Clarinet: A One-step Approach Towards Budget-friendly Unsupervised Domain Adaptation

Yiyang Zhang[1,2,*] , **Feng Liu**[2,*] , **Zhen Fang**[2] , **Bo Yuan**[1] , **Guangquan Zhang**[2] and **Jie Lu**[2,†]
[1]Shenzhen International Graduate School, Tsinghua University
[2]Centre for Artificial Intelligence, University of Technology Sydney
zhangyiy18@mails.tsinghua.edu.cn, feng.liu@uts.edu.au, zhen.fang@student.uts.edu.au,
yuanb@sz.tsinghua.edu.cn, {guangquan.zhang, jie.lu}@uts.edu.au

Abstract

In *unsupervised domain adaptation* (UDA), classifiers for the target domain are trained with massive *true-label* data from the source domain and unlabeled data from the target domain. However, it may be difficult to collect fully-true-label data in a source domain given limited budget. To mitigate this problem, we consider a novel problem setting where the classifier for the target domain has to be trained with *complementary-label* data from the source domain and unlabeled data from the target domain named *budget-friendly UDA* (BFUDA). The key benefit is that it is much less costly to collect *complementary-label* source data (required by BFUDA) than collecting the *true-label* source data (required by ordinary UDA). To this end, *complementary label adversarial network* (CLARINET) is proposed to solve the BFUDA problem. CLARINET maintains two deep networks simultaneously, where one focuses on classifying *complementary-label* source data and the other takes care of the source-to-target distributional adaptation. Experiments show that CLARINET significantly outperforms a series of competent baselines.

1 Introduction

Domain Adaptation (DA) aims to train a target-domain classifier with data in source and target domains [Yan *et al.*, 2017; Zhou *et al.*, 2019b]. 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 [Sukhija *et al.*, 2016], semi-supervised DA [Ao *et al.*, 2017; Zhou *et al.*, 2019a] and *unsupervised DA* (UDA) [Gong *et al.*, 2018; Liu *et al.*, 2020; Saito *et al.*, 2017; Fang *et al.*, 2019]. In practice, UDA methods have been applied to many real-world problems, such as object recognition [Agresti *et al.*, 2019; Deng *et al.*, 2019; Zhao *et al.*, 2019].

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 categorised into the following three groups: integral-probability-metrics based UDA [Long *et al.*, 2015]; adversarial-training based UDA [Ganin *et al.*, 2016; Long *et al.*, 2018]; and causality-based UDA [Gong *et al.*, 2016; Gong *et al.*, 2018; Zhang *et al.*, 2015]. Since adversarial-training based UDA methods extract better domain-invariant representations via deep networks, they usually have good target-domain accuracy [Sankaranarayanan *et al.*, 2018].

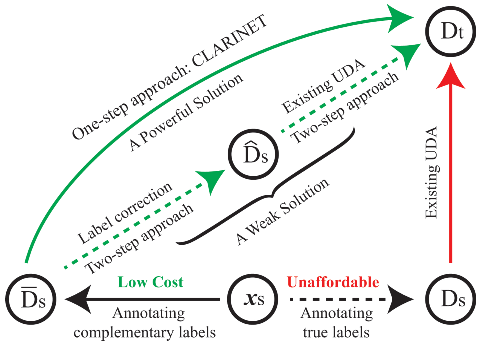


Figure 1: Budget-friendly unsupervised domain adaptation. The red line denotes that UDA methods transfer knowledge from D_s (true-label source data) to D_t (unlabeled target data). However, acquiring fully-true-label source data is *costly* and *unaffordable* (black dash line, $\mathbf{x}_s \to D_s$, $\mathbf{x}_s$ means unlabeled source data). This brings *budget-friendly unsupervised domain adaptation* (BFUDA), namely transferring knowledge from $\overline{D}_s$ (complementary-label source data) to D_t. It is much less costly to collect complementary-label source data (black line, required by BFUDA) than collecting the true-label one (black dash line, required by UDA). To handle BFUDA, a weak solution is a two-step approach (green dash line), which sequentially combines complementary-label learning methods ($\overline{D}_s \to \hat{D}_s$, label correction) and existing UDA methods ($\hat{D}_s \to D_t$). This paper proposes a one-step approach called *complementary label adversarial network* (CLARINET, green line, $\overline{D}_s \to D_t$ directly).

However, the success of UDA still highly relies on the scale of true-label source data. Namely, the target-domain accuracy of a UDA method (e.g., CDAN) decays when the scale of true-label source data decreases. Hence, massive true-label source data are inevitably required by UDA methods, which is very expensive and prohibitive especially when the budget

[*]Equal Contribution

[†]Corresponding Author

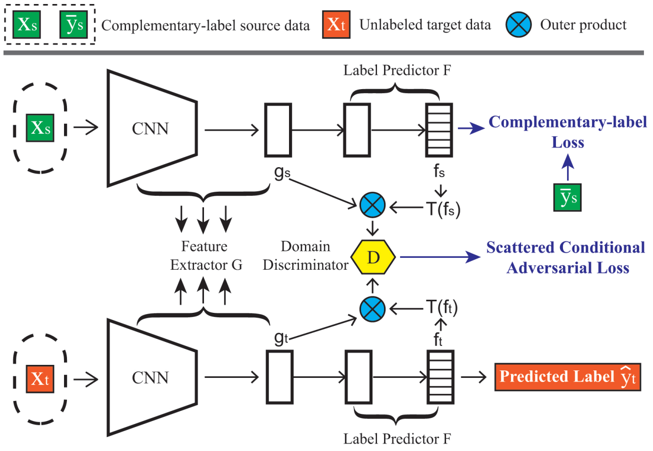


Figure 2: Overview of the proposed *complementary label adversarial network* (CLARINET). It consists of feature extractor G, label predictor F and conditional domain discriminator D. g_s and g_t are the outputs of G, representing the extracted features of source and target data. f_s and f_t represent classifier predictions. T is a mapping function which we propose to scatter the classifier predictions. In Algorithm 1, we show how to use two losses mentioned in this figure to train CLARINET.

is limited [Liu *et al.*, 2019]. This circumstance may hinder the promotion of DA to more areas.

While determining the correct label from many candidates is laborious, choosing one of the incorrect labels (i.e., complementary labels) would be much easier and thus less costly, especially when we have many candidates [Ishida *et al.*, 2017]. For example, suppose that we need to annotate labels of a bunch of animal images from $1,000$ candidates. One strategy is to ask crowd-workers to choose the true labels from $1,000$ candidates, while the other is to judge the correctness of a label randomly given by the system from the candidates. Apparently, the cost of the second strategy is much lower than that of the first one [Ishida *et al.*, 2019].

This brings us a novel problem setting, *budget-friendly UDA* (abbreviated as BFUDA), which aims to transfer knowledge from complementary-label source data to unlabeled target data (Figure 1). We describe this problem setting using the word *budget-friendly* since, compared to ordinary UDA, we can greatly save the labeling cost by annotating complementary labels in the source domain rather than annotating true labels [Ishida *et al.*, 2017; Yu *et al.*, 2018]. Please note that, existing UDA methods cannot handle BFUDA, as they require fully-true-label source data [Liu *et al.*, 2020; Saito *et al.*, 2017] or at least 20% true-label source data (demonstrated in [Liu *et al.*, 2019; Shu *et al.*, 2019]).

A straightforward but weak solution to BFUDA is a two-step approach, which sequentially combines *complementary-label learning* (CLL) methods and existing UDA methods (green dash line in Figure 1)[1]. CLL 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 domain-adaptation performance of such two-step approach [Liu *et al.*, 2019].

Therefore, we propose a powerful one-step solution to BFUDA, *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. [2018] and Song et. al. [2009] have shown that multimodal structures of distributions can only be captured sufficiently by the cross-covariance dependency between the features and classes (i.e., true labels), we set the input of domain discriminator D as the outer product of feature representation (e.g., $\boldsymbol{g_s}$ in Figure 2) and mapped classifier prediction (e.g., $\boldsymbol{T(f_s)}$ in Figure 2).

Due to the nature of complementary-label classification, predicted probability of each class (i.e., each element of $\boldsymbol{f_s}$, Figure 2) is relatively close. According to [Song *et al.*, 2009], this kind of predicted probabilities could not provide sufficient information to capture the multimodal structure of distributions. To fix it, we propose a mapping function T to make the predicted probabilities more scattered (i.e., $\boldsymbol{T(f_s)}$, Figure 2) than previous ones (i.e., $\boldsymbol{f_s}$, Figure 2). By doing so, the mapped 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 2) verifies that T indeed helps improve the target-domain accuracy.

We conduct experiments on 6 BFUDA tasks and compare CLARINET with a series of competent baselines. Empirical results demonstrated that CLARINET can effectively transfer knowledge from complementary-label source data to unlabeled target data and is superior to all baselines.

2 Budget-friendly Unsupervised Domain Adaptation

In this section, we propose a novel problem setting, called *budget-friendly unsupervised domain adaptation* (BFUDA), and prove a learning bound of this new problem. Then, we show how BFUDA brings benefits to domain adaptation field.

2.1 Problem Setting

Let $\mathcal{X} \subset \mathbb{R}^d$ be a feature (input) space and $\mathcal{Y} := \{1, ..., K\}$ be a label (output) space. A *domain* is defined as follows.

Definition 1 (Domains for BFUDA)**.** *Given random variables $X_s, X_t \in \mathcal{X}$, $Y_s, \overline{Y}_s, Y_t \in \mathcal{Y}$, the source and target domains are joint distributions $P(X_s, \overline{Y}_s)$ and $P(X_t, Y_t)$, where the joint distributions $P(X_s, Y_s) \neq P(X_t, Y_t)$ and $P(\overline{Y}_s = c|Y_s = c) = 0$ for all $c \in \mathcal{Y}$.*

Then, we propose BFUDA problem as follows.

Problem 1 (BFUDA)**.** *Given independent and identically distributed (i.i.d.) labeled samples $\overline{D}_s = \{(\mathbf{x}_s^i, \overline{y}_s^i)\}_{i=1}^{\overline{n}_s}$ drawn from the source domain $P(X_s, \overline{Y}_s)$ 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 BFUDA is to train a classifier $F : \mathcal{X} \to \mathcal{Y}$ with $\overline{D}_s$ and D_t such that F can accurately classify target data drawn from $P(X_t)$.*

[1] We implement this two-step approach and take it as a baseline.

It is clear that it is impossible to design a suitable learning procedure without any assumptions on $P(X_s, \overline{Y}_s)$. In this paper, we use the assumption for unbiased complementary learning proposed by [Ishida *et al.*, 2019; Ishida *et al.*, 2017]:

$$P(\overline{Y}_s = k|X_s) = \frac{1}{K-1} \sum_{c=1, c\neq k}^{K} P(Y_s = c|X_s), \qquad (1)$$

for all $k, c \in \mathcal{Y} = \{1, ..., K\}$ and $c \neq k$.

2.2 Learning Bound of BFUDA

This subsection presents a learning bound of BFUDA. Practitioner may safely skip it. The $\ell : \mathbb{R}^K \times \mathcal{Y} \to \mathbb{R}_+$ is a loss function. The *decision function* is a vector-valued function $\boldsymbol{h} : \mathcal{X} \to \mathbb{R}^K$ and $\boldsymbol{h}_k$ is the k-th element of $\boldsymbol{h}$. The complementary risk for $\boldsymbol{h}$ with respect to ℓ over $P(X_s, \overline{Y}_s)$ is

$$L_{\overline{s}}(\boldsymbol{h}) = \mathbb{E}\ell(\boldsymbol{h}(X_s), \overline{Y}_s).$$

The risks for the decision function $\boldsymbol{h}$ with respect to loss ℓ over implicit distribution $P(X_s, Y_s), P(X_t, Y_t)$ are:

$$L_s(\boldsymbol{h}) = \mathbb{E}\ell(\boldsymbol{h}(X_s), Y_s), L_t(\boldsymbol{h}) = \mathbb{E}\ell(\boldsymbol{h}(X_t), Y_t).$$

Then, we introduce our main theorem as follows.

Theorem 1. *Given a loss function ℓ and a hypothesis $\mathcal{H} \subset \{\boldsymbol{h} : \mathcal{X} \to \mathbb{R}^K\}$, then under unbiased assumption, for any $\boldsymbol{h} \in \mathcal{H}$, we have*

$$L_t(\boldsymbol{h}) \leq \overline{L}_s(\boldsymbol{h}) + \frac{1}{2} d^{\ell}_{\mathcal{H}}(P_{X_s}, P_{X_t}) + \Lambda,$$

where $\overline{L}_s(\boldsymbol{h}) := \sum_{k=1}^{K} \int_{\mathcal{X}} \ell(\boldsymbol{h}(\mathbf{x}), k) \mathrm{d}P_{X_s} - (K-1) L_{\overline{s}}(\boldsymbol{h})$, P_{X_s}, P_{X_t} are source and target marginal distributions, $\Lambda = \min_{\boldsymbol{h} \in \mathcal{H}} R_s(\boldsymbol{h}) + R_t(\boldsymbol{h})$, and $d^{\ell}_{\mathcal{H}}(P_{X_s}, P_{X_t})$ is the distribution discrepancy defined in [Ben-David et al.*, 2010].*

Proof. We firstly investigate the connection between $L_s(\boldsymbol{h})$ and $L_{\overline{s}}(\boldsymbol{h})$ under unbiased assumption in Eq. (1). Given $K \times K$ matrix Q whose diagonal elements are 0 and other elements are $1/K$, we represent the unbiased assumption by $\overline{\boldsymbol{\eta}} = Q\boldsymbol{\eta}$, where $\overline{\boldsymbol{\eta}} = [P(\overline{Y}_s = 1|X_s), ..., P(\overline{Y}_s = K|X_s)]^T$ and $\boldsymbol{\eta} = [P(Y_s = 1|X_s), ..., P(Y_s = K|X_s)]^T$. Note that Q has inverse matrix Q^{-1} whose diagonal elements are $-(K-2)$ and other elements are 1. Thus, we have that

$$Q^{-1}\overline{\boldsymbol{\eta}} = \boldsymbol{\eta}. \qquad (2)$$

According to Eq. (2), we have $P(Y_s = k|X_s) = 1 - (K-1)P(\overline{Y}_s = k|X_s)$, which implies that

$$L_s(\boldsymbol{h}) = \sum_{k=1}^{K} \int_{\mathcal{X}} \ell(\boldsymbol{h}(\mathbf{x}), k) \mathrm{d}P_{X_s} - (K-1) L_{\overline{s}}(\boldsymbol{h}). \qquad (3)$$

Hence, $L_s(\boldsymbol{h}) = \overline{L}_s(\boldsymbol{h})$. Combining Eq. (3) with the domain adaptation bound presented in [Ben-David *et al.*, 2010]

$$L_t(\boldsymbol{h}) \leq L_s(\boldsymbol{h}) + \frac{1}{2} d^{\ell}_{\mathcal{H}}(P_{X_s}, P_{X_t}) + \Lambda,$$

we prove this theorem. □

In this bound, the first term, i.e., Eq. (3), is the source classification error based on complementary-label source data. The second term is the distribution discrepancy distance between two domains and Λ is the difference in labeling functions across the two domains. The empirical form of Eq. (3) is known as complementary-label loss (see Eq. (5)).

2.3 How does BFUDA Bring Benefits to DA Field?

Collecting true-label data is always expensive in the real world. Thus, learning from less expensive data [Kumar *et al.*, 2017; Sakai *et al.*, 2017] has been extensively studied in machine learning field, including label-noise learning [Han *et al.*, 2018b; Han *et al.*, 2018a], complementary-label learning [Ishida *et al.*, 2019; Yu *et al.*, 2018; Ishida *et al.*, 2017] and so on. Among all these research directions, obtaining complementary labels 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 complementary labels 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 [Ishida *et al.*, 2019].

At present, the success of DA still highly relies on the scale of true-label source data, which is a critical bottleneck. With limited budget, it is unrealistic to obtain enough true-label source data and thus cannot achieve a good distribution adaptation result. For the same budget, 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, as they have sufficient true labels to support distributional adaptation. Thus if we can reduce the labeling cost in the source domain, for example, by using complementary-label data to replace true-label data (i.e. BFUDA), we can promote DA to more fields. Due to existing UDA methods require at least 20% true-label source data [Shu *et al.*, 2019], they cannot handle BFUDA problem. To address BFUDA problem directly, we propose a powerful solution, CLARINET, as follows.

3 CLARINET: One-step BFUDA Approach

The proposed CLARINET (Figure 2) mainly consists of feature extractor G, label predictor F and domain discriminator D. Furthermore, we add a mapping function T between the label predictor F and domain discriminator D to take full advantage of classifier predictions. In this section, we first introduce two losses used to train CLARINET, and then the whole training procedures of CLARINET is presented.

3.1 Loss Function in CLARINET

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

Complementary-label Loss. We first divided $\overline{d}_s$ into K disjoint subsets according to the complementary labels in $\overline{d}_s$,

$$\overline{d}_s = \cup_{k=1}^{K} \overline{d}_{s,k},\ \overline{d}_{s,k} = \{(\mathbf{x}_k^i, k)\}_{i=1}^{\overline{n}_{s,k}}, \qquad (4)$$

where $\overline{d}_{s,k} \cap \overline{d}_{s,k'} = \varnothing$ if $k \neq k'$ and $\overline{n}_{s,k} = |\overline{d}_{s,k}|$. Then, following Eq. (3), the complementary-label loss on $\overline{d}_{s,k}$ is

$$\begin{aligned}\overline{L}_s(G, F, \overline{d}_{s,k}) = &-(K-1)\frac{\overline{\pi}_k}{\overline{n}_{s,k}} \sum_{i=1}^{\overline{n}_{s,k}} \ell(F \circ G(\mathbf{x}_k^i), k) \\ &+ \sum_{j=1}^{K} \frac{\overline{\pi}_j}{\overline{n}_{s,j}} \sum_{l=1}^{\overline{n}_{s,j}} \ell(F \circ G(\mathbf{x}_j^l), k),\end{aligned} \qquad (5)$$

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

$$\overline{L}_s(G, F, \overline{d}_s) = \sum_{k=1}^{K} \overline{L}_s(G, F, \overline{d}_{s,k}). \tag{6}$$

As shown in Section 2.2, the complementary-label loss (i.e., Eq. (6)) 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 ℓ and model $F \circ G$ [Ishida *et al.*, 2019].

Remark 1. *Due to the negative part in $\overline{L}_s(G, F, \overline{d}_s)$, minimizing it will cause over-fitting [Kiryo* et al.*, 2017]. To overcome this problem, we use a correctional way [Ishida* et al.*, 2019] to minimize $\overline{L}_s(G, F, \overline{d}_s)$ (lines* 7-13 *in Algorithm 1).*

Scattered Conditional Adversarial Loss. According to [Song *et al.*, 2009], it is significant to capture multimodal structures of distributions using cross-covariance dependency between the features and classes (i.e., true labels). Since 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 [Long *et al.*, 2018]. The newly constructed features have shown great ability to discriminate source and target domains, since 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 $\boldsymbol{f_s}$ in Figure 2) is relatively close. Namely, it is hard to find significant predictive preference from the classifier predictions. According to [Song *et al.*, 2009], this kind of predictions cannot provide sufficient information to capture the multimodal structure of distributions. To fix it, we propose a mapping function T to scatter the classifier predictions $\boldsymbol{f} = [f_1, ..., f_K]^T$ ($\boldsymbol{f}$ could be $\boldsymbol{f_s}$ or $\boldsymbol{f_t}$ in Figure 2),

$$T(\boldsymbol{f}) = \left[\frac{f_1^{\frac{1}{l}}}{\sum_{j=1}^{K} f_j^{\frac{1}{l}}}, .., \frac{f_k^{\frac{1}{l}}}{\sum_{j=1}^{K} f_j^{\frac{1}{l}}}, ..., \frac{f_K^{\frac{1}{l}}}{\sum_{j=1}^{K} f_j^{\frac{1}{l}}} \right]^T. \tag{7}$$

It is a common approach of adjusting the "temperature" of categorical distribution. As $l \to 0$, the output of $T(\boldsymbol{f})$ will approach a Dirac ("one-hot") distribution.

Then to prioritize the discriminator on those easy-to-transfer examples, following [Long *et al.*, 2018], we measure the uncertainty of the prediction for sample $\mathbf{x}$ by

$$H(G, F, \mathbf{x}) = -\sum_{k=1}^{K} T(f_k(\mathbf{x})) \log T(f_k(\mathbf{x})). \tag{8}$$

Thus the scattered conditional adversarial loss is as follows,

$$\begin{aligned} L_{adv}(G, F, D, \overline{d}_s, d_t) = & \frac{\sum_{\mathbf{x} \in \overline{d}_s[X]} \omega_{\overline{s}}(\mathbf{x}) \log(D(\boldsymbol{g}(\mathbf{x})))}{\sum_{\mathbf{x} \in \overline{d}_s[X]} \omega_{\overline{s}}(\mathbf{x})} \\ & + \frac{\sum_{\mathbf{x} \in d_t} \omega_t(\mathbf{x}) \log(1 - D(\boldsymbol{g}(\mathbf{x})))}{\sum_{\mathbf{x} \in d_t} \omega_t(\mathbf{x})}, \end{aligned} \tag{9}$$

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

Algorithm 1 CLARINET: One-step BFUDA Approach

Input: $\overline{D}_s = \{(\mathbf{x}_s^i, \overline{y}_s^i)\}_{i=1}^{\overline{n}_s}$, $D_t = \{\mathbf{x}_t^i\}_{i=1}^{n_t}$.
Parameters: learning rate γ_1 and γ_2, epoch T_{max}, start epoch T_s, iteration N_{max}, class number K, tradeoff λ, network parameter $\theta_{F \circ G}$ and θ_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 $t = 1, 2 \ldots\ldots T_{max}$ do
 3:     Shuffle the training set $\overline{D}_s$, $D_t$;
 4:     for $N = 1, 2 \ldots\ldots N_{max}$ do
 5:         Fetch mini-batch $\overline{d}_s$, $d_t$ from $\overline{D}_s$, $D_t$;
 6:         Divide $\overline{d}_s$ into $\{\overline{d}_{s,k}\}_{k=1}^K$;
 7:         Calculate $\{\overline{L}_s(G, F, \overline{d}_{s,k})\}_{k=1}^K$ using Eq. (5), and
            $\overline{L}_s(G, F, \overline{d}_s)$ using Eq. (6);
 8:         if $\min_k\{\overline{L}_s(G, F, \overline{d}_{s,k})\}_{k=1}^K \geq 0$ then
 9:             Update $\theta_{F \circ G} = \theta_{F \circ G} - \gamma_1 \triangledown \overline{L}_s(G, F, \overline{d}_s)$;
10:         else
11:             Calculate $\overline{L}_{neg} = \sum_{k=1}^K \min\{0, \overline{L}_s(G, F, \overline{d}_{s,k})\}$;
12:             Update $\theta_{F \circ G} = \theta_{F \circ G} + \gamma_1 \triangledown \overline{L}_{neg}$;
13:         end if
14:         if $t > T_s$ then
15:             Calculate $L_{adv}(G, F, D, \overline{d}_s, d_t)$ using Eq. (9);
16:             Update $\theta_D = \theta_D - \gamma_2 \triangledown L_{adv}(G, F, D, \overline{d}_s, d_t)$;
17:             Update $\theta_{F \circ G} = \theta_{F \circ G} + \gamma_2 \lambda \triangledown L_{adv}(G, F, D, \overline{d}_s, d_t)$;
18:         end if
19:     end for
20: end for
```

3.2 Training Procedure of CLARINET

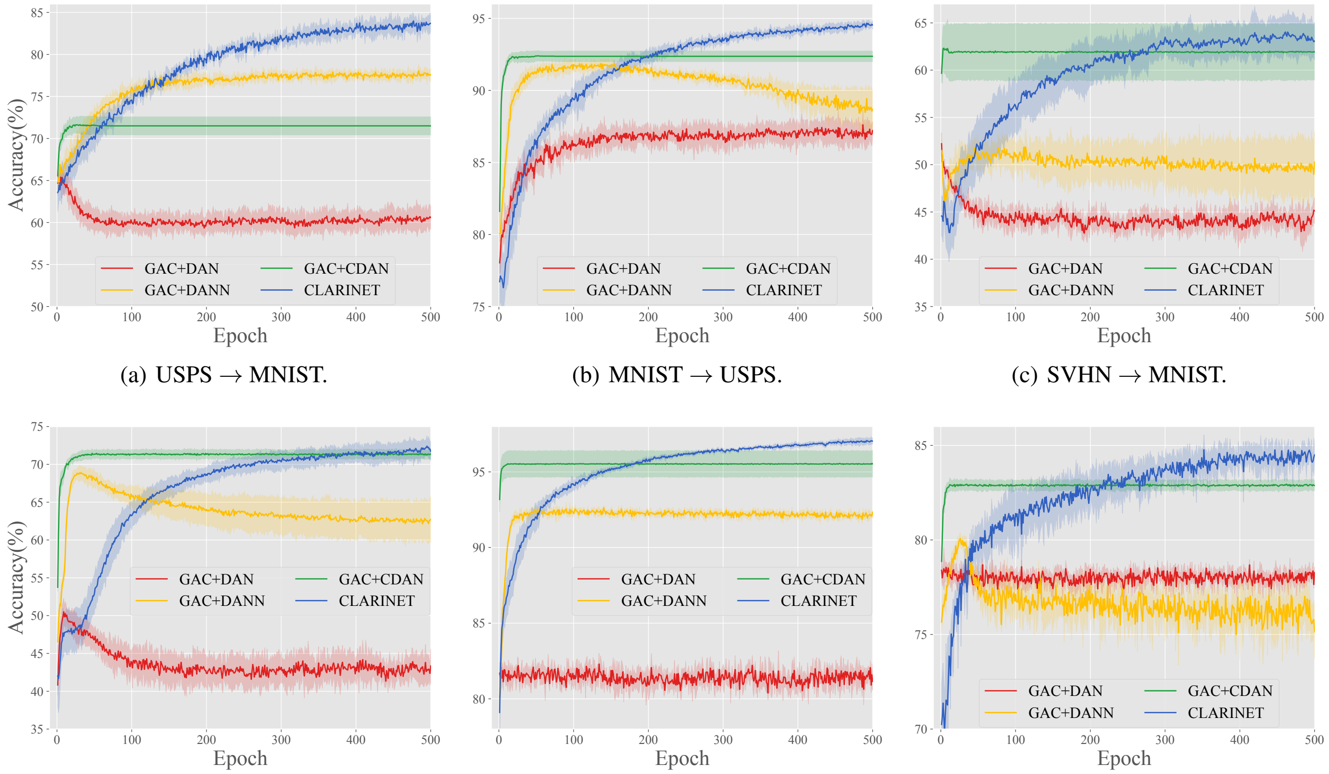
Based on the two losses proposed in Section 3.1, in CLARINET, we try to solve the following optimization problem,

$$\begin{aligned} & \min_{G,F} \overline{L}_s(G, F, \overline{D}_s) - \lambda L_{adv}(G, F, D, \overline{D}_s, D_t), \\ & \min_{D} L_{adv}(G, F, D, \overline{D}_s, D_t), \end{aligned} \tag{10}$$

where D tries to distinguish the samples from different domains by minimizing L_{adv}, while $F \circ G$ wants to maximize the L_{adv} to make domains indistinguishable. To solve the minimax optimization problem in Eq. (10), we add a gradient reversal layer [Ganin *et al.*, 2016] between the domain discriminator and the classifier, which multiplies the gradient by a negative constant (-λ) during the back-propagation. λ is a hyper-parameter between the two losses to tradeoff source risk and domain discrepancy.

The training procedures of CLARINET are shown in Algorithm 1. First, we initialize the whole network (line 1) and shuffle the training set (line 3). During each epoch, after minbatch $\overline{d}_s$ and d_t are fetched (line 5), we divide the source mini-batch $\overline{d}_s$ into $\{\overline{d}_{s,k}\}_{k=1}^K$ using Eq. (4) (line 6). Then, $\{\overline{d}_{s,k}\}_{k=1}^K$ are used to calculate the complementary-label loss for each class (i.e., $\{\overline{L}_s(G, F, \overline{d}_{s,k})\}_{k=1}^K$) and the whole complementary-label loss $\overline{L}_s(G, F, \overline{d}_s)$ (line 7).

If $\min_k\{\overline{L}_s(G, F, \overline{d}_{s,k})\}_{k=1}^K \geq 0$, we calculate the gradient $\triangledown \overline{L}_s(G, F, \overline{d}_s)$ and update parameters of G and F using



(a) USPS → MNIST.

(b) MNIST → USPS.

(c) SVHN → MNIST.

(d) MNIST → MNIST-M.

(e) SYND → MNIST.

(f) SYND → SVHN.

Figure 3: Test Accuracy vs. Epochs on 6 BFUDA Task. In (a)-(f), we compare target-domain accuracy of one-step approach, i.e., CLARINET (*ours*), with that of two-step approach (*ours*). It is clear to see that the accuracy of CLARINET gradually and steadily increases and eventually converges, achieving the best accuracy on each task.

gradient descent (lines 8-9). Otherwise, we sum negative elements in $\{\overline{L}_s(G, F, \overline{d}_{s,k})\}_{k=1}^{K}$ as $\overline{L}_{neg}$ (line 11) and calculate the gradient with $\nabla\overline{L}_{neg}$ (line 12). Then, we update parameters of G and F using gradient ascent (line 12), which is suggested by [Ishida *et al.*, 2019]. 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_{adv} (line 15). Then, L_{adv} is minimized over D (line 16), but maximized over $F \circ G$ (line 17) for adversarial training.

4 Experiments

Based on five commonly used datasets: *MNIST* (*M*), *USPS* (*U*), *SVHN* (*S*), *MNIST-M* (*m*) and *SYN-DIGITS* (*Y*), we verify efficacy of CLARINET on 6 BFUDA tasks: *M* → *U*, *U* → *M*, *S* → *M*, *M* → *m*, *Y*→ *M* and *Y*→ *S*. Note that, we generate complementary-label data according to [Ishida *et al.*, 2019].

4.1 Baselines

We compare CLARINET with the following baselines: *gradient ascent complementary label learning* (GAC) [Ishida *et al.*, 2019], namely non-transfer method, and several two-step methods, which sequentially combine GAC with UDA methods (including DAN [Long *et al.*, 2015], DANN [Ganin *et al.*, 2016] and CDAN [Long *et al.*, 2018]). In two-step approach, GAC method is first used to assign pseudo labels for complementary-label source data. Then, we train the classifier with pseudo-label source data and unlabeled target data using UDA methods. 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 paper, we use the entropy conditioning variant of CDAN.

4.2 Experimental Setup

We design our feature extractor G, label predictor F and domain discriminator D according to the architecture from previous works. More precisely, we pick the structures of feature extractor G from [Ganin *et al.*, 2016; Long *et al.*, 2018]. The label predictor F and domain discriminator D all share the same structure in all tasks, following CDAN [Long *et al.*, 2018]. We follow the standard protocols for unsupervised domain adaptation and compare the average classification accuracy based on 5 random experiments. For each experiment, we take the result of the last epoch. The batch size is set to 128 and the number of epochs is set to 500. SGD optimizer (momentum = 0.9, weight_decay = 5×10^{-5}) is with an initial learning rate of 0.005 in adversarial network and 5×10^{-5} in classifier. In mapping function T, l is set to 0.5. We update λ according to [Long *et al.*, 2018]. For parameters of each

Tasks	GAC	Two-step approaches (*ours*)			CLARINET (*ours*)
		GAC+DAN	GAC+DANN	GAC+CDAN_E	
$U \rightarrow M$	51.860	60.692±1.300	77.580±0.770	71.498±1.077	**83.692±0.928**
$M \rightarrow U$	77.796	87.215±0.603	88.688±1.280	92.366±0.365	**94.538±0.292**
$S \rightarrow M$	39.260	45.132±1.363	50.882±2.440	61.922±2.983	**63.070±1.990**
$M \rightarrow m$	45.045	43.346±2.224	62.273±2.261	71.379±0.620	**71.717±1.262**
$Y \rightarrow M$	77.070	81.150±0.591	92.328±0.138	95.532±0.873	**97.040±0.212**
$Y \rightarrow S$	72.480	78.270±0.311	75.147±1.401	82.878±0.278	**84.499±0.537**
Average	60.585	65.968	74.483	79.263	**82.426**

Table 1: Results on 6 BFUDA Tasks. Bold value represents the highest accuracy (%) on each row. Please note, the two-step methods and CLARINET are all first proposed in our paper.

Methods	$U \rightarrow M$	$M \rightarrow U$	$S \rightarrow M$	$M \rightarrow m$	$Y \rightarrow M$	$Y \rightarrow S$	Average
C w/ L_{CE}	0.445±0.722	0.055±0.129	3.708±0.688	7.088±0.424	1.832±0.102	1.298±0.070	2.404
C w/o T	83.192±1.796	93.419±0.588	52.438±1.927	**72.128±1.569**	95.442±1.004	83.055±0.652	79.946
CLARINET	**83.692±0.928**	**94.538±0.292**	**63.070±1.990**	71.717±1.262	**97.040±0.212**	**84.499±0.537**	**82.426**

Table 2: Ablation Study. Bold value represents the highest accuracy (%) on each column. We prove UDA methods cannot handle BFUDA tasks directly and the mapping function T can help improve the adaptation performance under BFUDA.

baseline, we all follow the original settings. We implement all methods with default parameters by PyTorch. The code of CLARINET is available at github.com/Yiyang98/BFUDA.

4.3 Results on BFUDA Tasks

Table 1 reports the target-domain accuracy of 5 methods on 6 BFUDA tasks. As shown, 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. Since CDAN has shown much better adaptation performance than DANN and DAN [Long *et al.*, 2018], GAC+CDAN should outperform other two-step methods on each task. However, on the task $U \rightarrow M$, 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 BFUDA approach.

Figure 3 shows the target-domain accuracy of two-step methods and CLARINET on 6 BFUDA tasks when increasing epochs. It is clear to see that the accuracy of CLARINET gradually and steadily increases and eventually converges, achieving the best accuracy on each task. The accuracy of GAC+CDAN always reaches plateau quickly. For GAC+DANN, its accuracy is unstable and significantly drops after certain epochs on $M \rightarrow U$, $M \rightarrow m$ tasks. While the accuracy of GAC+DAN is relatively stable but not satisfactory.

4.4 Ablation Study

Finally, we conduct experiments to show the contributions of components in CLARINET. We consider following baselines:

- C w/ L_{CE}: train CLARINET by Algorithm 1, while replacing $\overline{L}_s(G, F, \overline{D}_s)$ by cross-entropy loss.
- C w/o T: train CLARINET by Algorithm 1, without mapping function T.

C w/ L_{CE} uses the cross-entropy loss to replace the complementary-label loss. The target-domain accuracy of C w/ L_{CE} will show if UDA methods can address the BFUDA problem. Comparing CLARINET with C w/o T reveals whether the mapping function T takes effect. As shown in Table 2, the target-domain accuracy of C w/ L_{CE} is much lower than that of other methods. Namely UDA methods cannot handle BFUDA tasks directly. Although C w/o T achieves better accuracy than two-step methods, its accuracy is still worse than CLARINET's. Thus, the mapping function T can help improve the adaptation performance under BFUDA.

5 Conclusion

This paper presents a new problem setting for the domain adaptation field, called *budget-friendly unsupervised domain adaptation* (BFUDA), which exploits economical complementary-label source data instead of expensive true-label source data. Since existing UDA methods cannot address BFUDA problem, we propose a novel one-step BFUDA approach, called *complementary label adversarial network* (CLARINET). Experiments conducted on 6 BFUDA tasks confirm that CLARINET effectively achieves distributional adaptation from complementary-label source data to unlabeled target data and outperforms competitive baselines.

Acknowledgements

The work presented in this paper was supported by the Australian Research Council (ARC) under FL190100149 and DP170101632. We would like to thank the anonymous reviewers for their thoughtful comments. The first author particularly thanks the support by UTS-CAI during her visit.

References

[Agresti *et al.*, 2019] G. Agresti, H. Schäfer, P. Sartor, and P. Zanuttigh. Unsupervised domain adaptation for tof data denoising with adversarial learning. In *CVPR*, pages 5584–5593, 2019.

[Ao *et al.*, 2017] S. Ao, X. Li, and C.X. Ling. Fast generalized distillation for semi-supervised domain adaptation. In *AAAI*, pages 1719–1725, 2017.

[Ben-David *et al.*, 2010] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J.W. Vaughan. A theory of learning from different domains. *MLJ*, 79(1-2):151–175, 2010.

[Deng *et al.*, 2019] Z. Deng, Y. Luo, and J. Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *ICCV*, pages 9944–9953, 2019.

[Fang *et al.*, 2019] Zhen Fang, Jie Lu, Feng Liu, and Guangquan Zhang. Unsupervised domain adaptation with sphere retracting transformation. In *IJCNN*, pages 1–8. IEEE, 2019.

[Ganin *et al.*, 2016] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(59):1–35, 2016.

[Gong *et al.*, 2016] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf. Domain adaptation with conditional transferable components. In *ICML*, pages 2839–2848, 2016.

[Gong *et al.*, 2018] M. Gong, K. Zhang, B. Huang, C. Glymour, D. Tao, and K. Batmanghelich. Causal generative domain adaptation networks. *CoRR*, abs/1804.04333, 2018.

[Han *et al.*, 2018a] B. Han, J. Yao, G. Niu, M. Zhou, I. Tsang, Y. Zhang, and M. Sugiyama. Masking: A new perspective of noisy supervision. In *NeurIPS*, pages 5836–5846, 2018.

[Han *et al.*, 2018b] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8527–8537, 2018.

[Ishida *et al.*, 2017] T. Ishida, G. Niu, W. Hu, and M. Sugiyama. Learning from complementary labels. In *NeurIPS*, pages 5639–5649, 2017.

[Ishida *et al.*, 2019] T. Ishida, G. Niu, A. Menon, and M. Sugiyama. Complementary-label learning for arbitrary losses and models. In *ICML*, pages 2971–2980, 2019.

[Kiryo *et al.*, 2017] R. Kiryo, G. Niu, M.C. du Plessis, and M. Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *NeurIPS*, pages 1675–1685, 2017.

[Kumar *et al.*, 2017] A. Kumar, P. Sattigeri, and T. Fletcher. Semi-supervised learning with gans: Manifold invariance with improved inference. In *NeurIPS*, pages 5534–5544, 2017.

[Liu *et al.*, 2019] 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 *NeurIPS LTS Workshop*, 2019.

[Liu *et al.*, 2020] F. Liu, G. Zhang, and J. Lu. Heterogeneous domain adaptation: An unsupervised approach. *TNNLS*, Early Access, 2020.

[Long *et al.*, 2015] M. Long, Y. Cao, J. Wang, and M.I. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, page 97–105, 2015.

[Long *et al.*, 2018] M. Long, Z. Cao, J. Wang, and M.I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, pages 1640–1650, 2018.

[Saito *et al.*, 2017] K. Saito, Y. Ushiku, and T. Harada. Asymmetric tri-training for unsupervised domain adaptation. In *ICML*, pages 2988–2997, 2017.

[Sakai *et al.*, 2017] T. Sakai, M.C. du Plessis, G. Niu, and M. Sugiyama. Semi-supervised classification based on classification from positive and unlabeled data. In *ICML*, pages 2998–3006, 2017.

[Sankaranarayanan *et al.*, 2018] S. Sankaranarayanan, Y. Balaji, C.D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, pages 8503–8512, 2018.

[Shu *et al.*, 2019] Y. Shu, Z. Cao, M. Long, and J. Wang. Transferable curriculum for weakly-supervised domain adaptation. In *AAAI*, pages 4951–4958, 2019.

[Song *et al.*, 2009] L. Song, J. Huang, A. Smola, and K. Fukumizu. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *ICML*, pages 961–968, 2009.

[Sukhija *et al.*, 2016] S. Sukhija, N.C. Krishnan, and G. Singh. Supervised heterogeneous domain adaptation via random forests. In *IJCAI*, pages 2039–2045, 2016.

[Yan *et al.*, 2017] Y. Yan, W. Li, M. K. P. Ng, M. Tan, H. Wu, H. Min, and Q. Wu. Learning discriminative correlation subspace for heterogeneous domain adaptation. In *IJCAI*, pages 3252–3258, 2017.

[Yu *et al.*, 2018] X. Yu, T. Liu, M. Gong, and D. Tao. Learning with biased complementary labels. In *ECCV*, pages 69–85, 2018.

[Zhang *et al.*, 2015] K. Zhang, M. Gong, and B. Schölkopf. Multi-source domain adaptation: A causal view. In *AAAI*, pages 3150–3157, 2015.

[Zhao *et al.*, 2019] S. Zhao, H. Fu, M. Gong, and D. Tao. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *CVPR*, pages 9788–9798, 2019.

[Zhou *et al.*, 2019a] J. T. Zhou, S. J. Pan, and I. W. Tsang. A deep learning framework for hybrid heterogeneous transfer learning. *AIJ*, 275:310 – 328, 2019.

[Zhou *et al.*, 2019b] J. T. Zhou, H. Zhang, D. Jin, and X. Peng. Dual adversarial transfer for sequence labeling. *TPAMI*, Early Access, 2019.