

# Learning From a Complementary-Label Source Domain: Theory and Algorithms

Yiyang Zhang, Feng Liu<sup>ID</sup>, Member, IEEE, Zhen Fang, Student Member, IEEE,  
Bo Yuan<sup>ID</sup>, Senior Member, IEEE, Guangquan Zhang<sup>ID</sup>, and Jie Lu<sup>ID</sup>, Fellow, IEEE

**Abstract**—In unsupervised domain adaptation (UDA), a classifier for the target domain is trained with massive *true-label* data from the source domain and unlabeled data from the target domain. However, collecting true-label data in the source domain can be expensive and sometimes impractical. Compared to the true label (TL), a complementary label (CL) specifies a class that a pattern does not belong to, and hence, collecting CLs would be less laborious than collecting TLs. In this article, we propose a novel setting where the source domain is composed of *complementary-label* data, and a theoretical bound of this setting is provided. We consider two cases of this setting: one is that the source domain only contains complementary-label data [completely complementary UDA (CC-UDA)] and the other is that the source domain has plenty of complementary-label data and a small amount of true-label data [partly complementary UDA (PC-UDA)]. To this end, a *complementary label adversarial network* (CLARINET) is proposed to solve CC-UDA and PC-UDA problems. CLARINET maintains two deep networks simultaneously, with one focusing on classifying the complementary-label source data and the other taking care of the source-to-target distributional adaptation. Experiments show that CLARINET significantly outperforms a series of competent baselines on handwritten digit-recognition and object-recognition tasks.

**Index Terms**—Complementary labels (CLs), deep learning, domain adaptation (DA), transfer learning.

## I. INTRODUCTION

DOMAIN adaptation (DA) aims to train a target-domain classifier with data in source and target domains [1]–[4]. Based on the availability of data in the target domain (e.g., fully-labeled, partially-labeled and unlabeled), DA is divided into three categories: supervised DA [5]–[7], semisupervised

Manuscript received July 13, 2020; revised January 31, 2021 and May 14, 2021; accepted May 24, 2021. This work was supported by the Australian Research Council (ARC) under Grant FL190100149. The work of Yiyang Zhang was supported by the Australian Artificial Intelligence Institute, University of Technology Sydney (UTS-AAII). (Yiyang Zhang, Feng Liu, and Zhen Fang contributed equally to this work.) (Corresponding author: Jie Lu.)

Yiyang Zhang is with the Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, NSW 2007, Australia, and also with the Shenzhen International Graduate School, Tsinghua University, Shenzhen 100084, China (e-mail: zhangyiy18@mails.tsinghua.edu.cn).

Feng Liu, Zhen Fang, Guangquan Zhang, and Jie Lu are with the Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: feng.liu@uts.edu.au; zhen.fang@student.uts.edu.au; guangquan.zhang@uts.edu.au; jie.lu@uts.edu.au).

Bo Yuan is with the Shenzhen International Graduate School, Tsinghua University, Shenzhen 100084, China (e-mail: boyuan@ieee.org).

This article has supplementary material provided by the authors and color versions of one or more figures available at <https://doi.org/10.1109/TNNLS.2021.3086093>.

Digital Object Identifier 10.1109/TNNLS.2021.3086093

DA [8]–[10], and *unsupervised DA* (UDA) [11]–[13]. In practical applications, UDA is more challenging and promising than the other two as the labeled target domain data are not needed [14]–[16].

UDA methods train a target-domain classifier with massive true-label data from the source domain (true-label source data) and unlabeled data from the target domain (unlabeled target data). Existing works in the literature can be roughly categorized into the following three groups: integral-probability-metrics based UDA [17], [18]; adversarial-training based UDA [19], [20]; and causality-based UDA [21], [22]. Because adversarial-training-based UDA methods extract better domain-invariant representations using deep networks, they usually have good target-domain accuracy [23].

However, the success of UDA still highly relies on the scale of true-label source data (black dashed line in Fig. 1). Namely, the target-domain accuracy of a UDA method, e.g., conditional domain adversarial network (CDAN) [20], decays when the scale of true-label source data decreases and we prove this phenomenon in the experiment section. Hence, massive true-label source data are inevitably required by UDA methods, which is very expensive and even prohibitive.

Although determining the correct label from many candidates is laborious, choosing one of the incorrect labels [i.e., complementary labels (CLs)], e.g., labeling a cat as “Not Monkey” (as shown in Fig. 2) would be much easier and quicker, thus less costly, especially when we have many candidates [24]. Comparing with picking out the true label (TL) from many candidates, judging the correctness of a label randomly given by the system is much easier. Besides, it is impossible to find crowd-workers who have the knowledge of all classes in some areas (e.g., translation), and it is hard to collect TLs in traditional way in this case. However, we can choose a class randomly, and then assign the unlabeled sample to the expert of that randomly chosen class. It would be feasible to judge the correctness of the chosen label.

In addition to reducing costs, CLs also could help ensure data privacy. Even when we can collect TLs, it is sometimes better to convert them to CLs on purpose. This way, when companies suffer from data leakage, they won not reveal the TL of customers. In addition, such business strategy might make customers who do not want their data to be saved in databases more comfortable.

This brings us a novel setting, complementary-label based UDA, which aims to transfer knowledge from

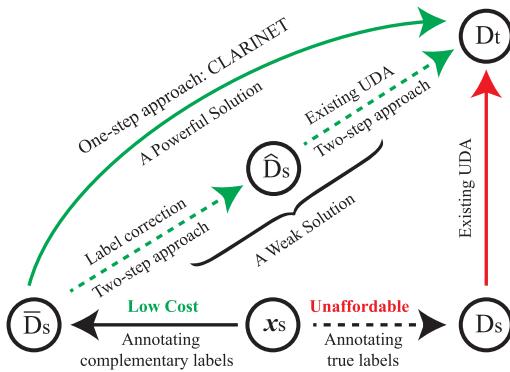


Fig. 1. Complementary label-based UDA. The red line denotes that UDA methods transfer knowledge from  $D_s$  (true-label source data) to  $D_t$  (unlabeled target data). However, acquiring true-label source data is *costly* and *unaffordable* (black dashed line,  $x_s \rightarrow D_s$ ,  $x_s$  means unlabeled source data). This brings complementary label-based UDA, namely transferring knowledge from  $\bar{D}_s$  (complementary-label source data) to  $D_t$ . It is much less costly to collect complementary-label source data (black line, required by our setting) than collecting the true-label one (black dashed line, required by UDA). To handle complementary label-based UDA, a weak solution is a two-step approach (green dashed line), which sequentially combines complementary label-learning methods ( $\bar{D}_s \rightarrow \hat{D}_s$ , label correction) and existing UDA methods ( $\hat{D}_s \rightarrow D_t$ ). This article proposes a one-step approach called CLARINET, (green line,  $\bar{D}_s \rightarrow D_t$  directly).

complementary-label source data to unlabeled target data (Fig. 1). Compared to ordinary UDA, we can greatly save the labeling cost by annotating CLs in the source domain rather than annotating TLs [24], [25]. Please note, existing UDA methods cannot handle the complementary-label based UDA, as they require source data with complete TLs [12], [13] or at least 20% true-label source data [26], [27].<sup>1</sup>

In the previous work [28], we consider using completely complementary-label data in the source domain, while actually we could also get a small amount of TLs when collecting CLs [24]. Therefore, the previous work was flawed as it did not make good use of the existing true-label data. Furthermore, experiments were conducted only on some digit datasets and a thorough learning bound was not provided. Aiming at these defects, in this work, we consider a generalized and completed version of the complementary-label based UDA problem setting.

A straightforward but weak solution to complementary-label based UDA is a two-step approach, which sequentially combines *complementary-label learning* methods and existing UDA methods (green dashed line in Fig. 1).<sup>2</sup> Complementary-label learning methods are used to assign pseudo labels for complementary-label source data. Then, we can train a target-domain classifier with pseudo-label source data and unlabeled target data using existing UDA methods. Nevertheless, pseudo-label source data contain noise, which may cause poor DA performance of this two-step approach [26].

<sup>1</sup>In [26] and [27], they consider the case where the sample in the source domain is noisy. With only 20% TLs, some of the baseline models achieve very low target domain accuracy. Therefore, Liu *et al.* [26] believe that at least 20% true-label source data are needed to realize DA.

<sup>2</sup>We implement this two-step approach and take it as a baseline.

Therefore, we propose a powerful one-step solution, *complementary label adversarial network* (CLARINET). It maintains two deep networks trained by adversarial way simultaneously, where one can accurately classify complementary-label source data, and the other can discriminate source and target domains. Since Long *et al.* [20] and Song *et al.* [29] have shown that the multimodal structures of distributions can only be captured sufficiently by the cross-covariance dependence between the features and classes (i.e., TLs), we set the input of domain discriminator  $D$  as the outer product of feature representation (e.g.,  $\mathbf{g}_s$  in Fig. 3) and scattered classifier prediction (e.g.,  $\mathbf{T}(\mathbf{f}_s)$  in Fig. 3).

Due to the nature of complementary-label classification, the predicted probability of each class (i.e., each element of  $\mathbf{f}_s$ , Fig. 3) is relatively close. According to [29], this kind of predicted probabilities could not provide sufficient information to capture the multimodal structure of distributions. To fix it, we add a sharpening function  $T$  to make the predicted probabilities more scattered (i.e.,  $\mathbf{T}(\mathbf{f}_s)$ , Fig. 3) than previous ones (i.e.,  $\mathbf{f}_s$ , Fig. 3). By doing so, the scattered classifier predictions can better indicate their choice. In this way, we can take full advantage of classifier predictions and effectively align distributions of two domains. Our ablation study (see Table III) verifies that the sharpening function  $T$  indeed helps improve the target-domain accuracy.

We conduct experiments on seven complementary label-based UDA tasks and compare CLARINET with a series of competent baselines. Empirical results demonstrated that CLARINET effectively transfers knowledge from complementary-label source data to unlabeled target data and is superior to all baselines. We also show that the target-domain accuracy of CLARINET will increase if a small amount of true-label source data are available. To make up for the defects of previous conference paper [28], the main contributions of this article are summarized as follows:

- 1) We present a generalized version of the complementary label-based UDA. This article considers two cases of complementary label-based UDA: one is that the source domain only contains complementary-label data [completely complementary UDA (CC-UDA)], and the other is that the source domain also contains a small amount of true-label data [partly complementary UDA (PC-UDA)].
- 2) We provide a thorough theoretical analysis of the expected target-domain risk of our approach, presenting a learning bound of complementary label-based UDA.
- 3) Apart from the handwritten digit datasets, we also conduct experiments on more complex image datasets, proving the applicability of complementary label-based UDA.

This article is organized as follows: Section II reviews the works related to DA, complementary-label learning, and low-cost UDA. Section III introduces the problem setting and proves a learning bound of this setting. Section IV introduces a straightforward but weak two-step approach to complementary label-based UDA. The proposed powerful one-step solution is shown in Section V. Experimental results and analyses are



Fig. 2. TL (top) versus CL (bottom).

provided in Section VI. Finally, Section VII concludes this article.

## II. RELATED WORKS

In this section, we discuss previous works that are most related to our work, and highlight our differences from them. We mainly review some related works about DA, complementary-label learning, and low-cost UDA.

### A. Domain Adaptation

DA generalizes a learner across different domains by matching the distributions of source and target domains. It has wide applications in computer vision [30]–[32] and natural language processing [33], [34], and so on. Previous DA methods in the shallow regime try to either bridge the source and target domains by learning invariant feature representations or estimate instance importance using labeled source data and unlabeled target data [35], [36]. Later, it is confirmed that deep learning methods formed by the composition of multiple nonlinear transformations yield abstract and ultimately useful representations [37]. Besides, the learned deep representations to some extent are general and are transferable to similar tasks [38]. Hence, deep neural networks have been explored for DA.

Concurrently, multiple methods of matching the feature distributions in the source and the target domains have been proposed for UDA. The first category learns domain invariant features by minimizing a distance between distributions, such as maximum mean discrepancy (MMD) [39]. In deep adaptation network (DAN), Long *et al.* [17] minimize the marginal distributions of two domains by *multikernel MMD* (MK-MMD) metric. An alternative way of learning domain-invariant features in UDA is inspired by the generative adversarial networks (GANs). By confusing a domain classifier (or discriminator), the deep networks can explore nondiscriminative representations. The adversarial training-based UDA methods always try to play a two-player minimax game. Domain-adversarial neural network (DANN) [40] employs a gradient reversal layer to realize the minimax optimization. In [20], they propose CDAN, which conditions the models on discriminative information conveyed in the classifier predictions. Some works study the UDA problem from a causal point of view where they consider the label  $Y$  is the cause for feature representation  $X$ . In, Gong *et al.* [21] aim to extract conditional transferable components whose conditional distribution is invariant after proper transformations.

However, the aforementioned methods are all based on the true-label source data, which require high labeling costs. In our work, we propose a new setting by using complementary-label source data instead of true-label source data, which significantly saves the labeling cost.

### B. Complementary-Label Learning

Complementary-label learning (shown in Fig. 2) is one type of weak supervision-learning approaches, which was first proposed by Ishida *et al.* [24]. They give theoretical analysis with a statistical consistency guarantee to show classification risk can be recovered only from complementary-label data. Nevertheless, they require the CL must be chosen in an unbiased way and allow only one-versus-all and pairwise comparison multiclass loss functions with certain nonconvex binary losses. Namely, softmax cross-entropy loss, which is the most popular loss used in deep learning could not be used to solve the problem.

Later, Yu *et al.* [25] extended the problem setting to where CL could be chosen in the biased way with the assumption that a small set of easily distinguishable true-label data are available in practice. In their point of view, because humans are biased toward their own experience, it is unrealistic to guarantee that CL is chosen in an unbiased way. For example, if an annotator is more familiar with one class than with another, he/she is more likely to employ the more familiar one as a CL. They solve the problem by employing the forward loss correction technique to adjust the learning objective, but limiting the loss function to softmax cross-entropy loss. They theoretically ensure that the classifier learned with CLs converges to the optimal one learned with TLs.

Recently, Ishida *et al.* [41] proposed a new unbiased risk estimator under the unbiased label-chosen assumption. They make any loss functions available for use and have no implicit assumptions on the classifier, namely the estimator could be used for arbitrary models and losses, including softmax cross-entropy loss. They further investigate correction schemes to make CL learning practical and demonstrate the performance. Thus, in our article, we take advantage of this estimator for the source domain classification and generalize it to the UDA field.

### C. Low-Cost UDA

The UDA with low-cost source data has recently attracted much attention. For instance, in [26], they consider the situation where the labeled data in the source domain come from

amateur annotators or the Internet [42], [43]. As in the wild, acquiring a large amount of perfectly clean labeled data in the source domain is high cost and sometimes impossible. They name the problem as wildly UDA (WUDA), which aims to transfer knowledge from noisy labeled data in the source domain to unlabeled target data. They show that WUDA ruins all UDA methods if no care is being taken of label noise in the source domain and propose a Butterfly framework, a powerful and efficient solution to WUDA.

Long *et al.* [20] consider the weakly supervised DA, where the source domain with noises in labels, features, or both could be tolerated [27]. Label noise refers to incorrect labels of images because of errors in manual annotation, and feature noise refers to low-quality pixels of images, which may come from blur, overlap, occlusion, or corruption, and so on. They present a transferable curriculum learning (TCL) approach, extending from curriculum learning and adversarial learning. The TCL model aims to be robust to both sample noises and distribution shift by employing a curriculum, which could tell whether a sample is easy and transferable.

In [28], we consider another way to save the labeling cost by using completely complementary-label data in the source domain and prove that distributional adaptation can be effectively realized from complementary-label source data to unlabeled target data. In this article, we consider two cases of using complementary-label data in the source domain and prove that we could use a small amount of true-label data to improve the transfer result. Besides, as shown in [24], we can obtain true-label data and complementary-label data simultaneously, so that getting a small amount of true-label data is guaranteed to be low cost. Furthermore, we provide an analysis of the expected target-domain risk of our approach. In Sections III–V, we will introduce the complementary label-based UDA and explain how to address such tasks.

### III. COMPLEMENTARY LABEL-BASED UDA

This section presents a novel problem setting, complementary label-based UDA, and prove a learning bound for it. Then, we show its benefits for DA field.

#### A. Problem Setting

In complementary label-based UDA, we aim to realize distributional adaptation from complementary-label source data to unlabeled target data. We first consider the situation where there are only complementary-label data in the source domain, namely *completely complementary UDA* (CC-UDA). Let  $\mathcal{X} \subset \mathbb{R}^d$  be a feature (input) space and  $\mathcal{Y} := \{\mathbf{y}_1, \dots, \mathbf{y}_c, \dots, \mathbf{y}_K\}$  be a label (output) space, where  $\mathbf{y}_c$  is the one-hot vector for label  $c$ . A *domain* is defined as follows:

*Definition 1 (Domains for CC-UDA):* Given random variables  $X_s, X_t \in \mathcal{X}$ ,  $Y_s, Y_t \in \mathcal{Y}$ , the source and target domains are joint distributions  $P(X_s, \bar{Y}_s)$  and  $P(X_t, Y_t)$ , where the joint distributions  $P(X_s, Y_s) \neq P(X_t, Y_t)$  and  $P(\bar{Y}_s = \mathbf{y}_c | Y_s = \mathbf{y}_c) = 0$  for all  $\mathbf{y}_c \in \mathcal{Y}$ .

Then, we propose CC-UDA problem as follows:

*Problem 1 (CC-UDA):* Given independent and identically distributed (i.i.d.) labeled samples  $\bar{D}_s = \{(\mathbf{x}_s^i, \bar{\mathbf{y}}_s^i)\}_{i=1}^{\bar{n}_s}$  drawn from the source domain  $P(X_s, \bar{Y}_s)$  and i.i.d. unlabeled samples  $D_t = \{\mathbf{x}_t^i\}_{i=1}^{n_t}$  drawn from the target marginal distribution  $P(X_t)$ , the aim of CC-UDA is to train a classifier  $F_t : \mathcal{X} \rightarrow \mathcal{Y}$  with  $\bar{D}_s$  and  $D_t$  such that  $F_t$  can accurately classify target data drawn from  $P(X_t)$ .

It is clear that it is impossible to design a suitable learning procedure without any assumptions on  $P(X_s, \bar{Y}_s)$ . In this article, we use the assumption for unbiased complementary-label learning proposed by Ishida *et al.* [24] and Bousmalis *et al.* [41]:

$$P(\bar{Y}_s = \mathbf{y}_k | X_s) = \frac{1}{K-1} \sum_{c=1, c \neq k}^K P(Y_s = \mathbf{y}_c | X_s) \quad (1)$$

for all  $k, c \in \{1, \dots, K\}$  and  $c \neq k$ . This unbiased assumption indicates that the selection of CLs for samples is with equal probability.

Ishida *et al.* [24] propose an efficient way to collect labels through crowdsourcing: we choose one of the classes randomly and ask crowd-workers whether a pattern belongs to the chosen class or not. Then the chosen class is treated as TL if the answer is yes; otherwise, the chosen class is regarded as CL. Such a yes/no question is much easier and quicker than selecting the correct class from the list of all candidate classes, which sometimes could even be impossible. In addition, we could guarantee that the data obtained through this way are under unbiased assumption in (1).

As we can obtain true-label data and complementary-label data simultaneously, we also consider the problem that the source domain contains a few true-label data. We name this problem as *partly complementary UDA* (PC-UDA).

*Problem 2 (PC-UDA):* Given i.i.d labeled samples  $\bar{D}_s = \{(\mathbf{x}_s^i, \bar{\mathbf{y}}_s^i)\}_{i=1}^{\bar{n}_s}$  drawn from the domain  $P(X_s, \bar{Y}_s)$ ,  $D_s = \{(\mathbf{x}_s^i, \mathbf{y}_s^i)\}_{i=\bar{n}_s+1}^{\bar{n}_s+n_s}$  drawn from the domain  $P(X_s, Y_s)$ , and i.i.d. unlabeled samples  $D_t = \{\mathbf{x}_t^i\}_{i=1}^{n_t}$  drawn from the target marginal distribution  $P(X_t)$ , the aim is to find a target classifier  $F_t : \mathcal{X} \rightarrow \mathcal{Y}$  such that  $F_t$  classifies target samples into the correct classes.

It is actually a more common situation to have a small amount of true-label data. If we leverage both kinds of labeled data properly, we could obtain a more accurate classifier. Ishida *et al.* [24] have demonstrated the usefulness of combining true-label and complementary-label data in classification problem. We will further show that in UDA field, we could also use both true-label and complementary-label source data to realize knowledge transfer and utilize the true-label data to improve the result.

#### B. Learning Bound of Complementary Label-Based UDA

A learning bound of complementary label-based UDA is presented in this section. We could prove that we can limit the risk in the target domain. A practitioner may safely skip it.

If given a feature transformation

$$\begin{aligned} G : \mathcal{X} &\rightarrow \mathcal{X}_G := G(\mathcal{X}) \\ \mathbf{x} &\rightarrow \mathbf{x}_G := G(\mathbf{x}) \end{aligned} \quad (2)$$

then the induced distributions related to  $P_{X_s}$  and  $P_{X_t}$  are

$$\begin{aligned} G_{\#}P_{X_s} &:= P(G(X_s)) \\ G_{\#}P_{X_t} &:= P(G(X_t)). \end{aligned} \quad (3)$$

Following the notations in [44], consider a multiclass classification task with a hypothesis space  $\mathcal{H}_G$  of the classifiers:

$$\begin{aligned} F : \mathcal{X}_G &\rightarrow \mathcal{Y} \\ \mathbf{x} &\rightarrow [C_1(\mathbf{x}), \dots, C_K(\mathbf{x})]^T. \end{aligned} \quad (4)$$

Let

$$\begin{aligned} \ell : \mathbb{R}^K \times \mathbb{R}^K &\rightarrow \mathbb{R}_{\geq 0} \\ (\mathbf{y}, \tilde{\mathbf{y}}) &\rightarrow \ell(\mathbf{y}, \tilde{\mathbf{y}}) \end{aligned} \quad (5)$$

be the loss function. For convenience, we also require  $\ell$  satisfying the following conditions in theoretical part:

- 1)  $\ell$  is symmetric and satisfies triangle inequality;
- 2)  $\ell(\mathbf{y}, \tilde{\mathbf{y}}) = 0$  iff  $\mathbf{y} = \tilde{\mathbf{y}}$ ; and
- 3)  $\ell(\mathbf{y}, \tilde{\mathbf{y}}) \equiv 1$  if  $\mathbf{y} \neq \tilde{\mathbf{y}}$  and  $\mathbf{y}, \tilde{\mathbf{y}}$  are one-hot vectors.

We can check many losses satisfying the above conditions, such as 0–1 loss  $1_{\mathbf{y} \neq \tilde{\mathbf{y}}}$  and  $\ell_2$  loss  $(1/2)\|\mathbf{y} - \tilde{\mathbf{y}}\|_2^2$ . The complementary risk for  $F \circ G$  with respect to  $\ell$  over  $P(X_s, \bar{Y}_s)$  is

$$L_{\bar{s}}(F \circ G) = \mathbb{E}\ell(F \circ G(X_s), \bar{Y}_s).$$

The risks for the decision function  $F \circ G$  with respect to loss  $\ell$  over implicit distribution  $P(X_s, Y_s), P(X_t, Y_t)$  are

$$\begin{aligned} L_s(F \circ G) &= \mathbb{E}\ell(F \circ G(X_s), Y_s) \\ L_t(F \circ G) &= \mathbb{E}\ell(F \circ G(X_t), Y_t). \end{aligned}$$

In this article, we propose a tighter distance named tensor discrepancy distance. The tensor discrepancy distance can future match the pseudo-conditional distributions.

We consider the following tensor mapping:

$$\begin{aligned} \otimes_F : \mathcal{X}_G &\rightarrow \mathcal{X}_G \otimes \mathcal{Y}_t \\ \mathbf{x}_G &\rightarrow \mathbf{x}_G \otimes F(\mathbf{x}_G). \end{aligned} \quad (6)$$

Then we induce two important distributions:

$$\begin{aligned} \otimes_{F\#}P_{X_s} &:= P(\otimes_F(G(X_s))) \\ \otimes_{F\#}P_{X_t} &:= P(\otimes_F(G(X_t))). \end{aligned} \quad (7)$$

Using  $\mathcal{H}_G$ , we reconstruct a new hypothetical set

$$\Delta_{F,G} := \{\delta_{\bar{F}} : \mathcal{X}_G \otimes \mathcal{Y}_t \rightarrow \mathbb{R} : \bar{F} \in \mathcal{H}_G\} \quad (8)$$

where  $\delta_{\bar{F}}(\mathbf{x}_G \otimes \mathbf{y}) = |\otimes_F(\mathbf{x}_G) - \otimes_{\bar{F}}(\mathbf{x}_G)|$ . Then the distance between  $\otimes_{F\#}P_{X_s}$  and  $\otimes_{F\#}P_{X_t}$  is

$$\begin{aligned} d_{\Delta_{F,G}}^\ell(\otimes_{F\#}P_{X_s}, \otimes_{F\#}P_{X_t}) &= \sup_{\delta \in \Delta_{F,G}} \left| \mathbb{E}_{\mathbf{z} \sim \otimes_{F\#}P_{X_s}} \text{sgn} \circ \delta(\mathbf{z}) - \mathbb{E}_{\mathbf{z} \sim \otimes_{F\#}P_{X_t}} \text{sgn} \circ \delta(\mathbf{z}) \right| \end{aligned} \quad (9)$$

where  $\text{sgn}$  is the sign function.

It is easy to prove that under the conditions 1–3 for loss  $\ell$  and for any  $F \in \mathcal{H}_G$ , we have

$$d_{\Delta_{F,G}}^\ell(\otimes_{F\#}P_{X_s}, \otimes_{F\#}P_{X_t}) \leq d_{\mathcal{H}_G}^\ell(G_{\#}P_{X_s}, G_{\#}P_{X_t}) \quad (10)$$

where  $d_{\mathcal{H}_G}^\ell$  is the distribution discrepancy defined in [45] and [46]. Then, we introduce our main theorem as follows:

**Theorem 1:** Given a loss function  $\ell$  satisfying conditions 1–3 and a hypothesis space  $\mathcal{H}_G \subset \{F : \mathcal{X}_G \rightarrow \mathcal{Y}\}$ , under unbiased assumption, for any  $F \in \mathcal{H}_G$ , we have

$$L_t(F \circ G) \leq \bar{L}_s(F \circ G) + \Lambda + d_{\Delta_{F,G}}^\ell(\otimes_{F\#}P_{X_s}, \otimes_{F\#}P_{X_t})$$

where  $\bar{L}_s(F \circ G) := \sum_{k=1}^K \int_{\mathcal{X}} \ell(F \circ G(\mathbf{x}), k) dP_{X_s} - (K-1)L_{\bar{s}}(F \circ G)$ ,  $P_{X_s}, P_{X_t}$  are source and target marginal distributions,  $\Lambda = \min_{F \in \mathcal{H}_G} R_s(F \circ G) + R_t(F \circ G)$ .

**Proof:** First, we prove that  $L_s(F \circ G) = \bar{L}_s(F \circ G)$ . To prove it, we investigate the connection between  $L_s(F \circ G)$  and  $L_{\bar{s}}(F \circ G)$  under unbiased assumption in (1). Given  $K \times K$  matrix  $Q$  whose diagonal elements are 0 and other elements are  $1/K$ , we represent the unbiased assumption by

$$\bar{\eta} = Q\eta \quad (11)$$

where  $\bar{\eta} = [P(\bar{Y}_s = \mathbf{y}_1 | X_s), \dots, P(\bar{Y}_s = \mathbf{y}_K | X_s)]^T$  and  $\eta = [P(Y_s = \mathbf{y}_1 | X_s), \dots, P(Y_s = \mathbf{y}_K | X_s)]^T$ . Note that  $Q$  has inverse matrix  $Q^{-1}$  whose diagonal elements are  $-(K-2)$  and other elements are 1. Thus, we have

$$Q^{-1}\bar{\eta} = \eta. \quad (12)$$

According to (12), we have  $P(Y_s = \mathbf{y}_k | X_s) = 1 - (K-1)P(\bar{Y}_s = \mathbf{y}_k | X_s)$ , which implies that

$$\begin{aligned} L_s(F \circ G) &= \sum_{k=1}^K \int_{\mathcal{X}} \ell(F \circ G(\mathbf{x}), k) dP_{X_s} \\ &\quad - (K-1)L_{\bar{s}}(F \circ G). \end{aligned} \quad (13)$$

Hence,  $L_s(F \circ G) = \bar{L}_s(F \circ G)$ . The empirical form of (13) is known as complementary-label loss [see (20)].

Next, we will prove that

$$L_t(F \circ G) - L_s(F \circ G) \leq \Lambda + d_{\Delta_{F,G}}^\ell(\otimes_{F\#}P_{X_s}, \otimes_{F\#}P_{X_t}).$$

As if it is true, combined with  $L_s(F \circ G) = \bar{L}_s(F \circ G)$ , we could easily prove the theorem. It is clearly that

$$\begin{aligned} L_t(F \circ G) - L_s(F \circ G) &= \int_{\mathcal{X} \times \mathcal{Y}_t} \ell(F \circ G(\mathbf{x}), \mathbf{y}) dP_{X_t Y_t} - \int_{\mathcal{X} \times \mathcal{Y}_s} \ell(F \circ G(\mathbf{x}), \mathbf{y}) dP_{X_s Y_s} \\ &\leq L_t(\tilde{F} \circ G) + \int_{\mathcal{X} \times \mathcal{Y}^t} \ell(F \circ G(\mathbf{x}), \tilde{F} \circ G(\mathbf{x})) dP_{X_t Y_t} \\ &\quad + L_s(\tilde{F} \circ G) - \int_{\mathcal{X} \times \mathcal{Y}_s} \ell(F \circ G(\mathbf{x}), \tilde{F} \circ G(\mathbf{x})) dP_{X_s Y_s} \end{aligned} \quad (14)$$

where  $\tilde{F}$  is any function from  $\mathcal{H}_G$ . According to conditions 1–3, we have

$$\begin{aligned} \mathbb{E}_{\mathbf{z} \sim \otimes_{F\#}P_{X_s}} \text{sgn} \circ \delta_{\tilde{F}}(\mathbf{z}) &= \int \text{sgn} \circ \delta_{\tilde{F}}(\mathbf{z}) d\otimes_{F\#}P_{X_s} \\ &= \int_{\mathcal{X}} |F \circ G(\mathbf{x}) - \tilde{F} \circ G(\mathbf{x})| dP_{X_s} \\ &= \int_{\mathcal{X}} \ell(F \circ G(\mathbf{x}), \tilde{F} \circ G(\mathbf{x})) dP_{X_s} \end{aligned} \quad (15)$$

similarly

$$\mathbb{E}_{\mathbf{z} \sim \otimes_{F\#}P_{X_t}} \text{sgn} \circ \delta_{\tilde{F}}(\mathbf{z}) = \int_{\mathcal{X}} \ell(F \circ G(\mathbf{x}), \tilde{F} \circ G(\mathbf{x})) dP_{X_t} \quad (16)$$

hence, according to the definition of (9), we have

$$\begin{aligned} d_{\Delta_{F,G}}^\ell(\otimes_{F\#} P_{X_s}, \otimes_{F\#} P_{X_t}) \\ = \sup_{\tilde{F}, G \in \Delta_{F,G}} \left| \int_{\mathcal{X}} \ell(F \circ G(\mathbf{x}), \tilde{F} \circ G(\mathbf{x})) dP_{X_s} \right. \\ \left. - \int_{\mathcal{X}} \ell(F \circ G(\mathbf{x}), \tilde{F} \circ G(\mathbf{x})) dP_{X_t} \right|. \end{aligned} \quad (17)$$

Combining (14) and (17), we have

$$\begin{aligned} L_t(F \circ G) - L_s(F \circ G) \\ \leq \min(L_t(\tilde{F} \circ G) + L_s(\tilde{F} \circ G)) + d_{\Delta_{F,G}}^\ell(\otimes_{F\#} P_{X_s}, \otimes_{F\#} P_{X_t}) \\ = \Lambda + d_{\Delta_{F,G}}^\ell(\otimes_{F\#} P_{X_s}, \otimes_{F\#} P_{X_t}). \end{aligned} \quad (18)$$

Hence, we prove this theorem.  $\square$

### C. Benefits for DA Field

Collecting true-label data is always expensive in the real world. Thus, learning from less expensive data [47]–[50] has been extensively studied in machine-learning field, including label-noise learning [51]–[53], pairwise/triple-wise constraints learning [54]–[56], positive-unlabeled learning [57]–[59], complementary-label learning [24], [25], [41], and so on. Among all these research directions, obtaining CLs is a cost-effective option. As described in the previous works mentioned above, compared with choosing the true class out of many candidate classes precisely, collecting CLs is obviously much easier and less costly. In addition, a classifier trained with complementary-label data is equivalent to a classifier trained with true-label data as shown in [41].

Actually in the field of DA, the high cost of true-label data is also an important issue. At present, the success of DA still highly relies on the scale of true-label source data, which is a critical bottleneck. Under low-cost limitation, it is unrealistic to obtain enough true-label source data and thus cannot achieve a good distribution adaptation result. For the same cost, we can get multiple times more complementary-label data than the true-label data. In addition, the adaptation scenario is limited to some commonly used datasets, e.g., handwritten digit datasets, as they have sufficient TLs to support distributional adaptation. This fact makes it difficult to generalize DA to more real-world scenarios where it is needed. Thus, if we can reduce the labeling cost in the source domain, for example, by using complementary-label data to replace true-label data (complementary label-based UDA), we can promote DA to more fields.

Because existing UDA methods require at least 20% true-label source data [26], they cannot handle complementary label-based UDA problem. To address the problem, we introduce a two-step approach, straightforward but weak solution, and then propose a powerful one-step solution, CLARINET.

### IV. TWO-STEP APPROACH

To solve the problem that existing UDA methods cannot be applied to complementary label-based UDA problems directly, a straightforward way is to apply a two-step strategy. Namely, we could sequentially combine complementary label-learning

---

### Algorithm 1 Two-Step Approach for CC-UDA Tasks

---

**Input:**  $\bar{D}_s = \{(\mathbf{x}_s^i, \bar{\mathbf{y}}_s^i)\}_{i=1}^{\bar{n}_s}$ ,  $D_t = \{\mathbf{x}_t^i\}_{i=1}^{n_t}$ .

**Output:** the target-domain classifier.

- 1: **Train** a classifier  $C$  using  $\bar{D}_s = \{(\mathbf{x}_s^i, \bar{\mathbf{y}}_s^i)\}_{i=1}^{\bar{n}_s}$  based on the complementary label-learning algorithm.
  - 2: **Use**  $C$  to pseudo-label  $\bar{D}_s = \{\mathbf{x}_s^i\}_{i=1}^{\bar{n}_s}$ , namely generate pseudo-label source data  $\hat{D}_s = \{(\mathbf{x}_s^i, \hat{\mathbf{y}}_s^i)\}_{i=1}^{\hat{n}_s}$ .
  - 3: **Apply** normal UDA methods on  $\hat{D}_s = \{(\mathbf{x}_s^i, \hat{\mathbf{y}}_s^i)\}_{i=1}^{\hat{n}_s}$  and  $D_t = \{\mathbf{x}_t^i\}_{i=1}^{n_t}$  to train a target-domain classifier.
- 

methods and existing UDA methods. Algorithm 1 presents how we realize the two-step approach for CC-UDA tasks specifically. In the two-step approach, we first use the complementary label-learning algorithm to train a classifier on the complementary-label source data (line 1). Then, we take advantage of the classifier to assign pseudo labels for source domain data (line 2). Finally, we train the target-domain classifier with pseudo-label source data and unlabeled target data using existing UDA methods (line 3). In this way, we can transfer knowledge from the newly formed pseudo-label source data to unlabeled target data. As for PC-UDA tasks, we could combine the pseudo-label source data obtained by following the first two steps and existing true-label source data together to train the target-domain classifier.

Nevertheless, the pseudo-label source data contain noise, as complementary label-learning algorithms cannot be trained to produce a completely accurate classifier. As the noise will bring poor domain-adaptation performance [26], the two-step approach is a suboptimal choice. To solve this problem, we consider implementing both complementary-label learning and UDA in a network. In this way, the network will always try to classify source domain data accurately during the adaptation procedure. Besides, we consider using entropy conditioning to make the transfer process mainly based on the classification results with high confidence, which can greatly eliminate the noise effect compared with the two-step approach. Therefore, we propose a powerful one-step solution to complementary label-based UDA, CLARINET.

### V. CLARINET: POWERFUL ONE-STEP APPROACH

The proposed CLARINET (as shown in Fig. 3) realizes distributional adaptation in an adversarial way, which mainly consists of feature extractor  $G$ , label predictor  $F$ , and domain discriminator  $D$ . By working adversarially to domain discriminator  $D$ , feature extractor  $G$  encourages domain-invariant features to emerge. Label predictor  $F$  are trained to discriminate different classes based on such features.

In this section, we first introduce two losses used to train CLARINET, complementary-label loss and scattered conditional adversarial loss. Then the whole training procedure of CLARINET is presented. Finally, we show how to adjust CLARINET for PC-UDA tasks if a small amount of true-label source data are available.

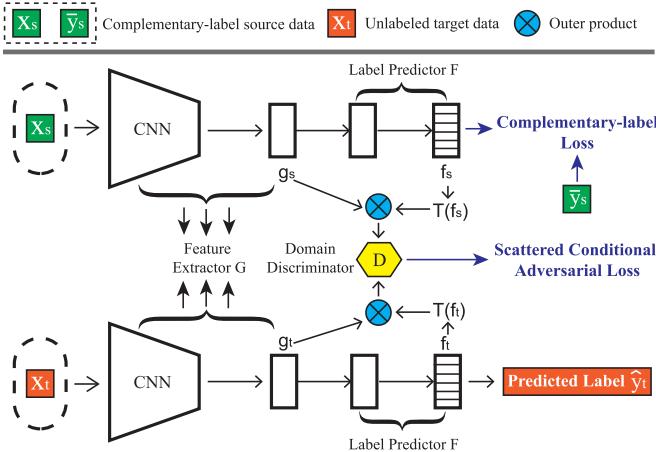


Fig. 3. Overview of the proposed CLARINET. It consists of feature extractor  $G$ , label predictor  $F$ , and conditional domain discriminator  $D$ .  $g_s$  and  $g_t$  are outputs of  $G$ , representing extracted features of source and target data.  $f_s$  and  $f_t$  represent classifier predictions.  $T$  is a sharpening function, which we propose to scatter the classifier predictions. In Algorithm 2, we show how to use two losses mentioned in this figure to train CLARINET.

#### A. Loss Function in CLARINET

In this section, we introduce how to compute the two losses mentioned above in CLARINET after obtaining mini-batch  $\bar{d}_s$  from  $\overline{D}_s$  and  $d_t$  from  $D_t$ .

1) *Complementary-Label Loss*: It is designed to reduce the source classification error based on complementary-label data (the first part in the bound). We first divided  $\overline{d}_s$  into  $K$  disjoint subsets according to the CLs in  $\overline{d}_s$

$$\overline{d}_s = \cup_{k=1}^K \overline{d}_{s,k}, \quad \overline{d}_{s,k} = \{(\mathbf{x}_k^i, \mathbf{y}_k)\}_{i=1}^{\overline{n}_{s,k}} \quad (19)$$

where  $\overline{d}_{s,k} \cap \overline{d}_{s,k'} = \emptyset$  if  $k \neq k'$  and  $\overline{n}_{s,k} = |\overline{d}_{s,k}|$ . Then, following (13), the complementary-label loss on  $\overline{d}_{s,k}$  is

$$\begin{aligned} \overline{L}_s(G, F, \overline{d}_{s,k}) &= -(K-1) \frac{\overline{\pi}_k}{\overline{n}_{s,k}} \sum_{i=1}^{\overline{n}_{s,k}} \ell(F \circ G(\mathbf{x}_k^i), \mathbf{y}_k) \\ &\quad + \sum_{j=1}^K \frac{\overline{\pi}_j}{\overline{n}_{s,j}} \sum_{l=1}^{\overline{n}_{s,j}} \ell(F \circ G(\mathbf{x}_j^l), \mathbf{y}_k) \end{aligned} \quad (20)$$

where  $\ell$  can be any loss and we use the cross-entropy loss and  $\overline{\pi}_k$  is the proportion of the samples complementary-labeled  $k$ . The total complementary-label loss on  $\overline{d}_s$  is as follows:

$$\overline{L}_s(G, F, \overline{d}_s) = \sum_{k=1}^K \overline{L}_s(G, F, \overline{d}_{s,k}). \quad (21)$$

As shown in Section III-B, the complementary-label loss [i.e., (21)] is an unbiased estimator of the true-label-data risk. Namely, the minimizer of complementary-label loss agrees with the minimizer of the true-label-data risk with no constraints on the loss  $\ell$  and model  $F \circ G$  [41].

*Remark 1:* Because of the negative part in  $\overline{L}_s(G, F, \overline{d}_s)$ , minimizing it directly will cause over-fitting [60]. To overcome this problem, we use a correctional way [41] to minimize  $\overline{L}_s(G, F, \overline{d}_s)$  (lines 7–13 in Algorithm 2).

2) *Scattered Conditional Adversarial Loss*: It is designed to reduce distribution discrepancy distance between two domains (the third part in the bound). Adversarial DA methods [16], [19] is inspired by GANs [61]. Normally, a domain discriminator is learned to distinguish the source domain and the target domain, whereas the label predictor learns transferable representations that are indistinguishable by the domain discriminator. Namely, the final classification decisions are made based on features that are both discriminative and invariant to the change of domains [19]. It is an efficient way to reduce distribution discrepancy distance between the marginal distributions.

However, when data distributions have complex multimodal structures, which is a real scenario because of the nature of multiclass classification, adapting only the feature representation is a challenge for adversarial networks. Namely, even if the domain discriminator is confused, we could not confirm the two distributions are sufficiently similar [62].

According to [29], it is significant to capture multimodal structures of distributions using cross-covariance dependence between the features and classes (i.e., TLs). Because there are no true-label target data in UDA, CDAN adopts outer product of feature representations and classifier predictions (i.e., outputs of the softmax layer) as new features of two domains [20], which is inspired by conditional GANs (CGANs) [63], [64]. The newly constructed features have shown a great ability to discriminate source and target domains, because classifier predictions of true-label source data are dispersed, expressing the predicted goal clearly.

However, in the complementary-label classification mode, we observe that the predicted probability of each class (i.e., each element of  $f_s$  in Fig. 3) is relatively close. Namely, it is hard to find significant predictive preference from the classifier predictions. According to [29], this kind of predictions cannot provide sufficient information to capture the multimodal structure of distributions. To fix it, we add a sharpening function  $T$  to scatter the predicted probability (the output of  $f = [f_1, \dots, f_K]^T$  after softmax function,  $f$ , could be  $f_s$  or  $f_t$  in Fig. 3).

In [65], a common approach of adjusting the “temperature” of this categorical distribution is defined as follows:

$$T(f) = \left[ \frac{f_1^{\frac{1}{T}}}{\sum_{j=1}^K f_j^{\frac{1}{T}}}, \dots, \frac{f_k^{\frac{1}{T}}}{\sum_{j=1}^K f_j^{\frac{1}{T}}}, \dots, \frac{f_K^{\frac{1}{T}}}{\sum_{j=1}^K f_j^{\frac{1}{T}}} \right]^T. \quad (22)$$

As  $T \rightarrow 0$ , the output of  $T(f)$  will approach a Dirac (one-hot) distribution [66].

Then to prioritize the discriminator on those easy-to-transfer examples, following [20], we measure the uncertainty of the prediction for sample  $\mathbf{x}$  by

$$H(G, F, \mathbf{x}) = - \sum_{k=1}^K T(f_k(\mathbf{x})) \log T(f_k(\mathbf{x})). \quad (23)$$

The small result of  $H(G, F, \mathbf{x})$  implies that  $T(f_k(\mathbf{x}))$  is close to 0 or 1, which could be regarded as the prediction is with high confidence due to the existing of the final softmax layer [67].

**Algorithm 2** CLARINET for CC-UDA Tasks

---

**Input:**  $\bar{D}_s = \{(\mathbf{x}_s^i, \bar{\mathbf{y}}_s^i)\}_{i=1}^{n_s}$ ,  $D_t = \{\mathbf{x}_t^i\}_{i=1}^{n_t}$ .  
**Parameters:** learning rate  $\gamma_1$  and  $\gamma_2$ , epoch  $T_{\max}$ , start epoch  $T_s$ , iteration  $N_{\max}$ , class number  $K$ , tradeoff  $\lambda$ , network parameter  $\theta_{F \circ G}$  and  $\theta_D$ .  
**Output:** the neural network  $F \circ G$ , namely the target domain classifier for  $D_t$ .

```

1: Initialize  $\theta_{F \circ G}$  and  $\theta_D$ ;
2: for  $t = 1, 2, \dots, T_{\max}$  do
3:   Shuffle the training set  $\bar{D}_s$ ,  $D_t$ ;
4:   for  $N = 1, 2, \dots, N_{\max}$  do
5:     Fetch mini-batch  $\bar{d}_s$ ,  $d_t$  from  $\bar{D}_s$ ,  $D_t$ ;
6:     Divide  $\bar{d}_s$  into  $\{\bar{d}_{s,k}\}_{k=1}^K$ ;
7:     Calculate  $\{\bar{L}_s(G, F, \bar{d}_{s,k})\}_{k=1}^K$  using Eq. (20), and
        $\bar{L}_s(G, F, \bar{d}_s)$  using Eq. (21);
8:     if  $\min_k \{\bar{L}_s(G, F, \bar{d}_{s,k})\}_{k=1}^K \geq 0$  then
9:       Update  $\theta_{F \circ G} = \theta_{F \circ G} - \gamma_1 \nabla \bar{L}_s(G, F, \bar{d}_s)$ ;
10:      else
11:        Calculate  $\bar{L}_{\text{neg}} = \sum_{k=1}^K \min\{0, \bar{L}_s(G, F, \bar{d}_{s,k})\}$ ;
12:        Update  $\theta_{F \circ G} = \theta_{F \circ G} + \gamma_1 \nabla \bar{L}_{\text{neg}}$ ;
13:      end if
14:      if  $t > T_s$  then
15:        Calculate  $L_{\text{adv}}(G, F, D, \bar{d}_s, d_t)$  using Eq. (24);
16:        Update  $\theta_D = \theta_D - \gamma_2 \nabla L_{\text{adv}}(G, F, D, \bar{d}_s, d_t)$ ;
17:        Update  $\theta_{F \circ G} = \theta_{F \circ G} + \gamma_2 \lambda \nabla L_{\text{adv}}(G, F, D, \bar{d}_s, d_t)$ ;
18:      end if
19:    end for
20:  end for

```

---

Thus, the scattered conditional adversarial loss is as follows:

$$L_{\text{adv}}(G, F, D, \bar{d}_s, d_t) = -\frac{\sum_{\mathbf{x} \in \bar{d}_s[X]} \omega_{\bar{s}}(\mathbf{x}) \log(D(\mathbf{g}(\mathbf{x})))}{\sum_{\mathbf{x} \in \bar{d}_s[X]} \omega_{\bar{s}}(\mathbf{x})} - \frac{\sum_{\mathbf{x} \in d_t} \omega_t(\mathbf{x}) \log(1 - D(\mathbf{g}(\mathbf{x})))}{\sum_{\mathbf{x} \in d_t} \omega_t(\mathbf{x})} \quad (24)$$

where  $\omega_{\bar{s}}(\mathbf{x})$  and  $\omega_t(\mathbf{x})$  are  $1 + e^{-H(G, F, \mathbf{x})}$ ,  $\mathbf{g}(\mathbf{x})$  is  $G(\mathbf{x}) \otimes T(F \circ G(\mathbf{x}))$ , and  $\bar{d}_s[X]$  is the feature part of  $\bar{d}_s$ .

**B. Training Procedures of CLARINET**

Based on two losses proposed in Section V-A, in CLARINET, we try to solve the following optimization problem:

$$\begin{aligned} & \min_{G, F} \bar{L}_s(G, F, \bar{D}_s) - \lambda L_{\text{adv}}(G, F, D, \bar{D}_s, D_t) \\ & \min_D L_{\text{adv}}(G, F, D, \bar{D}_s, D_t) \end{aligned} \quad (25)$$

where  $D$  tries to distinguish the samples from different domains by minimizing  $L_{\text{adv}}$ , while  $F \circ G$  wants to maximize the  $L_{\text{adv}}$  to make domains indistinguishable. To solve the minimax optimization problem in (25), we add a gradient reversal layer [19] between the domain discriminator and the classifier, which multiplies the gradient by a negative constant  $(-\lambda)$  during the back propagation.  $\lambda$  is a hyperparameter between the two losses to tradeoff source risk and domain discrepancy.

The training procedures of CLARINET are shown in Algorithm 2. First, we initialize the whole network (line 1) and shuffle the training set (line 3). During each epoch, after mini-batch  $\bar{d}_s$  and  $d_t$  are fetched (line 5), we divide the source mini-batch  $\bar{d}_s$  into  $\{\bar{d}_{s,k}\}_{k=1}^K$  using (19) (line 6). Then,  $\{\bar{d}_{s,k}\}_{k=1}^K$  are used to calculate the complementary-label loss for each class (i.e.,  $\{\bar{L}_s(G, F, \bar{d}_{s,k})\}_{k=1}^K$ ) and the whole complementary-label loss  $\bar{L}_s(G, F, \bar{d}_s)$  (line 7).

If  $\min_k \{\bar{L}_s(G, F, \bar{d}_{s,k})\}_{k=1}^K \geq 0$ , we calculate the gradient  $\nabla \bar{L}_s(G, F, \bar{d}_s)$  and update parameters of  $G$  and  $F$  using gradient descent (lines 8–9). Otherwise, we sum the negative elements in  $\{\bar{L}_s(G, F, \bar{d}_{s,k})\}_{k=1}^K$  as  $\bar{L}_{\text{neg}}$  (line 11) and calculate the gradient with  $\nabla \bar{L}_{\text{neg}}$  (line 12). Then, we update parameters of  $G$  and  $F$  using gradient ascent (line 12), which is suggested by Ishida *et al.* [41]. When the number of epochs (i.e.,  $t$ ) is over  $T_s$ , we start to update parameters of  $D$  (line 14). We calculate the scattered conditional adversarial loss  $L_{\text{adv}}$  (line 15). Then,  $L_{\text{adv}}$  is minimized over  $D$  (line 16), but maximized over  $F \circ G$  (line 17) for adversarial training.

In this paragraph, we analyze the time complexity of training CLARINET. Let  $C1$  denote the cost of computing (21) and  $C2$  denote the cost of computing (24). Each epoch of training in Algorithm 2 costs  $\mathcal{O}(mC1 + mC2)$ , where  $m$  is the number of batches in each epoch.

**C. CLARINET for PC-UDA Tasks**

For PC-UDA tasks, we have both complementary-label data and true-label data in the source domain. In such cases, we want to leverage both kinds of labeled source data to help realize better adaptation results. The two loss functions mentioned in Section V-A are adjusted as follows:

After obtaining mini-batch  $d_s$  from  $D_s$ , we could calculate the classification loss based on true-label data by

$$L_s(G, F, d_s) = \ell(F \circ G(\mathbf{x}_i), \mathbf{y}_i) \quad (26)$$

where  $\ell$  is cross-entropy loss,  $d_s = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n'_s}$ , and  $n'_s = |d_s|$ . We could use a convex combination of classification risks derived from true-label data and complementary-label data to replace the oral complementary label-based only classification risk shown as follows:

$$L_c = \alpha L_s(G, F, d_s) + (1 - \alpha) \bar{L}_s(G, F, \bar{d}_s) \quad (27)$$

where  $\alpha$  depends on the cost of labeling the two kinds of data.

The new scattered conditional adversarial loss for PC-UDA tasks is as follows:

$$\begin{aligned} & L_{\text{adv}}(G, F, D, d_s, \bar{d}_s, d_t) \\ & = -\frac{\sum_{\mathbf{x} \in d_s[X]} \omega_s(\mathbf{x}) \log(D(\mathbf{g}(\mathbf{x})))}{\sum_{\mathbf{x} \in d_s[X]} \omega_s(\mathbf{x})} \\ & - \frac{\sum_{\mathbf{x} \in \bar{d}_s[X]} \omega_{\bar{s}}(\mathbf{x}) \log(D(\mathbf{g}(\mathbf{x})))}{\sum_{\mathbf{x} \in \bar{d}_s[X]} \omega_{\bar{s}}(\mathbf{x})} \\ & - \frac{\sum_{\mathbf{x} \in d_t} \omega_t(\mathbf{x}) \log(1 - D(\mathbf{g}(\mathbf{x})))}{\sum_{\mathbf{x} \in d_t} \omega_t(\mathbf{x})} \end{aligned} \quad (28)$$

where  $\omega_s(\mathbf{x})$ ,  $\omega_{\bar{s}}(\mathbf{x})$ , and  $\omega_t(\mathbf{x})$  are  $1 + e^{-H(G, F, \mathbf{x})}$ ;  $\mathbf{g}(\mathbf{x})$  is  $G(\mathbf{x}) \otimes T(F \circ G(\mathbf{x}))$ ; and  $d_s[X]$  and  $\bar{d}_s[X]$  are the feature part

**Algorithm 3** CLARINET for PC-UDA Tasks

---

**Input:**  $D_s = \{(\mathbf{x}_s^i, \mathbf{y}_s^i)\}_{i=1}^{n_s}$ ,  $\overline{D}_s = \{(\mathbf{x}_s^i, \overline{\mathbf{y}}_s^i)\}_{i=1}^{\overline{n}_s}$ ,  $D_t = \{\mathbf{x}_t^i\}_{i=1}^{n_t}$ .  
**Parameters:** learning rate  $\gamma_1$  and  $\gamma_2$ , epoch  $T_{\max}$ , start epoch  $T_s$ , iteration  $N_{\max}$ , class number  $K$ , tradeoff  $\lambda$  and  $\alpha$ , network parameter  $\theta_{F \circ G}$  and  $\theta_D$ .  
**Output:** the neural network  $F \circ G$ , namely the target domain classifier for  $D_t$ .

- 1: **Initialize**  $\theta_{F \circ G}$  and  $\theta_D$ ;
- 2: **for**  $T = 1, 2, \dots, T_{\max}$  **do**
- 3:   **Shuffle** the training set  $D_s$ ,  $\overline{D}_s$ ,  $D_t$ ;
- 4:   **for**  $N = 1, 2, \dots, N_{\max}$  **do**
- 5:     **Fetch** mini-batch  $d_s$ ,  $\overline{d}_s$ ,  $d_t$  from  $D_s$ ,  $\overline{D}_s$ ,  $D_t$ ;
- 6:     **Calculate**  $L_s(G, F, d_s)$  using Eq. (26);
- 7:     **Update**  $\theta_{F \circ G} = \theta_{F \circ G} - \gamma_1 \alpha \nabla L_s(G, F, d_s)$ ;
- 8:     **Divide**  $\overline{d}_s$  into  $\{\overline{d}_{s,k}\}_{k=1}^K$ ;
- 9:     **Calculate**  $\{\overline{L}_s(G, F, \overline{d}_{s,k})\}_{k=1}^K$  using Eq. (20), and  $\overline{L}_s(G, F, \overline{d}_s)$  using Eq. (21);
- 10:    **if**  $\min_k \{\overline{L}_s(G, F, \overline{d}_{s,k})\}_{k=1}^K \geq 0$  **then**
- 11:     **Update**  $\theta_{F \circ G} = \theta_{F \circ G} - \gamma_1 (1 - \alpha) \nabla \overline{L}_s(G, F, \overline{d}_s)$ ;
- 12:    **else**
- 13:     **Calculate**  $\overline{L}_{\text{neg}} = \sum_{k=1}^K \min\{0, \overline{L}_s(G, F, \overline{d}_{s,k})\}$ ;
- 14:     **Update**  $\theta_{F \circ G} = \theta_{F \circ G} + \gamma_1 (1 - \alpha) \nabla \overline{L}_{\text{neg}}$ ;
- 15:    **end if**
- 16:    **if**  $T > T_s$  **then**
- 17:     **Calculate**  $L_{\text{adv}}(G, F, D, d_s, \overline{d}_s, d_t)$  using Eq. (28);
- 18:     **Update**  $\theta_D = \theta_D - \gamma_2 \nabla L_{\text{adv}}(G, F, D, d_s, \overline{d}_s, d_t)$ ;
- 19:     **Update**  $\theta_{F \circ G} = \gamma_2 \lambda \nabla L_{\text{adv}}(G, F, D, d_s, \overline{d}_s, d_t)$ ;
- 20:    **end if**
- 21:   **end for**
- 22: **end for**

---

of  $d_s$  and  $\overline{d}_s$ . The entire training procedures of CLARINET for PC-UDA are shown in Algorithm 3.

## VI. EXPERIMENTS

This section conducts extensive evaluations of CLARINET on several common transfer tasks against many state-of-the-art transfer-learning methods (e.g., two-step approach).

## A. Datasets and Tasks

We investigate seven image and digits datasets: *CIFAR-10* [68], *STL* [69], *MNIST* [70], *USPS* [71], *SVHN* [72], *MNIST-M* [73], and *SYN-DIGITS* [73]. We adopt the evaluation protocol of DANN [19], CDAN [20], asymmetric tri-training for unsupervised DA (ATDA) [12], and DIRT-T [74] with seven transfer tasks: *CIFAR-10* to *STL* ( $C \rightarrow T$ ), *MNIST* to *USPS* ( $M \rightarrow U$ ), *USPS* to *MNIST* ( $U \rightarrow M$ ), *SVHN* to *MNIST* ( $S \rightarrow M$ ), *MNIST* to *MNIST-M* ( $M \rightarrow m$ ), *SYN-DIGITS* to *MNIST* ( $Y \rightarrow M$ ), and *SYN-DIGITS* to *SVHN* ( $Y \rightarrow S$ ).

We train our model using the following training sets: *CIFAR-10* (45 000), *STL* (4500), *MNIST* (60 000), *USPS* (7438), *SVHN* (73 257), *MNIST-M* (59 001), *SYN-DIGITS* (479 400). Evaluation is reported on the standard test sets: *STL* (7200), *MNIST* (10 000), *USPS* (1860), *MNIST-M* (9001), *SVHN* (26 032) (the numbers of images are in parentheses).

Because all datasets carry TLs, following [41], we generate completely and partly complementary-label data. Generating complementary-label data is straightforward when the dataset is ordinary labeled, as it reduces to just choosing a class randomly other than true class.

## B. Baselines

We compare CLARINET with the following baselines: gradient ascent complementary label learning (GAC) [41], namely nontransfer method, and several two-step methods, which sequentially combine GAC with UDA methods (including DAN [17], DANN [19] and CDAN [20]). Thus, we have four possible baselines: GAC, GAC+DAN, GAC+DANN, and GAC+CDAN. For two-step methods, they share the same pseudo-label source data on each task. Note that, in this article, we use the entropy-conditioning variant of CDAN (CDAN\_E).

## C. Experimental Setup

We follow the standard protocols for UDA and compare the average classification accuracy based on five random experiments. For each experiment, we take the result of the last epoch.

The batch size is set to 128 and we train 500 epochs. Stochastic gradient descent (SGD) optimizer (momentum = 0.9, weight\_decay =  $5e - 5$ ) is with an initial learning rate of 0.005 in the adversarial network and  $5e - 5$  in the classifier. In the sharpening function  $T$ ,  $l$  is set to 0.5. For other special parameters in baselines, we all follow the original setting. We implement all methods with default parameters by PyTorch. The code of CLARINET is available at [github.com/Yiyang98/BFUDA](https://github.com/Yiyang98/BFUDA).

## D. Results on CC-UDA Tasks

Table I reports the target-domain accuracy of five methods on seven CC-UDA tasks. As can be seen, our CLARINET performs best on each task and the average accuracy of CLARINET is significantly higher than those of baselines. Compared with GAC method, CLARINET successfully transfers knowledge from complementary-label source data to unlabeled target data. Because CDAN has shown much better adaptation performance than DANN and DAN [20], GAC+CDAN should outperform other two-step methods on each task. However, on the  $U \rightarrow M$  task, the accuracy of GAC+CDAN is much lower than that of GAC+DANN. This abnormal phenomenon shows that the noise contained in pseudo-label source data significantly reduces transferability of existing UDA methods. Namely, we cannot obtain the reliable adaptation performance by using two-step CC-UDA approach.

*CIFAR-10*  $\rightarrow$  *STL*: CIFAR-10 and STL are ten-class object recognition datasets. We remove the nonoverlapping classes (“frog” and “monkey”) and readjust the labels to align the two datasets. Namely, this task is reduced to a nine-class classification problem. Furthermore, we downscale the  $96 \times 96$  image dimension of STL to match the  $32 \times 32$  dimension of CIFAR-10. As shown in Fig. 4(a), two-step methods could hardly transfer knowledge, whereas our CLARINET’s performance surpasses others by a comfortable margin.

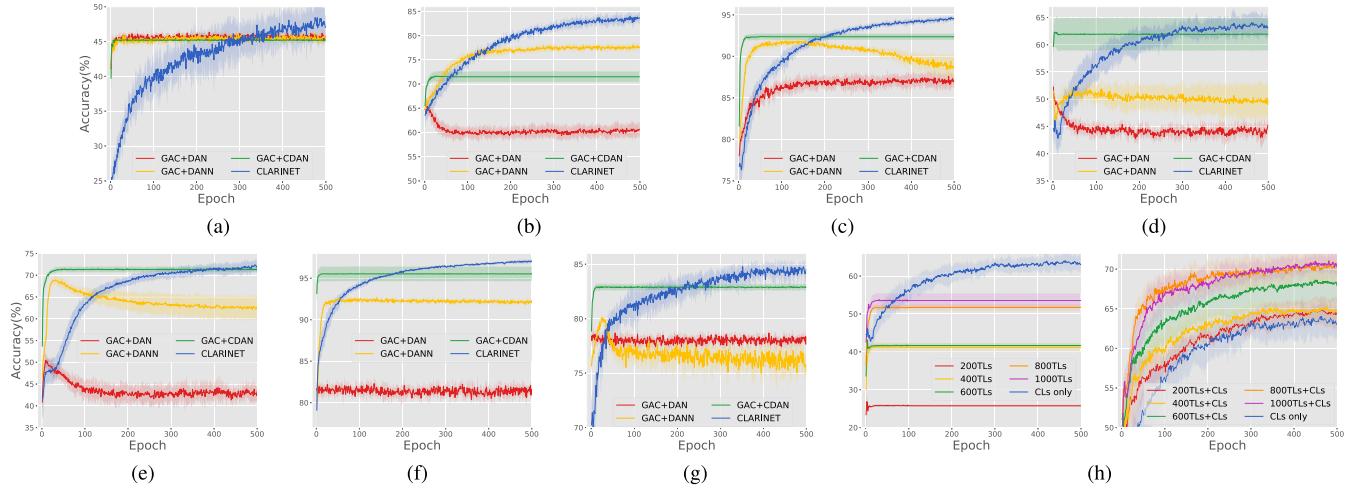


Fig. 4. Test Accuracy versus Epochs on seven CC-UDA Tasks in (a)–(g), and TL versus CL in (h). In (a)–(g), we compare the target-domain accuracy of one-step approach, i.e., CLARINET (*ours*), with that of two-step approach (*ours*). In (h), “200TLs” represents ordinary UDA method trained with 200 true-label source data. “200TLs+CLs” means a CLARINET trained with 200 true-label source data and complementary-label source data and “CLs Only” represents a CLARINET trained with complementary-label source data: (a) CIFAR-10 → STL; (b) USPS → MNIST; (c) MNIST → USPS; (d) SVHN → MNIST; (e) MNIST → MNIST-M; (f) SYN-DIGITS → MNIST; (g) SYN-DIGITS → SVHN; and (h) TL versus CL.

TABLE I

RESULTS ON SEVEN CC-UDA TASKS. BOLD VALUE REPRESENTS THE HIGHEST ACCURACY (%) ON EACH ROW. PLEASE NOTE, THE TWO-STEP METHODS AND CLARINET ARE ALL FIRST PROPOSED IN OUR ARTICLE

Tasks	GAC	Two-step approaches ( <i>ours</i> )			CLARINET ( <i>ours</i> )
		GAC+DAN	GAC+DANN	GAC+CDAN_E	
$C \rightarrow T$	45.167	$45.711 \pm 0.535$	$45.628 \pm 0.572$	$45.228 \pm 0.270$	<b><math>47.083 \pm 1.395</math></b>
$U \rightarrow M$	51.860	$60.692 \pm 1.300$	$77.580 \pm 0.770$	$71.498 \pm 1.077$	<b><math>83.692 \pm 0.928</math></b>
$M \rightarrow U$	77.796	$87.215 \pm 0.603$	$88.688 \pm 1.280$	$92.366 \pm 0.365$	<b><math>94.538 \pm 0.292</math></b>
$S \rightarrow M$	39.260	$45.132 \pm 1.363$	$50.882 \pm 2.440$	$61.922 \pm 2.983$	<b><math>63.070 \pm 1.990</math></b>
$M \rightarrow m$	45.045	$43.346 \pm 2.224$	$62.273 \pm 2.261$	$71.379 \pm 0.620$	<b><math>71.717 \pm 1.262</math></b>
$Y \rightarrow M$	77.070	$81.150 \pm 0.591$	$92.328 \pm 0.138$	$95.532 \pm 0.873$	<b><math>97.040 \pm 0.212</math></b>
$Y \rightarrow S$	72.480	$78.270 \pm 0.311$	$75.147 \pm 1.401$	$82.878 \pm 0.278$	<b><math>84.499 \pm 0.537</math></b>
Average	58.383	63.074	70.361	74.400	<b>77.377</b>

**$MNIST \leftrightarrow USPS$ :** MNIST and USPS are both grayscale digits images, thus the distribution discrepancy between the two tasks is relatively small. As shown in Fig. 4(b) and (c), in both adaptation directions, CLARINET all achieve the best performance far above other baselines.

**$SVHN \rightarrow MNIST$ :** SVHN and MNIST are both digit datasets. Whereas MNIST consists of black-and-white handwritten digits, SVHN consists of crops of colored, street house numbers. MNIST has a lower image dimensionality than SVHN, thus we adopt the dimension of MNIST to  $32 \times 32$  with three channels to match SVHN. Because of the above factors, the gap between two distributions are relatively larger compared to that of the  $MNIST \leftrightarrow USPS$ . As shown in Fig. 4(d), GAC+CDAN performs much better than GAC+DAN and GAC+DANN, but still worse than our CLARINET.

**$MNIST \rightarrow MNIST - M$ :** MNIST-M is a transformed dataset from MNIST, which is composed of merging clips of a background from the BSDS500 datasets [75]. For a human, the classification task on MNIST-M only becomes slightly harder, whereas for a convolutional neural network (CNN) trained on MNIST, this domain is quite different, as the background and the strokes are no longer constant. As shown in Fig. 4(e), our method is slightly more effective

than GAC+CDAN and far more effective than the other two methods.

**$SYN - DIGITS \rightarrow MNIST$ :** This adaptation reflects a common adaptation problem of transferring from synthetic images to real images. The SYN-DIGITS dataset consists of a huge amount of data, generated from Windows fonts by varying the text, positioning, orientation, background, stroke color, and amount of blur. As shown in Fig. 4(f), our method outperforms other baselines and achieves pretty high accuracy. Thus, with sufficient source data, CLARINET could achieve excellent results.

**$SYN - DIGITS \rightarrow SVHN$ :** This adaptation is another common adaptation problem of transferring from synthetic images to real images, but is more challenging than in the case of the MNIST experiment. As shown in Fig. 4(g), our method is obviously more effective than other baselines. GAC+DANN does not apply to this task, achieving the lowest accuracy.

### E. Results on PC-UDA Tasks

Table II reports the target-domain accuracy of CLARINET on PC-UDA tasks with different amounts of true-label source data. “true only” means training on a certain number of

TABLE II

RESULTS ON SEVEN PC-UDA TASKS. AMOUNT REPRESENTS THE NUMBER OF TRUE-LABEL DATA IN THE SOURCE DOMAIN. IN GENERAL, THE ACCURACY OF CLARINET INCREASES WHEN INCREASING THE AMOUNT OF TRUE-LABEL SOURCE DATA

Tasks	Amount of true-label source data					
	0		200		400	
	true only	com only	true only	com+true	true only	com+true
$C \rightarrow T$	-	47.083±1.395	11.839±0.019	49.408±1.776	13.875±1.366	49.553±1.362
$U \rightarrow M$	-	83.692±0.928	74.180±1.218	88.584±1.040	79.200±0.837	89.480±1.660
$M \rightarrow U$	-	94.538±0.292	78.011±1.473	93.204±1.398	83.204±1.545	94.677±0.576
$S \rightarrow M$	-	63.070±1.990	25.772±0.146	64.734±2.096	41.232±1.089	64.912±0.928
$M \rightarrow m$	-	71.717±1.262	59.414±1.381	70.730±1.620	59.805±0.951	71.198±0.623
$Y \rightarrow M$	-	97.040±0.212	49.232±1.354	97.182±0.383	60.640±1.570	97.242±0.117
$Y \rightarrow S$	-	84.499±0.537	23.009±1.102	84.269±0.814	49.120±1.236	85.538±0.596
Average	-	77.377	45.922	78.302	55.297	78.943
Tasks	600			800		1000
	true only	com+true	true only	com+true	true only	com+true
	17.722±2.626	50.897±0.969	19.278±0.853	51.058±1.737	20.972±1.061	53.297±1.655
$U \rightarrow M$	82.532±0.859	90.358±1.938	85.800±0.621	91.106±0.561	88.184±1.280	93.342±1.294
$M \rightarrow U$	83.925±1.511	94.839±0.254	85.839±2.074	94.796±0.104	85.699±0.777	95.022±0.280
$S \rightarrow M$	41.680±0.525	67.898±1.625	51.652±0.850	70.416±1.819	53.500±1.872	70.446±1.358
$M \rightarrow m$	63.757±1.344	72.732±0.947	65.161±0.766	73.050±1.264	68.522±1.285	73.336±0.727
$Y \rightarrow M$	76.802±1.649	97.178±0.396	85.286±1.363	96.842±0.267	86.470±1.646	96.948±0.266
$Y \rightarrow S$	67.922±1.079	85.921±1.098	67.788±1.878	86.772±0.291	74.654±1.054	87.024±0.542
Average	62.049	79.975	65.829	80.577	68.286	81.345

TABLE III

ABLATION STUDY. BOLD VALUE REPRESENTS THE HIGHEST ACCURACY (%) ON EACH COLUMN. OBVIOUSLY TO SEE, UDA METHODS CANNOT HANDLE COMPLEMENTARY LABEL-BASED UDA TASKS DIRECTLY. WE ALSO PROVE THAT THE CONDITIONING ADVERSARIAL PART AND THE SHARPENING FUNCTION  $T$  CAN HELP IMPROVE THE ADAPTATION PERFORMANCE

Methods	$C \rightarrow T$	$U \rightarrow M$	$M \rightarrow U$	$S \rightarrow M$	$M \rightarrow m$	$Y \rightarrow M$	$Y \rightarrow S$	Average
C w/ $L_{CE}$	6.481±2.536	0.455±0.722	0.055±0.129	3.708±0.688	7.088±0.424	1.832±0.102	1.298±0.070	2.987
C w/o $c$	41.908±2.796	<b>84.302±1.127</b>	93.301±0.465	44.500±2.088	70.994±0.749	94.382±0.150	83.408±0.545	73.256
C w/o $T$	43.075±2.553	83.192±1.796	93.419±0.588	52.438±1.927	<b>72.128±1.569</b>	95.442±1.004	83.055±0.652	74.678
CLARINET	<b>47.083±1.395</b>	83.692±0.928	<b>94.538±0.292</b>	<b>63.070±1.990</b>	71.717±1.262	<b>97.040±0.212</b>	<b>84.499±0.537</b>	<b>77.377</b>

true-label source data with ordinary UDA method. “com only” means training on complementary-label source data only with CLARINET. “com+true” stands for training on a certain number of true-label source data and complementary-label source data with CLARINET. In general, the accuracy of CLARINET increases when increasing the amount of true-label source data from 0 to 1000. Thus, it is proved that CLARINET can sufficiently leverage true-label source data to improve adaptation performance.

The improvement is especially evident on  $U \rightarrow M$  task and  $S \rightarrow M$  task. For  $U \rightarrow M$  task, this is probably because the dataset sample size of USPS is relatively small, true-label data actually has occupied a big part. For  $S \rightarrow M$  task, SVHN is complicated for complementary-label learning. Hence, adding a small amount of true-label data could help to train a more accurate classifier. This phenomenon also reminds us that for complex datasets, adding some true-label data to assist training would be pretty appropriate. On  $Y \rightarrow M$  task, adding true-label source data does not bring significant improvement, which is most likely because the result on complementary-label data is already relatively good and true-label source data is unable to assist in achieving better result.

We also compare the efficacy of true-label source data with complementary-label source data. Taking  $S \rightarrow M$  task as an example [as shown in the left part of Fig. 4(h)], we compare the target-domain accuracy of ordinary UDA method trained with different amount of true-label source data and that of CLARINET trained with complementary-label source data only (CLs Only). The accuracy decreases significantly when reducing the amount of true-label source data, which suggests that sufficient true-label source data are inevitably required in UDA scenario. Then we compare the target-domain accuracy of CLARINET trained with complementary-label source data only with that of CLARINET trained with different amount of true-label and complementary-label source data. It is clear that CLARINET effectively uses two kinds of data to obtain better adaptation performance than using complementary-label source data only. Besides, as the number of true-label source data used increases, the classification accuracy becomes higher [as shown in the right part of Fig. 4(h)].

#### F. Analysis

1) *Labeling Cost*: From a theoretical analysis, the information carried by the TL is  $K - 1$  times that of the CL.

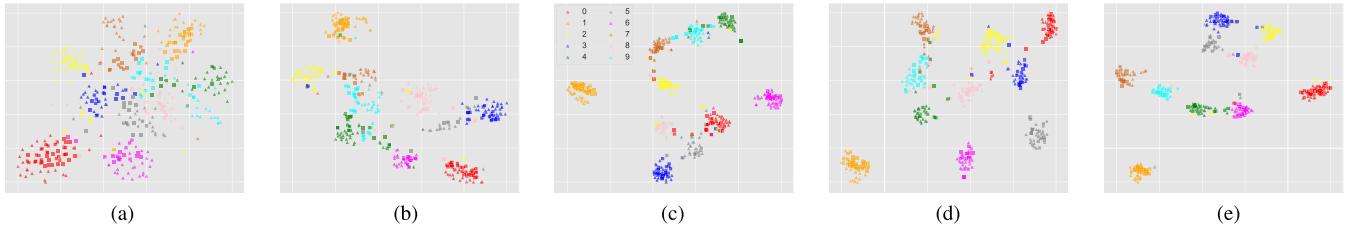


Fig. 5. Feature visualization of target and source features on  $M \rightarrow U$  task.  $\triangle$  indicates source samples.  $\square$  indicates target samples. Different colors indicate different classes. (a) GAC. (b) GAC+DAN. (c) GAC+DANN. (d) GAC+CDAN. (e) CLARINET.

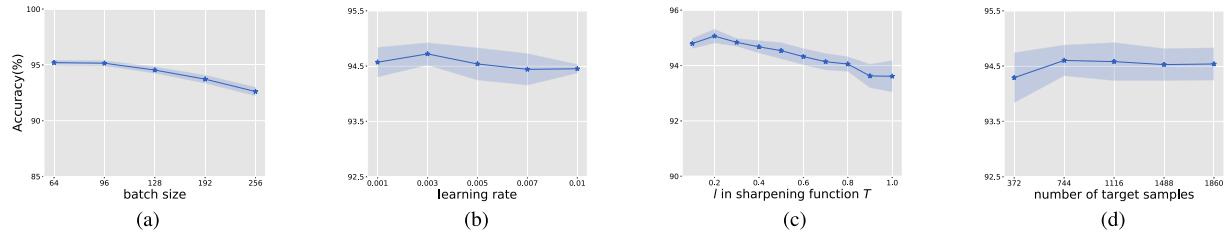


Fig. 6. Parameter analyses. Hyperparameter sensitivity studies are carried out to find their effect on the performance. Experiments are conducted on  $M \rightarrow U$  task. In (a)–(d), we show how these four parameters influence the test accuracy.

We conduct experiments and prove the ratio is actually far less than  $K - 1$  when obtaining the same result, which means using CL is low cost. More detailed analysis can be found in Appendix C.

2) *Ablation Study*: We conduct experiments to show the contributions of different components in CLARINET. We consider following baselines:

- 1) C w/  $L_{CE}$ : train CLARINET by Algorithm 2, while replacing  $\overline{L}_s(G, F, \overline{D}_s)$  by cross-entropy loss.
- 2) C w/o  $c$ : train CLARINET without conditioning, namely train the domain discriminator  $D$  only based on feature representations  $g_s$  and  $g_t$ .
- 3) C w/o  $T$ : train CLARINET by Algorithm 2, without sharpening function  $T$ .

C w/  $L_{CE}$  uses the cross-entropy loss to take place of complementary-label loss. Actually, it stands for applying ordinary UDA methods directly on complementary label-based UDA tasks. The target-domain accuracy of C w/  $L_{CE}$  will show whether UDA methods can address the complementary label-based UDA problem. C w/o  $c$  train the domain discriminator  $D$  only based on feature representations  $g_s$  and  $g_t$ , thus the result could indicate whether the conditional adversarial way could capture the multimodal structures so as to improve the transfer effect. Please notice, the sharpening function  $T$  is useless in this network as it works on the label prediction  $f_s$  and  $f_t$ . Comparing CLARINET with C w/o  $T$  reveals whether the sharpening function  $T$  takes effect.

As summarized in Table III, the target-domain accuracy of C w/  $L_{CE}$  is much lower than that of other methods. Namely, UDA methods cannot handle complementary label-based UDA tasks directly. Its result is not even as good as random classification, as the network is trained taking the wrong label as the target result. Compared with C w/o  $T$ , C w/o  $c$  has a worse performance, which proves that the conditional adversarial way could really improve the transfer effect. Therefore, it is

necessary to capture the multimodal structures of distributions with cross-covariance dependence between the features and classes in the field of adversarial based UDA. Although C w/o  $T$  achieves better accuracy than other baselines, its accuracy is still worse than CLARINET's. The result reveals that the sharpening function  $T$  helps to capture multimodal structures of distributions on the basis of the characteristics of complementary-label learning. Thus, the sharpening function  $T$  can improve the adaptation performance.

3) *Visualization*: To intuitively demonstrate the effect of our method, we show the feature visualization of source and target domains by  $t$ -distributed stochastic neighbor embedding ( $t$ -SNE) [76], which is an effective dimensionality reduction method. Fig. 5 shows the effect of DA of all the baselines and our method in  $M \rightarrow U$  task. Clearly, our method outperforms baselines in aligning the distributions of two domains.

In Fig. 5(a), although the samples of two domains tend to gather in general, they are not actually tightly clustered together, and the class 9 (cyan) is an obvious example. As GAC is a nontransfer method, it could be used to compare the effect of before-after domain discrepancy with other methods. In Fig. 5(e), it is clear that the intraclass centroids of two domains are closer than other baselines and there are fewer error-clustered samples, namely CLARINET does better in aligning the distributions of source and target domains.

4) *Parameter Analysis*: To investigate their effect on the performance, we carry out hyperparameter sensitivity studies. Taking  $M \rightarrow U$  task, for example, we conduct experiments to demonstrate the effect of four parameters for our method, including batch size, learning rate,  $l$  in sharpening function  $T$ , and the number of unlabeled target domain samples.

As shown in Fig. 6, the accuracy of CLARINET decreases in general when increasing the batch size from 64 to 256. We increase the learning rate from 0.001 to 0.01, and it achieves the best result at 0.003. The  $l$  in sharpening function

$T$  is an important parameter for CLARINET, as the sharpening function  $T$  could effectively help improve the adaptation performance. It could be seen that when  $l$  is small, namely the output of sharpening function  $T$  approaches an one-hot distribution, CLARINET could achieve a good result in this task. We also test how the number of target samples would affect the performance. As shown in Fig. 6(d), when sufficient samples are available, the accuracy stays stable.

## VII. CONCLUSION

This article presents a new setting, complementary label-based UDA, which exploits economical complementary-label source data instead of expensive true-label source data. We consider two cases of the complementary label-based UDA: one is that the source domain only contains complementary-label data (CC-UDA) and the other is that the source domain has plenty of complementary-label data and a small amount of true-label data (PC-UDA). Because existing UDA methods cannot address the complementary label-based UDA problem, we propose a novel one-step approach called CLARINET. CLARINET can handle both CC-UDA and PC-UDA tasks. Experiments conducted on seven complementary label-based UDA tasks confirm that CLARINET effectively achieves distributional adaptation from complementary-label source data to unlabeled target data and outperforms a series of competitive baselines.

In the future, we plan to explore more effective ways to solve complementary label-based UDA and extend the application of CLs in DA. For example, instead of requiring the source domain and the target domain to share the same label set, we could apply CLs in the open set DA scenario in which the target domain contains unknown classes that are not observed in the source domain.

## REFERENCES

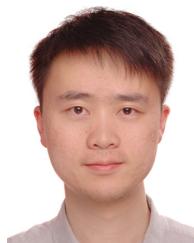
- [1] Y. Luo, Y. Wen, T. Liu, and D. Tao, "Transferring knowledge fragments for learning distance metric from a heterogeneous domain," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 1013–1026, Apr. 2019.
- [2] M. Xiao and Y. Guo, "Feature space independent semi-supervised domain adaptation via kernel matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 54–66, Jan. 2015.
- [3] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," in *Proc. ICML*, 2013, pp. 819–827.
- [4] L. Zhong, Z. Fang, F. Liu, J. Lu, B. Yuan, and G. Zhang, "How does the combined risk affect the performance of unsupervised domain adaptation approaches?" in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1–9.
- [5] S. Mottian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5715–5725.
- [6] S. Sukhija, N. Krishnan, and G. Singh, "Supervised heterogeneous domain adaptation via random forests," in *Proc. IJCAI*, 2016, pp. 2039–2045.
- [7] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4068–4076.
- [8] L. Cheng and S. J. Pan, "Semi-supervised domain adaptation on manifolds," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2240–2249, Dec. 2014.
- [9] S. Mehrkanoon and J. A. K. Suykens, "Regularized semipaired kernel CCA for domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3199–3213, Jul. 2018.
- [10] K. A. Islam, V. Hill, B. Schaeffer, R. Zimmerman, and J. Li, "Semi-supervised adversarial domain adaptation for seagrass detection in multispectral images," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2019, pp. 1120–1125.
- [11] P. Wei, Y. Ke, and C. K. Goh, "Feature analysis of marginalized stacked denoising autoencoder for unsupervised domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1321–1334, May 2019.
- [12] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," in *Proc. ICML*, 2017, pp. 2988–2997.
- [13] Y. Cao, M. Long, and J. Wang, "Unsupervised domain adaptation with distribution matching machines," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 2795–2802.
- [14] Z. Deng, Y. Luo, and J. Zhu, "Cluster alignment with a teacher for unsupervised domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9944–9953.
- [15] S. Zhao, H. Fu, M. Gong, and D. Tao, "Geometry-aware symmetric domain adaptation for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9788–9798.
- [16] G. Agresti, H. Schaefer, P. Sartor, and P. Zanuttigh, "Unsupervised domain adaptation for ToF data denoising with adversarial learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5584–5593.
- [17] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. ICML*, 2015, pp. 97–105.
- [18] M. Long, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. ICML*, 2017, pp. 2208–2217.
- [19] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1–35, Jan. 2016.
- [20] M. Long, Z. Cao, J. Wang, and M. Jordan, "Conditional adversarial domain adaptation," in *Proc. NIPS*, 2018, pp. 1640–1650.
- [21] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf, "Domain adaptation with conditional transferable components," in *Proc. ICML*, 2016, pp. 2839–2848.
- [22] K. Zhang, M. Gong, and B. Schölkopf, "Multi-source domain adaptation: A causal view," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 3150–3157.
- [23] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8503–8512.
- [24] T. Ishida, G. Niu, W. Hu, and M. Sugiyama, "Learning from complementary labels," in *Proc. NIPS*, 2017, pp. 5639–5649.
- [25] X. Yu, T. Liu, M. Gong, and D. Tao, "Learning with biased complementary labels," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 69–85.
- [26] F. Liu, J. Lu, B. Han, G. Niu, G. Zhang, and M. Sugiyama, "Butterfly: A panacea for all difficulties in wildly unsupervised domain adaptation," in *Proc. NIPS LTS Workshop*, 2019, pp. 1–23.
- [27] Y. Shu, Z. Cao, M. Long, and J. Wang, "Transferable curriculum for weakly-supervised domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 4951–4958.
- [28] Y. Zhang, F. Liu, Z. Fang, B. Yuan, G. Zhang, and J. Lu, "Clarinet: A one-step approach towards budget-friendly unsupervised domain adaptation," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 1–8.
- [29] L. Song, J. Huang, A. Smola, and K. Fukumizu, "Hilbert space embeddings of conditional distributions with applications to dynamical systems," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 961–968.
- [30] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 999–1006.
- [31] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066–2073.
- [32] J. Hoffman *et al.*, "LSDA: Large scale detection through adaptation," in *Proc. NIPS*, 2014, pp. 3536–3544.
- [33] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Nov. 2011.
- [34] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. ICML*, 2011, pp. 513–520.
- [35] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Proc. NIPS*, 2007, pp. 601–608.

- [36] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.
- [37] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
- [38] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. NIPS*, 2014, pp. 3320–3328.
- [39] F. Liu, W. Xu, J. Lu, G. Zhang, A. Gretton, and D. J. Sutherland, "Learning deep kernels for non-parametric two-sample tests," in *Proc. ICML*, 2020, pp. 6316–6326.
- [40] 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. (CVPR)*, Jul. 2017, pp. 3722–3731.
- [41] T. Ishida, G. Niu, A. Menon, and M. Sugiyama, "Complementary-label learning for arbitrary losses and models," in *Proc. ICML*, 2019, pp. 2971–2980.
- [42] K.-H. Lee, X. He, L. Zhang, and L. Yang, "CleanNet: Transfer learning for scalable image classifier training with label noise," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5447–5456.
- [43] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting image databases from the Web," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 754–766, Apr. 2011.
- [44] Y. Zhang, T. Liu, M. Long, and M. Jordan, "Bridging theory and algorithm for domain adaptation," in *Proc. ICML*, 2019, pp. 7404–7413.
- [45] Z. Fang, J. Lu, F. Liu, J. Xuan, and G. Zhang, "Open set domain adaptation: Theoretical bound and algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 3, 2020, doi: 10.1109/TNNLS.2020.3017213.
- [46] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1414–1430, Jul. 2017.
- [47] A. Kumar, P. Sattigeri, and T. Fletcher, "Semi-supervised learning with GANs: Manifold invariance with improved inference," in *Proc. NIPS*, 2017, pp. 5534–5544.
- [48] B. Freney and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2014.
- [49] Y.-F. Li and Z.-H. Zhou, "Towards making unlabeled data never hurt," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 175–188, Jan. 2015.
- [50] T. Sakai, M. C. du Plessis, G. Niu, and M. Sugiyama, "Semi-supervised classification based on classification from positive and unlabeled data," in *Proc. ICML*, 2017, pp. 2998–3006.
- [51] B. Han *et al.*, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8527–8537.
- [52] B. Han *et al.*, "Masking: A new perspective of noisy supervision," in *Proc. NIPS*, 2018, pp. 5836–5846.
- [53] Q. Miao, Y. Cao, G. Xia, M. Gong, J. Liu, and J. Song, "RBoost: Label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2216–2228, Nov. 2016.
- [54] E. Xing, M. Jordan, S. Russell, and A. Ng, "Distance metric learning with application to clustering with side-information," in *Proc. NIPS*, 2003, pp. 521–528.
- [55] T. F. Covoes, E. R. Hruschka, and J. Ghosh, "Competitive learning with pairwise constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 1, pp. 164–169, Jan. 2013.
- [56] J. Goldberger, G. Hinton, S. Roweis, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. NIPS*, 2005, pp. 513–520.
- [57] M. Du Plessis, G. Niu, and M. Sugiyama, "Analysis of learning from positive and unlabeled data," in *Proc. NIPS*, 2014, pp. 703–711.
- [58] G. Niu, M. C. du Plessis, T. Sakai, Y. Ma, and M. Sugiyama, "Theoretical comparisons of positive-unlabeled learning against positive-negative learning," in *Proc. NIPS*, 2016, pp. 1199–1207.
- [59] C. Gong, T. Liu, J. Yang, and D. Tao, "Large-margin label-calibrated support vector machines for positive and unlabeled learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3471–3483, Nov. 2019.
- [60] R. Kiryo, G. Niu, M. C. du Plessis, and M. Sugiyama, "Positive-unlabeled learning with non-negative risk estimator," in *Proc. NIPS*, 2017, pp. 1675–1685.
- [61] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [62] T. Ma, "Generalization and equilibrium in generative adversarial nets (GANs) (invited talk)," in *Proc. 50th Annu. ACM SIGACT Symp. Theory Comput.*, Jun. 2018, pp. 224–232.
- [63] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: <https://arxiv.org/abs/1411.1784>
- [64] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. ICML*, 2017, pp. 2642–2651.
- [65] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [66] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Proc. NIPS*, 2019, pp. 5050–5060.
- [67] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. ICML*, 2003, pp. 912–919.
- [68] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images." M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.
- [69] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.
- [70] 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.
- [71] T. Hastie, J. Friedman, and R. Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer Series in Statistics). Springer, 2001.
- [72] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS DLUFL Workshop*, 2011, pp. 1–9.
- [73] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, 2015, pp. 1180–1189.
- [74] X. Guo, W. Chen, and J. Yin, "A simple approach for unsupervised domain adaptation," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 1566–1570.
- [75] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.
- [76] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.



**Yiyang Zhang** received the B.E. degree in automation from Tsinghua University, Shenzhen, China, in 2018. She is currently pursuing the master's degree in data science with the Shenzhen International Graduate School, Tsinghua University.

She is a Visiting Student with the Australian Artificial Intelligence Institute (AAII), University of Technology Sydney (UTS), Sydney, NSW, Australia. Her research interests include transfer learning and domain adaptation.



**Feng Liu** (Member, IEEE) received the B.Sc. degree in pure mathematics and the M.Sc. degree in probability and statistics from the School of Mathematics and Statistics, Lanzhou University, Lanzhou, China, in 2013 and 2015, respectively, and the Ph.D. degree in computer science from Australian Artificial Intelligence Institute (AAII), University of Technology Sydney, Sydney, NSW, Australia, in 2020.

He is a Lecturer with AAII University of Technology Sydney. His research interests include machine learning and hypothesis testing.

Dr. Liu has served as a Senior Program Committee Member for ECAI and as Program Committee Member for NeurIPS, ICML, ICLR, AISTATS, ACMIL, IJCAI, AAAI, and so on. He was a recipient of the ICLR outstanding reviewer award (2021), the Best paper award from AAII, UTS (2020), the UTS-FEIT HDR Research Excellence Award (2019), the Best Student Paper Award of FUZZ-IEEE (2019), and the UTS Research Publication Award (2018). He serves as a Reviewer for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), the IEEE TRANSACTIONS ON FUZZY SYSTEMS (TFS), and the *Machine Learning Journal (MLJ)*.



**Zhen Fang** (Student Member, IEEE) received the M.Sc. degree in pure mathematics from the School of Mathematical Sciences, Xiamen University, Xiamen, China, in 2017. He is currently pursuing the Ph.D. degree with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia.

He is a Member of the Decision Systems and e-Service Intelligence (DeSI) Research Laboratory, AAII, University of Technology Sydney. His research interests include transfer learning and domain adaptation.



**Guangquan Zhang** received the Ph.D. degree in applied mathematics from the Curtin University of Technology, Bentley, WA, Australia, in 2001.

He is an Associate Professor and the Director of the Decision Systems and e-Service Intelligent (DeSI) Research Laboratory, Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, NSW, Australia. He has authored or coauthored five monographs, five textbooks, and 470 articles, including 225 refereed international journal articles. His research interests include fuzzy modeling in machine learning and data analytics.

Dr. Zhang was a recipient of the nine Australian Research Council (ARC) Discovery Project and many other research grants. He was awarded an ARC QEII Fellowship in 2005. He has served in Editorial Boards of several international journals, and as a Guest Editor of eight special issues for IEEE TRANSACTIONS and other international journals.



**Bo Yuan** (Senior Member, IEEE) received the B.E. degree in computer science from the Nanjing University of Science and Technology, Nanjing, China, in 1998, and the M.Sc. and Ph.D. degrees in computer science from The University of Queensland (UQ), St Lucia, QLD, Australia, in 2002 and 2006, respectively.

From 2006 to 2007, he was a Research Officer on a project funded by the Australian Research Council, UQ. He is currently an Associate Professor with the Division of Informatics, Shenzhen International Graduate School, Tsinghua University, Shenzhen, China. He has authored or coauthored more than 110 articles in refereed international conferences and journals. His research interests include data science, evolutionary computation, and reinforcement learning.



**Jie Lu** (Fellow, IEEE) received the Ph.D. degree from Curtin University, Bentley, WA, Australia, in 2000.

She is a Distinguished Professor and the Director of the Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, NSW, Australia. She has published six research books and 450 articles in IEEE TRANSACTIONS and other journals and conferences. She has completed over 20 Australian Research Council discovery grants and other research projects. Her main research expertise is in fuzzy transfer learning, decision support systems, concept drift and recommender systems.

Dr. Lu is an IFSA Fellow and Australian Laureate Fellow. She serves as Editor-In-Chief for *Knowledge-Based Systems* (Elsevier) and Editor-In-Chief for the *International Journal on Computational Intelligence Systems* (Atlantis), has delivered 25 keynote speeches at international conferences, and has chaired ten international conferences.