{BCE}(\mathbf{c}^{(x_t)}, \mathbf{u}^{(\hat{y}_t)}). \tag{8}$$

The total adversarial loss becomes:

$$\mathcal{L}_{Adv}(x_s, y_s, x_t) = \mathcal{L}_{Adv}(x_s, y_s) + \mathcal{L}_{Adv}(x_t). \tag{9}$$

The discriminator D needs to minimize the loss function to distinguish between the source and target feature. On the other hand, the generator G has to fool the discriminator, by maximizing the above loss function. Compared to the common domain-adversarial learning, this adversarial loss takes the classifier prediction and the label information into consideration. In this way, our noise-correcting domain discriminator can achieve the class-wise feature alignment.

Regularization Term

As revealed in the works of generative adversarial networks (GANs) (Mao et al. 2017), the training process of adversarial learning can be unstable. Following (Odena, Olah, and Shlens 2016), we add a classification task on the source domain to the discriminator to make its training more stable. Consequently, the discriminator not only has to distinguish the source and target domains but also correctly classify the source samples.

To embed the classification task into training, we add a regularization term to the loss of the discriminator:

$$\mathcal{L}_{Reg}(x_s, y_s) = \mathcal{L}_{CE}(\mathbf{p}_D^{(x_s)}, y_s), \tag{10}$$

where $\mathbf{p}_D^{(x_s)} = \text{softmax}(D(G(x_s)))$ and $\mathcal{L}_{CE}$ is the cross entropy loss. Then the final loss function for the discriminator becomes:

$$\min_D E_{(x_s,y_s),x_t}(\mathcal{L}_{Adv}(x_s, y_s, x_t) + \mathcal{L}_{Reg}(x_s, y_s)). \tag{11}$$

Corrected Loss Function

After the adversarial learning of the confusion matrix $\eta^{(x_t)}$, we can construct a proper loss function for the target samples. As the unhinged loss (Eq. 5) is robust to the uniform part of noise, we choose the unhinged loss $\mathcal{L}_{unh}$ as the basic loss function $\mathcal{L}$:

$$\mathcal{L}_T(x_t, \mathcal{L}_{unh}) = \sum_{k,l} \eta_{kl}^{(x_t)} p(\hat{y}_t = l|x_t) \mathcal{L}_{unh}(\mathbf{p}_t, k) \tag{12}$$

$$= \sum_k \mathbf{c}_k^{(x_t)} \mathcal{L}_{unh}(\mathbf{p}_t, k). \tag{13}$$

Together with the supervised loss on the source domain, the losses for the classifier and the generator become:

$$\min_C E_{(x_s,y_s),x_t}(\mathcal{L}_{CE}(p_s, y_s) + \lambda \mathcal{L}_T(x_t, \mathcal{L}_{unh})) \tag{14}$$

$$\min_G E_{(x_s,y_s),x_t}(\mathcal{L}_{CE}(p_s, y_s) + \lambda \mathcal{L}_T(x_t, \mathcal{L}_{unh}) - \lambda \mathcal{L}_{Adv}(x_s, y_s, x_t)), \tag{15}$$

where $\lambda \in [0, 1]$ is a trade-off parameter.

Theoretical Insight

In the feature space $\mathcal{F}$ generated by the generator G, the source and target feature distributions are $\mathcal{P}_s = \{G(x_s)|x_s \in \mathcal{D}_s\}$ and $\mathcal{P}_t = \{G(x_t)|x_t \in \mathcal{D}_t\}$ respectively. If we assume that both distributions are continuous with densities P_s and P_t, for a feature vector $f \in \mathcal{F}$, the probabilities that it belongs to source and target distributions are $P_s(f)$ and $P_t(f)$ respectively.

Theorem 1. *When the noise-correcting domain discrimination*

$$\max_G \min_D E_{(x_s,y_s),x_t} \mathcal{L}_{Adv}(x_s, y_s, x_t) \tag{16}$$

achieves the optimal point D^* and G^*, the feature distributions generated by G^* are aligned: $\mathcal{P}_s = \mathcal{P}_t$.

Proof. The proof is given in the supplemental material.

As a result, the noise-correcting domain discrimination can align the feature distribution between the source and target domain. According to the theory of (Ben-David et al. 2010), the expected error on the target samples can be bounded by the expected error on the source domain and feature discrepancy between domains. Therefore, the target expected error of our noise-correcting domain discrimination is theoretically bounded.

Furthermore, we can prove that by optimizing the corrected loss function, the noise in pseudo-labels is reduced.

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
ResNet-50 (He et al. 2016)	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DANN (Ganin et al. 2016)	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
ADDA (Tzeng et al. 2017)	86.2±0.5	96.2±0.3	98.4±0.3	77.8±0.3	69.5±0.4	68.9±0.5	82.9
JAN (Long et al. 2017b)	85.4±0.3	97.4±0.2	99.8±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
MADA (Pei et al. 2018)	90.0±0.1	97.4±0.1	99.6±0.1	87.8±0.2	70.3±0.3	66.4±0.3	85.2
CBST (Zou et al. 2018)	87.8±0.8	98.5±0.1	**100±0.0**	86.5±1.0	71.2±0.4	70.9±0.7	85.8
CAN (Zhang et al. 2018)	92.5	**98.8**	**100.0**	90.1	72.1	69.9	87.2
CDAN+E (Long et al. 2017a)	94.1±0.1	98.6±0.1	**100.0±0.0**	92.9±0.2	71.0±0.3	69.3±0.3	87.7
MCS (Liang et al. 2019)	-	-	-	-	-	-	87.8
ALDA	**95.6±0.5**	97.7±0.1	**100.0±0.0**	**94.0±0.4**	**72.2±0.4**	**72.5±0.2**	**88.7**

Table 1: Accuracy (%) of different unsupervised domain adaptation methods on Office-31 (ResNet-50)

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
ResNet-50 (He et al. 2016)	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
DANN (Ganin et al. 2016)	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 (Long et al. 2017b)	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
CDAN+E (Long et al. 2017a)	50.7	**70.6**	76.0	57.6	70.0	70.0	57.4	50.9	**77.3**	**70.9**	**56.7**	81.6	65.8
TAT (Liu et al. 2019)	51.6	69.5	75.4	59.4	69.5	68.6	**59.5**	50.5	76.8	**70.9**	56.6	81.6	65.8
ALDA	**53.7**	70.1	**76.4**	**60.2**	**72.6**	**71.5**	56.8	**51.9**	77.1	70.2	56.3	82.1	**66.6**

Table 2: Accuracy (%) of different unsupervised domain adaptation methods on Office-Home (ResNet-50)

Theorem 2. *When the optimal point D^* and G^* are achieved in Theorem 1, if there is a optimal labeling function $y^*(f_s) = y_s, \forall f_s \in \mathcal{P}_s$ in the feature space $\mathcal{F}$, then $\forall x_t \in \mathcal{P}_t$ and $f_t = G^*(x_t)$, we have:*

$$\mathbf{c}^{(x_t)} = \begin{cases} \mathbf{h}^{y^*(f_t)} & \hat{y}_t = y^*(f_t) \\ \mathbf{u}^{(\hat{y}_t)} & \textit{otherwise} \end{cases},$$

where $\mathbf{c}^{(x_t)} = \mathbf{h}^{y^*(f_t)}$ denotes that $\mathbf{c}_k^{(x_t)} = \frac{1}{2}$ for $k = \hat{y}_t$ and $\mathbf{c}_k^{(x_t)} = \frac{1}{2K-2}$ otherwise.

Proof. The proof is given in the supplemental material.

As Theorem 2 shows, when we optimize the target loss $\mathcal{L}_T(x_t, \mathcal{L}) = \sum_k \mathbf{c}_k^{(x_t)} \mathcal{L}(p_t, k)$, the loss of pseudo-labels $\mathcal{L}(p_t, \hat{y}_t)$ will be enhanced when $\hat{y}_t = y^*(x_t)$ ($\mathbf{c}_{\hat{y}_t}^{(x_t)} = \frac{1}{2}$) and suppressed otherwise ($\mathbf{c}_{\hat{y}_t}^{(x_t)} = 0$). In this way, the training of classifier can be corrected by the discriminator on the target domain and will be more efficient than the original pseudo-label method.

Experiments

We evaluate the proposed adversarial-learned loss for domain adaptation (ALDA) with state-of-the-art approaches on four standard unsupervised domain adaptation datasets: digits, office-31, office-home, and VisDA-2017.

Datasets

Digits. Following the evaluation protocol of (Long et al. 2017a), we experiment on three adaptation scenarios: USPS to MNIST (**U** → **M**), MNIST to USPS (**M** → **U**), and SVHN to MNIST (**S** → **M**). MNIST (LeCun 1998) contains $60,000$ images of handwritten digits and USPS (Hull 1994) contains $7,291$ images. Street View House Numbers (SVHN) (Netzer et al. 2011) consists of $73,257$ images with digits and numbers in natural scenes. We report the evaluation results on the test sets of MNIST and USPS.

Office-31 (Saenko and Kulis 2010) is a commonly used dataset for unsupervised domain adaptation, which contains $4,652$ images and 13 categories collected from three domains: *Amazon* (**A**), *Webcam* (**W**) and *DSLR* (**D**). We evaluate all methods across six domain adaptation tasks: **A** → **W**, **D** → **W**, **W** → **D**, **A** → **D**, **D** → **A** and **W** → **A**.

Office-Home (Venkateswara et al. 2017) is a more difficult domain adaptation dataset than office-31, including $15,500$ images from four different domains: Artistic images (**Ar**), Clip Art (**Cl**), Product images (**Pr**) and Real-World (**Rw**). For each domain, the dataset contains images of 65 object categories that are common in office and home scenarios. We evaluate all methods in 12 adaptation scenarios.

VisDA-2017 (Peng et al. 2017) is a large-scale dataset and challenge for unsupervised domain adaptation from simulation to real. The dataset contains $152,397$ synthetic images as the source domain and $55,388$ real-world images as the target domain. 12 object categories are shared by these two domains. Following previous works (Saito et al. 2018; Long et al. 2017a), we evaluate all methods on the validation set of VisDA.

Setup

For digits datasets, we adopt the generator and classifier networks used in (French, Mackiewicz, and Fisher 2018) and optimize the model using Adam (Kingma and Ba 2015) gradient descent with learning rate 1×10^{-3}.

For the other three datasets, we employ ResNet-50 (He et al. 2016) as the generator network. The ResNet-50 is pretrained on ImageNet (Deng et al. 2009). Our discriminator consists of three fully connected layers with dropout,

Method	Backbone	plane	bcycl	bus	car	house	knife	mcycl	person	plant	sktbrd	train	truck	Avg
Sourceonly		55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DANN (Ganin et al. 2016)		81.9	**77.7**	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
MCD (Saito et al. 2018)	ResNet101	87.0	60.9	**83.7**	64.0	88.9	79.6	84.7	**76.9**	88.6	40.3	83.0	25.8	71.9
CBST (Zou et al. 2018)		87.2	78.8	56.5	55.4	85.1	79.2	83.8	77.7	82.8	**88.8**	69.0	**72.0**	76.4
ALDA		**93.8**	74.1	82.4	**69.4**	**90.6**	87.2	89.0	67.6	93.4	76.1	87.7	22.2	**77.8**
Sourceonly		74.6	26.8	56.0	53.5	58.0	26.2	76.5	17.6	81.7	34.8	80.3	27.2	51.1
CDAN+E (Long et al. 2017a)	ResNet50	-	-	-	-	-	-	-	-	-	-	-	-	70.0
ALDA		87.0	61.3	78.7	67.9	83.7	**89.4**	**89.5**	71.0	**95.4**	71.9	**89.6**	33.1	**76.5**

Table 3: Accuracy (%) of different unsupervised domain adaptation methods on VisDA-2017.

Method	U → M	M → U	S → M	Avg
Sourceonly	77.5±0.8	82.0±1.2	66.5±1.9	75.3
DANN (Ganin et al. 2016)	74.0	91.1	73.9	79.7
ADDA (Tzeng et al. 2017)	90.1	89.4	76.0	85.2
CDAN+E (Long et al. 2017a)	98.0	95.6	89.2	94.3
MT+CT	92.3±8.6	88.1±0.34	93.3±5.8	91.2
(French, Mackiewicz, and Fisher 2018)				
MCD (Saito et al. 2018)	94.1±0.3	96.5±0.3	96.2±0.4	95.6
MCS (Liang et al. 2019)	98.2	**97.8**	91.7	95.9
ALDA ($\delta = 0.9$)	98.1±0.2	94.8±0.1	95.6±0.6	96.2
ALDA ($\delta = 0.8$)	98.2±0.1	95.4±0.4	97.5±0.3	97.0
ALDA ($\delta = 0.6$)	**98.6±0.1**	95.6±0.3	**98.7±0.2**	**97.6**
ALDA ($\delta = 0.0$)	98.4±0.2	95.0±0.1	97.0±0.2	96.8
Targetonly	99.5±0.0	97.3±0.2	99.6±0.1	98.8

Table 4: Accuracy (%) of different unsupervised domain adaptation methods on the digits datasets. We use the base model in (French, Mackiewicz, and Fisher 2018).

which is the same as other works (Ganin et al. 2016; Long et al. 2017a). As we train the classifier and discriminator from scratch, we set their learning rates to be 10 times that of the generator. We train the model with Stochastic Gradient Descent (SGD) optimizer with the momentum of 0.9. We schedule the learning rate with the strategy in (Ganin et al. 2016): the learning rate is adjusted by $\eta_p = \frac{\eta_0}{(1+\alpha q)^\beta}$, where q is the training progress linearly changing from 0 to 1, $\eta_0 = 0.01$, $\alpha = 10$, $\beta = 0.75$. We implement the algorithms using **PyTorch** (Paszke et al. 2017).

There are two hyper-parameters in our method: the threshold δ of pseudo-labels and the trade-off λ. If the prediction of a target sample is below the threshold, we ignore these samples in training. We set δ to 0.6 for digit adaptation and 0.9 for office-31, office-home datasets and VisDA dataset. In all experiment, λ is gradually increased from 0 to 1 by $\frac{2}{1+\exp(-10 \cdot q)} - 1$, same as (Long et al. 2017a).

Result

Image Results. Table 1 reports the results with ResNet-50 on Office-31. ALDA significantly outperforms state-of-the-art methods. Because ALDA combines with self-training methods to learn discriminative features, ALDA achieves better results than the domain-adversarial learning-based methods, e.g., DANN, JAN, MADA. Similar to ALDA, CDAN+E also takes the classification prediction into the discrimination and uses the entropy of prediction as an importance weight. However, ALDA outperforms CDAN+E on hard transfer tasks, e.g., A → W, A → D, D → A and W → A. The outstanding results show that it is important to combine the domain-adversarial learning and self-training based methods properly.

Table 2 summarizes the results with ResNet-50 on Office-home. For these more difficult adaptation datasets, ALDA still exceeds the most advanced methods. Compared to Office-31, Office-Home has more categories and has a larger appearance gap between domains. A larger number of categories indicates more components of the discriminator output ξ in ALDA, which results in a stronger capacity of class-wise domain discrimination.

Table 3 shows the quantitative results with ResNet-50 and ResNet-101 on VisDA classification dataset. Even though only based on ResNet-50, our ALDA performs better than other domain adaptation methods.

Digits Results. Table 4 summarizes the experimental results for digits adaption comparing with state-of-the-art methods. For fair comparisons, we only resize and normalize the image and do not apply any addition data augment like (French, Mackiewicz, and Fisher 2018). We conduct each experiment three times and report their average results and variance. As the table shows, ALDA outperforms the most advanced distribution alignment methods, e.g., DANN, MCD, CDAN, and self-training based methods, e.g., Mean Teacher with a confident threshold (MT+CT). ALDA also reduces the performance gap between UDA and the supervised learning on the target domain by a large margin.

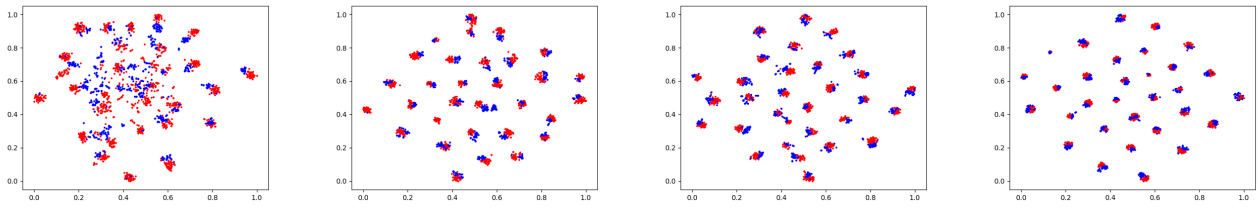
In Table 4, we also investigate the effect of the threshold δ for pseudo-labels on the digits datasets. As we decrease the threshold δ from 0.9 to 0.6, the performances are improved. It is because the digits datasets are relatively easy to transfer and do not require high thresholds to obtain high precision pseudo-labels. The lower threshold will take more target samples into training, which promotes the training of samples with low prediction confidence. For the digits datasets, ALDA with $\delta = 0.6$ achieves the best result.

Analysis

In Table 5, we perform an ablation study on Office-31 to investigate the effect of different components in ALDA. Firstly, we apply self-training (Zou et al. 2018) to unsupervised domain adaptation, which is denoted as "ST". "DANN+ST" denotes that we directly combine the domain-adversarial learning and the self-training methods. However, the performance of "DANN+ST" is inferior to "ALDA", proving the importance of properly combining these two

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
ResNet-50 (He et al. 2016)	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DANN (Ganin et al. 2016)	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
ST	89.0	**99.0**	100.0	86.3	67.5	63.0	84.1
DANN + ST	91.8	98.4	100.0	89.1	68.8	68.7	86.1
ALDA w/o $\mathcal{L}_{Reg}$	93.8	98.7	100.0	91.5	70.4	67.3	87.0
ALDA w/o $\mathcal{L}_T$	95.0	97.5	100.0	94.0	70.8	69.0	87.7
ALDA+ST w/o $\mathcal{L}_T$	94.8	98.0	100.0	**95.4**	71.0	65.9	87.8
ALDA w/ $\mathcal{L}_T(x, \mathcal{L}_{CE})$	95.1	97.6	100.0	92.7	69.4	70.5	87.6
ALDA	**95.6±0.5**	97.7±0.1	**100.0±0.0**	94.0±0.4	**72.2±0.4**	**72.5±0.2**	**88.7**

Table 5: Ablation study on Office-31 (ResNet-50). "ST" denotes self-training with pseudo-labels (Zou et al. 2018).



(a) ResNet-50 (b) Self-training (c) DANN (d) ALDA

Figure 3: T-SNE of (a) ResNet-50, (b) Self-training, (c) DANN, (d) ALDA for A → W adaptation(red: A; blue: W).

methods. To investigate the effect of the regularization term $\mathcal{L}_{Reg}$ in Eq. 10, we remove the $\mathcal{L}_{Reg}$ term in the final loss of the discriminator, denoted as "ALDA w/o $\mathcal{L}_{Reg}$". The results show that without $\mathcal{L}_{Reg}$, the performance of ALDA drops dramatically. This phenomenon is because the regularization term can enhance the stability of the adversarial process.

To investigate the effect of the corrected target loss $\mathcal{L}_T$ in Eq.13, we remove the $\mathcal{L}_T$ and only keep the noise-correcting domain discrimination, denotes as "ALDA w/o $\mathcal{L}_T$". As Table 5 shows, "ALDA w/o $\mathcal{L}_T$" can achieve competitive results but inferior to "ALDA". The phenomenon shows the superiority of our noise-correcting domain discrimination and the importance of combining domain discrimination and corrected pseudo-labels to enhance the performance. Additionally, we replace the corrected target loss $\mathcal{L}_T$ with uncorrected target loss, i.e., self-training with pseudo-labels, which is denoted as "ALDA+ST w/o $\mathcal{L}_T$". However, "ALDA+ST w/o $\mathcal{L}_T$" does not improve the performance, which manifests the importance of correcting pseudo-labels.

As mentioned before, the unhinged loss has been proved to be robust to the uniform part of the noise. To verify the effect of choosing the unhinged loss $\mathcal{L}_{unh}$ as basic loss function, we substitute the unhinged loss with the cross-entropy loss $\mathcal{L}_{CE}$ in the target loss $\mathcal{L}_T(x, \mathcal{L})$, denoted as "ALDA w/ $\mathcal{L}_T(x, \mathcal{L}_{CE})$". The results in Table 5 demonstrate that the cross-entropy loss performs worse than the unhinged loss in ALDA. The unhinged loss can remove the uniform part of the noise, which facilitates the noise-correcting process.

Visualization

We use t-SNE (van der Maaten and Hinton 2008) to visualize the feature extracted by ResNet-50, Self-training, DANN and ALDA for A → W adaptation (31 classes) in Fig. 3. When using ResNet-50 only, the target feature distribution is not aligned with the source. Although self-training and DANN can align the distributions of the source and target domain, their target clusters are not fully matched with source clusters. For ALDA, the target clusters are closely matched with the corresponding source clusters, which demonstrates the target features extracted by ALDA are well aligned and discriminative.

Conclusion

In this paper, we propose Adversarial-Learned Loss for Domain Adaptation (ALDA) to combine the strengths of domain-adversarial learning and self-training. We first introduce the confusion matrix to represent the noise in pseudo-labels. As the confusion matrix is unknown, we employ noise-correcting domain discrimination to learn the confusion matrix. Then the target classifier is optimized with the corrected loss function. Our ALDA is theoretically and experimentally proven to be effective for unsupervised domain adaption and achieves state-of-the-art performance on four standard datasets.

Acknowledgments

This work was supported in part by The National Key Research and Development Program of China (Grant Nos: 2018AAA0101400), in part by The National Nature Science Foundation of China (Grant Nos: 61936006, 61973271).

References

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A theory of learning from different domains. *Machine Learning* 79(1):151–175.

Che, T.; Li, Y.; Jacob, A. P.; Bengio, Y.; and Li, W. 2017. Mode regularized generative adversarial. In *ICLR*.

Chen, M.; Xue, H.; and Cai, D. 2019. Domain adaptation for semantic segmentation with maximum squares loss. In *The IEEE International Conference on Computer Vision (ICCV)*.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Li, F. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

French, G.; Mackiewicz, M.; and Fisher, M. 2018. Self-ensembling for visual domain adaptation. In *ICLR*.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. S. 2016. Domain-adversarial training of neural networks. *JMLR* 17:20962030.

Ghosh, A.; Kumar, H.; and Sastry, P. S. 2017. Robust loss functions under label noise for deep neural networks. In *AAAI*.

Grandvalet, Y., and Bengio, Y. 2004. Semi-supervised learning by entropy minimization. In *NIPS*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hong, W.; Wang, Z.; Yang, M.; and Yuan, J. 2018. Conditional generative adversarial network for structured domain adaptation. In *CVPR*.

Hull, J. J. 1994. A database for handwritten text recognition research. *PAMI* 16:550–554.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

LeCun, Y. 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, 22782324.

Lee, D.-H. 2013. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. In *ICML*.

Liang, J.; He, R.; Sun, Z.; and Tan, T. 2019. Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation. In *CVPR*.

Liu, H.; Long, M.; Wang, J.; and Jordan, M. 2019. Transferable adversarial training: A general approach to adapting deep classifiers. In *ICML*.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. I. 2015. Learning transferable features with deep adaptation networks. In *ICML*.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2017a. Conditional adversarial domain adaptation. In *NeurIPS*.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2017b. Deep transfer learning with joint adaptation networks. In *ICML*.

Mao, X.; Li, Q.; Xie, H.; Lau, R. Y. K.; Wang, Z.; and Smolley, S. P. 2017. Least squares generative adversarial networks. In *ICCV*.

Netzer, Y.; Fillet, M.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS*.

Odena, A.; Olah, C.; and Shlens, J. 2016. Conditional image synthesis with auxiliary classifier gans. In *ICML*.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *TKDE* 22(10):1345–1359.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Pei, Z.; Cao, Z.; Long, M.; and Wang, J. 2018. Multi-adversarial domain adaptation. In *AAAI*.

Peng, X.; Usman, B.; Kaushik, N.; Hoffman, J.; Wang, D.; and Saenko, K. 2017. Visda: The visual domain adaptation challenge. *ArXiv* abs/1710.06924.

Saenko, K., and Kulis, B. 2010. Adapting visual category models to new domains. In *ECCV*.

Saito, K.; Watanabe, K.; Ushiku, Y.; and Harada, T. 2018. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*.

Saito, K.; Kim, D.; Sclaroff, S.; Darrell, T.; and Saenko, K. 2019. Semi-supervised domain adaptation via minimax entropy. *ArXiv* abs/1904.06487.

Sukhbaatar, S., and Fergus, R. 2014. Learning from noisy labels with deep neural networks. In *ICLR*.

Sun, B., and Saenko, K. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV Workshops*.

Tarvainen, A., and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *ICLR*.

Torralba, A., and Efros, A. A. 2011. Unbiased look at dataset bias. In *CVPR*.

Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; and Darrell, T. 2014. Deep domain confusion: Maximizing for domain invariance. *CoRR* abs/1412.3474.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial discriminative domain adaptation. In *CVPR*.

van der Maaten, L., and Hinton, G. E. 2008. Visualizing data using t-sne. In *JMLR*.

van Rooyen, B.; Menon, A. K.; and Williamson, R. C. 2015. Learning with symmetric label noise: The importance of being unhinged. In *NIPS*.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep hashing network for unsupervised domain adaptation. In *CVPR*.

Xie, S.; Zheng, Z.; Chen, L.; and Chen, C. 2018. Learning semantic representations for unsupervised domain adaptation. In *ICML*.

Zhang, Z., and Sabuncu, M. R. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NIPS*.

Zhang, W.; Ouyang, W.; Li, W.; and Xu, D. 2018. Collaborative and adversarial network for unsupervised domain adaptation. In *CVPR*.

Zou, Y.; Yu, Z.; Kumar, B. V. K. V.; and Wang, J. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*.

Adversarial-Learned Loss for Domain Adaptation -Supplementary Material-

Paper ID: 1081

Theoretical Proof

In the feature space $\mathcal{F}$ generated by the generator G, the source and target feature distributions are $\mathcal{P}_s = \{G(x_s)|x_s \in \mathcal{D}_s\}$ and $\mathcal{P}_t = \{G(x_t)|x_t \in \mathcal{D}_t\}$ respectively. If we assume that both distributions are continuous with densities P_s and P_t, for a feature vector $f \in \mathcal{F}$, the probabilities that it belongs to source and target distributions are $P_s(f)$ and $P_t(f)$ respectively. We assume that there are an ideal classification function y^* such that $y^*(f_s) = y_s, \forall f_s \in \mathcal{P}_s$. To continuously expand the function y^* to the whole feature space $\mathcal{F}$, we define the function P_{y^*} :

$$\begin{aligned} &P_{y^*}(f) \in \mathbb{R}^K, f \in \mathcal{F} \\ &\text{s.t. } \forall f_s \in \mathcal{P}_s, P_{y^*}(f_s)_k = 1 \text{ for } k = y_s \\ &\qquad \text{and } P_{y^*}(f_s)_k = 0 \text{ for } k \neq y_s. \end{aligned}$$

Similarly, we can define the function $P_{\hat{y}}$ for the classifier predictions $\hat{y}$:

$$\begin{aligned} &P_{\hat{y}}(f) \in \mathbb{R}^K, f \in \mathcal{F} \\ &\text{s.t. } \forall f \in \mathcal{P}_s \cup \mathcal{P}_t, P_{\hat{y}}(f)_k = 1 \text{ for } k = \hat{y} \\ &\qquad \text{and } P_{\hat{y}}(f)_k = 0 \text{ for } k \neq \hat{y}. \end{aligned}$$

To prove the Theorem 1, we need to prove a lemma firstly.

Lemma 1.

$$\forall x_t, \mathcal{L}_{BCE}(\mathbf{c}^{(x_t)}, \mathbf{u}^{(\hat{y}_t)}) = -2\log(1 - \xi_{\hat{y}_t}^{(x_t)}) + C \quad (1)$$

$$\forall x_s, \mathcal{L}_{BCE}(\mathbf{c}^{(x_s)}, \mathbf{y}_s) = -2\log \xi_{y_s}^{(x_t)} + C \quad (2)$$

where C is a constant term that only depends on the class number K.

Proof. We first prove Eq. 1 and the proof of Eq. 2 is similarly available. As the definition, $k \in \{1, .., K\}, \mathbf{c}_k^{(x_t)} = \sum_l \eta_{kl}^{(x_t)} \hat{\mathbf{y}}_l$, where $\hat{\mathbf{y}}$ is the one-hot form of $\hat{y}$: $\hat{\mathbf{y}}_k = 1$ for $k = \hat{y}$ and $\hat{\mathbf{y}}_k = 0$ for $k \neq \hat{y}$. Therefore, $\mathbf{c}_k^{(x_t)} = \eta_{k\hat{y}}^{(x_t)}$. Because $\eta^{(x_t)}$ is *class-wise uniform* with vector $\xi^{(x_t)}$, $\mathbf{c}_k^{(x_t)} = \xi_{\hat{y}_t}^{(x_t)}$ for $k = \hat{y}$ and $\mathbf{c}_k^{(x_t)} = \frac{1-\xi_{\hat{y}_t}^{(x_t)}}{K-1}$ for $k \neq \hat{y}$.

Then

$$\begin{aligned} &\mathcal{L}_{BCE}(\mathbf{c}^{(x_t)}, \mathbf{u}^{(\hat{y}_t)}) \\ &= \sum_k -\mathbf{u}_k^{(\hat{y}_t)} \log \mathbf{c}_k^{(x_t)} - (1 - \mathbf{u}_k^{(\hat{y}_t)}) \log(1 - \mathbf{c}_k^{(x_t)}) \\ &= -\log(1 - \xi_{\hat{y}_t}^{(x_t)}) - \sum_{k \neq \hat{y}_t} \frac{1}{K-1} \log(1 - \xi_k^{(x_t)}) + (K-1) \\ &= -2\log(1 - \xi_{\hat{y}_t}^{(x_t)}) + (K-1) \end{aligned}$$

Together with the above lemma, we can prove the theorems in the section of "Theoretical Insight".

Theorem 1. *When the noise-correcting domain discrimination*

$$\max_G \min_D E_{(x_s,y_s),x_t} \mathcal{L}_{Adv}(x_s, y_s, x_t) \quad (3)$$

achieves the optimal point D^* and G^*, the feature distributions generated by G^* are aligned: $\mathcal{P}_s = \mathcal{P}_t$.

Proof. We assume that there are expand functions P_{y^*} and $P_{\hat{y}}$ as defined before. Then, we can expand the adversarial loss as:

$$\begin{aligned} &E_{(x_s,y_s),x_t} \mathcal{L}_{Adv}(x_s, y_s, x_t) \\ &= E_{(x_s,y_s)\in\mathcal{D}_s} \mathcal{L}_{BCE}(\mathbf{c}^{(x_s)}, \mathbf{y}_s) + E_{x_t \in \mathcal{D}_t} \mathcal{L}_{BCE}(\mathbf{c}^{(x_t)}, \mathbf{u}^{(\hat{y}_t)}) \\ &= E_{f\in\mathcal{P}_s} \mathcal{L}_{BCE}(\mathbf{c}^{(f)}, \mathbf{y}^*(f)) + E_{f\in\mathcal{P}_t} \mathcal{L}_{BCE}(\mathbf{c}^{(f)}, \mathbf{u}^{(\hat{y}(f))}), \end{aligned}$$

where in order to transfer the expression to the feature space, we modify the notations: $\xi_k^{(f)} := \xi_k^{(x)}$, $\eta_{kl}^{(f)} := \eta_{kl}^{(x)}$, and $\mathbf{c}_k^{(f)} := \sum_l \eta_{kl}^{(f)} \hat{\mathbf{y}}(f)_l$,

For a feature vector $f \in \mathcal{P}_s \cup \mathcal{P}_t$, its contribution to the adversarial loss is:

$$\mathcal{L}(f) = P_s(f)\sum_{k=1}^{K} P_{y^*}(f)_k \mathcal{L}_{BCE}(\mathbf{c}^{(f)}, \mathbf{k}) + P_t(f)\sum_{k=1}^{K} P_{\hat{y}}(f)_k \mathcal{L}_{BCE}(\mathbf{c}^{(f)}, \mathbf{u}^{(k)})$$
$$= -2P_s(f)\sum_{k=1}^{K} P_{y^*}(f)_k \log(\xi_k^{(f)}) - 2P_t(f)\sum_{k=1}^{K} P_{\hat{y}}(f)_k \log(1-\xi_k^{(f)}) + C$$

The above equation is derived from Lemma 1.

We take the extremum of the noise vector ξ for the above formula.

$$\forall k \in \{1,..,K\}, \frac{\partial \mathcal{L}(f)}{\partial \xi_k^{(f)}}$$
$$= -2P_s(f)P_{y^*}(f)_k \frac{1}{\xi_k^{(f)}} - 2P_t(f)P_{\hat{y}}(f)_k \frac{1}{1-\xi_k^{(f)}} = 0$$
$$\Longrightarrow P_s(f)P_{y^*}(f)_k(1-\xi_k^{(f)}) = P_t(f)P_{\hat{y}}(f)_k \xi_k^{(f)}$$
$$\Longrightarrow \xi_k^{(f)} = \frac{P_s(f)P_{y^*}(f)_k}{P_s(f)P_{y^*}(f)_k + P_t(f)P_{\hat{y}}(f)_k}$$

The above equation is optimal point of the discriminator D^*. When D^* can not distinguish the source and target features, we can get the optimal generator G^*. We take the above $\xi^{(f)}$ back into the formula of $\mathcal{L}(f)$ and compute the extremum of it. Because the induction process is too complicate to write down, we directly show the maximum point:

$$P_s(f) = P_t(f),\ \forall f \in \mathcal{P}_s \cup \mathcal{P}_t$$

Therefore, when the optimal D^* and G^* are achieved, $\mathcal{P}_s = \mathcal{P}_t$.

Theorem 2. *When the optimal point D^* and G^* are achieved in Theorem 1, if there is a optimal labeling function $y^*(f_s) = y_s, \forall f_s \in \mathcal{P}_s$ in the feature space $\mathcal{F}$, then $\forall x_t \in \mathcal{P}_t$ and $f_t = G^*(x_t)$, we have:*

$$\mathbf{c}^{(x_t)} = \begin{cases} \mathbf{h}^{y^*(f_t)} & \hat{y}_t = y^*(f_t) \\ \mathbf{u}^{(\hat{y}_t)} & otherwise \end{cases},$$

Proof. The optimal D^* and G^* are presented in the proof of Theorem 1.

$\forall x_t \in \mathcal{P}_t$, we have the discriminator outputs as:

$$\forall k \in \{1,..,K\}, \xi_k^{(x_t)}$$
$$= \frac{P_s(G^*(x_t))P_{y^*}(G^*(x_t))_k}{P_s(G^*(x_t))P_{y^*}(G^*(x_t))_k + P_t(G^*(x_t))P_{\hat{y}}(G^*(x_t))_k}$$
$$= \frac{P_{y^*}(G^*(x_t))_k}{P_{y^*}(G^*(x_t))_k + P_{\hat{y}}(G^*(x_t))_k}$$

In the case of $\hat{y}_t = y^*(x_t)$: $\xi_k^{(x_t)} = \frac{1}{2}, \forall k \in \{1,..,K\}$. Therefore, $\mathbf{c}_k^{(x_t)} = \frac{1}{2}$ for $k = \hat{y}_t$ and $\mathbf{c}_k^{(x_t)} = \frac{1}{2K-2}$ for $k \neq \hat{y}_t$. That is $\mathbf{c}^{(x_t)} = \mathbf{h}^{y^*(f_t)}$

In the case of $\hat{y}_t \neq y^*(f_t)$: $\xi_k^{(x_t)} = 1$ for $k = \hat{y}_t$ or $y^*(f_t)$, and $\xi_k^{(x_t)} = 0$ otherwise. Therefore, $\mathbf{c}_k^{(x_t)} = 0$ for $k = \hat{y}_t$ and $\mathbf{c}_k^{(x_t)} = \frac{1}{K}$ for $k \neq \hat{y}_t$. That is $\mathbf{c}^{(x_t)} = \mathbf{u}^{(\hat{y}_t)}$.

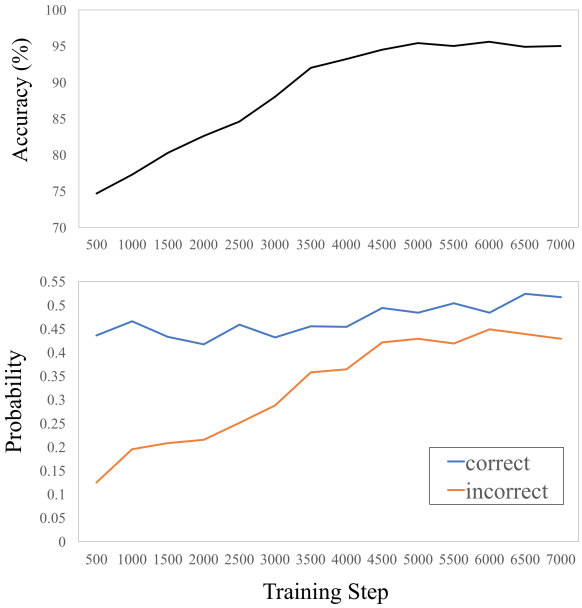


Figure 1: The upper figure: the training accuracy during training ALDA on $A \rightarrow W$. The bottom figure: the records of the weight of pseudo-labels: $\mathbf{c}_{\hat{y}_t}^{(x_t)}$ for correct samples and incorrect samples respectively.

Experimental Analysis

To show the mechanism of the corrected target loss: $\mathcal{L}_T(x_t, \mathcal{L}) = \sum_k \mathbf{c}_k^{(x_t)} \mathcal{L}(p_t, k)$, we conduct the following experiment on the $A \rightarrow W$ adaptation with ResNet-50. During the training process of ALDA, we record the weight of pseudo-labels: $\mathbf{c}_{\hat{y}_t}^{(x_t)}$, which estimates the probability $p(y_t = \hat{y}_t | \hat{y}_t, x_t)$, for correct samples ($y_t = \hat{y}_t$) and incorrect samples ($y_t \neq \hat{y}_t$) respectively. Theoretically, Theorem 2 shows that for correct samples, we should assign large weight $\mathbf{c}_{\hat{y}_t}^{(x_t)} = \frac{1}{2}$, while for incorrect samples, we should assign small weight $\mathbf{c}_{\hat{y}_t}^{(x_t)} = 0$.

As presented in Fig.1, at the beginning of training, the weight of pseudo-labels: $\mathbf{c}_{\hat{y}_t}^{(x_t)}$ is high for correct samples ($\mathbf{c}_{\hat{y}_t}^{(x_t)} \approx 0.5$) and low for incorrect samples ($\mathbf{c}_{\hat{y}_t}^{(x_t)} \approx 0.1$). That is because, in the beginning, the correct samples are those samples that are easy to transfer and can be well matched with the source distribution by the discriminator. Consequently, the loss of pseudo-labels $\mathcal{L}(p_t, \hat{y}_t)$ will be boosted for the correct samples and surpassed for the incorrect samples, as indicated by the theorem. However, after