Source-free Unsupervised Domain Adaptation with Surrogate Data Generation

Hao Yan[1]
haoyan6@cmail.carleton.ca
Yuhong Guo[1,2]
yuhong.guo@carleton.ca
Chunsheng Yang[3]
chunsheng.yang@nrc-cnrc.gc.ca

[1] Carleton University
Ottawa, Canada
[2] Canada CIFAR AI Chair, Amii
Edmonton, Canada
[3] National Research Council Canada
Ottawa, Canada

Abstract

Source-free unsupervised domain adaptation aims to learn a model that generalizes well on a target domain given the pre-trained source model and unlabeled target data. Traditional unsupervised domain adaptation methods are mostly not applicable to this setting since no source data are available. To tackle this problem, we propose to generate labeled surrogate source training data from the source model by fixing the model and optimizing the inputs. To avoid naive local fittings to individual instances and in light of the model optimization process, we further enforce model gradient based global fitting constraints on the whole dataset generation and solve the formulated optimization problem using an ADMM algorithm. The generated labeled source training data can then be used to deploy existing unsupervised domain adaptation methods. Furthermore, we propose to incorporate the unlabeled target data into the domain adaptation process to improve generalization in the target domain with a mutual information loss. Experiments show that our proposed method can achieve the state-of-the-art results on benchmark datasets.

1 Introduction

Supervised deep learning has achieved great success with the help of large amounts of labeled data, which induce expensive data annotation cost. Unsupervised domain adaptation (UDA) reduces such cost by exploiting labeled data from an auxiliary source domain to help train a prediction model in an unlabeled target domain. Due to the distribution discrepancy between source and target domains, the model trained on the source domain cannot generalize well on the target data. Thus most unsupervised domain adaptation methods seek to reduce the domain discrepancy given the theoretical guarantee in [1]. One prevailing paradigm is to learn domain-invariant representations by minimizing the cross-domain feature discrepancy with certain metric such as maximum mean discrepancy (MMD) [11] or through adversarial learning schemes [7, 21]. Some other works [5, 26, 30] improve domain adaptation by utilizing semi-supervised learning or self-training with pseudo-labels.

However, traditional unsupervised domain adaptation setting assumes the availability of source domain data when training models for the target domain. This may not be guaranteed in many real-world scenarios. For example, for privacy protection, users' personal

© 2021. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

data cannot be stored after models being trained according to the laws of many countries; patients' medical data are not allowed to be made public without permission. Companies may release their models but data won't be accessible for commercial purposes. All these situations induce a more challenging domain adaptation setting, source-free unsupervised domain adaptation (SFUDA). In this setting, only source model and unlabeled target data are available and the goal is still to learn a prediction model that generalizes well in the target domain. Traditional UDA methods may not be applicable in this setting as they require both source and target domain data. To tackle this challenge, some researchers have made efforts on developing new source-free domain adaptation methods. For example, Li et al. [18] utilize conditional GANs to generate labeled target data and fine-tune the source model with multiple semi-supervised model regularization terms. Kurmi et al. [15] use conditional GANs to simultaneously perform data generation and domain adaptation. Kim et al. [12] exploit pseudo-labels in the target domain to improve the model generalization performance.

In this paper, we propose an optimization-based training data generation method to simulate the source domain data and facilitate the reuse of existing UDA methods in the new source-free unsupervised domain adaptation setting. Different from the existing works, we have no need to introduce extra generative models such as conditional GANs. Instead, we simply generate a surrogate source training set based on the given source prediction model from the optimization perspective. First, we fix the model parameters and minimize the standard cross-entropy loss by updating the input training data. In order to avoid local fittings to individual instances, we further deploy model gradient based constraints to enforce global fitting to the whole simulated training set. We solve the resulted optimization problem by developing an Alternating Direction Method of Multipliers (ADMM) [3]. With the generated source training data, existing UDA methods can be reused to handle domain adaptation in the new problem setting. To increase the generalization capacity of the target model, we further exploit the unlabeled target domain data by incorporating a mutual information loss into the adaptation process. To show the effectiveness of the proposed source data generation method, we conduct experiments using several UDA methods with the generated source data. Our method yields similar domain adaptation results to the ones produced with the original source domain data. When unlabeled target domain data is incorporated, our method produces the state-of-the-art performance in the source-free unsupervised domain adaptation setting on standard benchmarks.

2 Related works

Unsupervised Domain Adaptation. Most unsupervised domain adaptation methods seek to reduce the cross-domain discrepancy based on the theoretical guarantee in [1]. Related works can be divided into two categories, metric-based methods and adversarial training methods. Metric-based methods enforce the model to learn domain-invariant representations by minimizing feature discrepancy between domains with certain distance metrics. Examples of these metrics include the maximum mean discrepancy (MMD) [20], the moment matching [9], and the Wasserstein distance [16]. Inspired by the Generative Adversarial Networks (GAN) [8], adversarial training has been utilized to align cross-domain distributions in different levels, including the feature-level [7, 21], input-level [25], and output-level [28]. Regularization terms from semi-supervised learning approaches have also been utilized to adapt the source model using unlabeled target data. The Mean teacher method [27] has been used in [5] to regularize the model predictions to be consistent across the student

and teacher models. Entropy minimization [9] for unlabeled target data enforces the model's decision boundaries to be far away from data-dense regions [26]. Virtual adversarial training [22] acts as a locally-Lipschitz constraint in [26] to guarantee the empirical approximation of conditional entropy when used together with the entropy minimization. Pseudo-labeling [17] has also inspired the self-training methods for unsupervised domain adaptation. The method in [30] alternately selects high-confident pseudo-labels with certain criteria and re-trains the model with the pseudo-labeled target data.

Source-Free Unsupervised Domain Adaptation. In source-free unsupervised domain adaptation setting, labeled source data are unavailable, which makes the problem more challenging. Although UDA has been popularly studied, the source-free UDA only starts attracting attentions. The effective performances in this setting are typically obtained by deploying semi-supervised techniques. Li et al. [18] propose to use the conditional GAN to generate labeled target data through the input-level adversarial training and semantic consistency constraint, and fine-tune the target model with several semi-supervised learning terms. PPDA in [12] assigns pseudo-labels to target samples based on a prototype classifier and introduces a sample-level re-weighting scheme to get more confident pseudo-labels. SDDA [15] uses conditional GANs to simultaneously generate labeled source data and perform domain adaptation to fine-tune the target model. Liang et al. [19] deal with a slightly different source-free unsupervised domain adaptation problem which allows additional efforts in training the original source models and changing the model architectures. Other different source-free settings considered in the literature include the universal source-free domain adaptation [13] and open-set source-free domain adaptation [14]. Different from these previous studies, our method simply simulates the source training data based on the principle of source model optimization without introducing extra generative models.

3 Proposed Method

We consider the following source-free unsupervised domain adaptation setting. We have access to the unlabeled data X_t in the target domain and a prediction model trained in the labeled source domain, $F_s(\Phi_s(\cdot))$, which consists of a feature extractor $\Phi_s(\cdot)$ and a classifier $F_s(\cdot)$. However, the original labeled source data (X_s, Y_s) used to produce the source model or any other data in the source domain are not accessible. We also assume the two domains share the same prediction label space and aim to produce a prediction model that performs well in the target domain.

In this section, we present the proposed two stages source-free unsupervised domain adaptation method as illustrated in Figure 1. First, we fix the source model and generate surrogate source data based on the model optimization principle by minimizing the cross-entropy training loss, while enforcing the model gradient based constraints to improve global fitting. Then with the generated surrogate source data, we deploy existing UDA methods and incorporate the unlabeled target data to learn a target predict model that can generalize well.

3.1 Surrogate Source Training Data Generation

We motivate our source training data generation idea from the supervised source model training problem. Given the labeled source domain data (X_s, Y_s), the source prediction model can

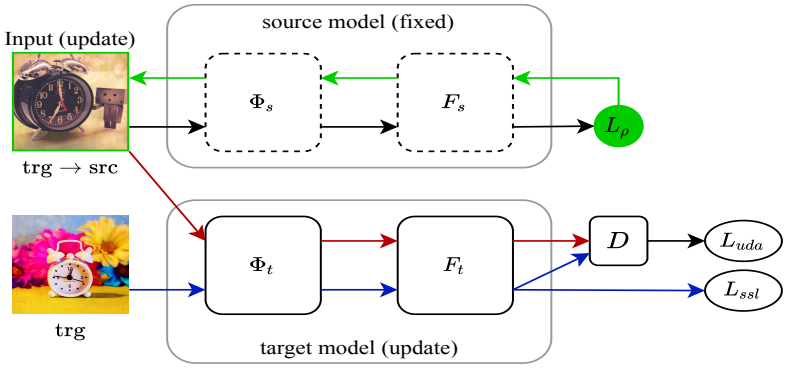

Figure 1: Illustration for the proposed method. Top: fix source model and optimize target-image initialized source training images. Bottom: the generated source training images are used to perform domain adaptation.

typically be trained by minimizing the following L2-norm regularized cross-entropy loss:

$$\min_{\Theta} \mathcal{L}(\Theta) = \mathcal{L}_{CE}(F_s \circ \Phi_s(X_s;\Theta), Y_s) + \frac{\gamma}{2}\|\Theta\|_F^2 \tag{1}$$

where $F_s \circ \Phi_s$ denotes the composition function of F_s and Φ_s, and $\Theta = \{\Theta_\phi, \Theta_f\}$ denote the corresponding model parameters; γ is a hyperparameter, $\|\cdot\|_F$ denotes the Frobenius norm, and the regularizer is typically incorporated to avoid overfitting.

Now given the prediction model with fixed parameters Θ, we can reverse the training process to treat the input data as variables and generate a source instance $\hat{x}_s$ from a given label $\hat{y}_s$ by minimizing the objective above in a similar way as follows:

$$\hat{x}_s = \arg\min_{x_s} \mathcal{L}_{CE}(F_s \circ \Phi_s(x_s;\Theta), \hat{y}_s) \tag{2}$$

Our purpose here is not to generate a source domain instance/image with good qualities, but rather to generate a source training dataset that can reproduce the given prediction model $F_s \circ \Phi_s(\cdot;\Theta)$. For this purpose, the set of source instances generated need to be sufficiently diverse to represent the original source distribution in the ideal case. As we do not have any information about the source distribution, we have a naturally diverse target dataset X_t. Hence we propose to generate a diverse surrogate source training set $(\hat{X}_s, \hat{Y}_s)$ by starting from X_t. Specifically, we initialize $\hat{X}_s$ as X_t and use the predicted labels on $\hat{X}_s$ under the current source prediction model as the corresponding labels $\hat{Y}_s$. Then we can generate each surrogate training instance in $\hat{X}_s$ through Eq.(2) by using a simple gradient descent algorithm.

Nevertheless, the data generation above only considers the local fitting of each instance to the source model by decoupling the data set generation into separate individual instance generations. Although the initialization procedure can induce diverse data generation, it can still deviate from the goal of generating surrogate training set to reproduce the source prediction model. To this end, we further propose to enforce the global fitting of the generated data set $(\hat{X}_s, \hat{Y}_s)$ based on the optimality condition of the prediction model optimization. Specifically, one necessary condition the optimal model parameters in Eq.(1) need to satisfy is that the gradient becomes zero; that is,

$$\nabla_\Theta \mathcal{L}(\Theta) = \nabla_\Theta \mathcal{L}_{CE} + \gamma\Theta = 0 \tag{3}$$

Note for given Θ, the gradient $\nabla_\Theta \mathcal{L}(\Theta)$ is a global function of all the input training instances, and it is not decomposable over individual training instances. Therefore, to ensure the generated surrogate training data satisfy the optimality condition for the model parameters, we can incorporate the condition in Eq.(3) as global equality constraints for our source training data generation process and solve the following global data fitting problem:

$$\hat{X}_s = \arg\min_{X_s} \mathcal{L}_{CE}(F_s \circ \Phi_s(X_s;\Theta), \hat{Y}_s) \quad \text{s.t. } \nabla_\Theta \mathcal{L}_{CE} + \gamma\Theta = 0 \tag{4}$$

Although the principle of the global surrogate source data generation above is clear, the gradients of $\mathcal{L}_{CE}$ *w.r.t.* the feature extraction parameters Θ_ϕ are hard to compute layerwise over the deep neural networks as an explicit function of the training data. We hence relax the constraints to consider only the gradients of $\mathcal{L}_{CE}$ *w.r.t.* the classifier parameters Θ_f, *i.e.*

$$\hat{X}_s = \arg\min_{X_s} \mathcal{L}_{CE}(F_s \circ \Phi_s(X_s;\Theta), \hat{Y}_s) \quad \text{s.t. } \nabla_{\Theta_f} \mathcal{L}_{CE} + \gamma\Theta_f = 0 \tag{5}$$

The commonly used deep learning models (*e.g.* ResNet) usually utilize a single fully-connected layer with softmax activations as the classifier F. In this case, the gradients of $\mathcal{L}_{CE}$ *w.r.t.* Θ_f can be computed as

$$\nabla_{\Theta_f} \mathcal{L}_{CE} = \frac{1}{N}(\text{softmax}(F_s \circ \Phi_s(X_s;\Theta)) - \hat{Y}_s)^\top \cdot \Phi_s(X_s;\Theta_\phi), \tag{6}$$

where $\text{softmax}(\cdot)$ is the softmax function applied on each row of its input matrix $F_s \circ \Phi_s(X_s;\Theta)$ that denotes the model output on each instance, each row of the label matrix $\hat{Y}_s$ is a one-hot label indicator vector, and $\Phi_s(X_s;\Theta_\phi)$ denotes the feature matrix of all the N instances.

We propose to solve the equality constrained optimization problem in Eq.(5) by using an alternating direction method of multipliers (ADMM) [3]. We first derive the original problem into the following augmented Lagrangian formulation: $\max_\Lambda \min_{X_s} L_\rho(X_s, \Lambda)$, where

$$L_\rho(X_s,\Lambda) = \mathcal{L}_{CE}(F_s \circ \Phi_s(X_s;\Theta), Y_s) + \text{tr}(\Lambda^\top(\nabla_{\Theta_f}\mathcal{L}_{CE} + \gamma\Theta_f)) + \frac{\rho}{2}\|\nabla_{\Theta_f}\mathcal{L}_{CE} + \gamma\Theta_f\|_F^2. \tag{7}$$

Here Λ is the Lagrangian dual variable matrix associated with the equality constraints and ρ is a penalty hyperparameter. This can then be solved using the ADMM algorithm presented in Algorithm 1. In each iteration of the ADMM algorithm, we alternately update the primal variables X_s and the dual variables Λ. As the number of instances in X_s, N, is typically large, it is unrealistic to optimize all the instances X_s simultaneously considering the limitation of the GPU memory. Inspired by the mini-batch SGD, we propose to update X_s in a batch-wise coordinate descent procedure. For each batch X_B, we fix the other variables and take K gradient descent steps to update X_B.

3.2 UDA with Generated Surrogate Source Data

The source data generation method above enforces that the generated training data can be utilized to produce the original source model parameters and hence can be used as a surrogate source training dataset to deploy standard unsupervised domain adaptation methods.

As reviewed in Section 2, the unsupervised domain adaptation methods typically train the target prediction model on the labeled source data while using a regularization term to reduce the cross-domain discrepancy based on the given source data and target data. For example,

Algorithm 1: ADMM algorithm

Input : source model $F_s \circ \Phi_s(\cdot;\Theta)$, unlabeled target data X_t, the maximum iteration # M, the maximum # of gradient steps K, stepsize η, penalty parameter ρ

Initialize $\Lambda = 0, X_s = X_t$;
$\hat{Y}_s \leftarrow F_s \circ \Phi_s(X_t;\Theta)$;
for $i = 1 : M$ **do**
  for *batch* $(X_B, \hat{Y}_B) \subset (X_s, \hat{Y}_s)$ **do** `// batch update on primal`
    for $k = 1 : K$ **do**
      $X_B = X_B - \eta \frac{\partial}{\partial X_B} L_\rho(X_s, \Lambda)$;
    end
  end
  $\Lambda = \Lambda + \rho(\nabla_{\Theta_f} \mathcal{L}_{CE} + \gamma \Theta_f)$; `// update dual variables`
end

Output: generated source data $(X_s, \hat{Y}_s)$

the adversarial learning based domain adaptation method CDAN [21] deploys a conditional adversarial loss to minimize the domain discrepancy in the extracted feature space:

$$\min_{\Phi_t} \max_{D} \mathcal{L}_{adv}(X_s, X_t; \Phi_t, D) = \mathbb{E}_{x_s \in X_s} \log[D(\Phi_t(x_s) \otimes g_s)] + \mathbb{E}_{x_t \in X_t} \log[1 - D(\Phi_t(x_t) \otimes g_t)], \tag{8}$$

where the domain discriminator D is introduced to align cross-domain conditional feature distributions by playing the min-max game, $\otimes$ denotes the Kronecker product, g_s and g_t are the predicted probability vectors on instance x_s and x_t, *i.e.* $g = \text{softmax}(F_t \circ \Phi_t(x))$. With the given source model and the generated surrogate source training data $(\hat{X}_s, \hat{Y}_s)$, we can initialize the target prediction model with the source model such as $F_t = F_s$ and $\Phi_t = \Phi_s$, and then fine-tune the target model, in particular the target feature extractor Φ_t, by playing the min-max game with the adversarial loss: $\min_{\Phi_t} \max_D \mathcal{L}_{adv}(\hat{X}_s, X_t; \Phi_t, D)$. With an effective surrogate source data generation, we would expect the target model obtained in this manner will have a good adaptation performance in the target domain; in the best case, its performance can be very close to the standard CDAN method applied with the original labeled source data.

Moreover, to better exploit the unlabeled target data, we further extend the deployment of UDA methods into a Semi-Supervised Fine-Tuning framework with the generated Surrogate Source Data (SSFT-SSD) by incorporating the unlabeled target domain data X_t into the prediction model fine-tuning process in a semi-supervised manner that involves updating both F_t and Φ_t. Specifically, we propose to deploy a mutual information loss on the unlabeled target data. The mutual information criterion has been shown to be efficient for semi-supervised learning [2]. This criterion captures the mutual information between the inputs and outputs of a given prediction model and its empirical computation form can be defined as follows:

$$\mathcal{L}_{MI}(X_t; \Phi_t, F_t) = H(\mathbb{E}_{x_t \in X_t}[p(\mathcal{Y}|x_t)]) - \mathbb{E}_{x_t \in X_t}[H(p(\mathcal{Y}|x_t))] \tag{9}$$

where $p(\mathcal{Y}|x_t) = \text{softmax}(F_t \circ \Phi_t(x_t)))$ is the predicted probability vector on x_t, and $H(\cdot)$ denotes the entropy function. The first term in this mutual information criterion is the entropy of the averaged prediction vector over all samples which encourages the model predictions to be balanced across all samples. The second term is the negative averaged entropy which enforces the model predictions to be confident. With the CDAN adaptation strategy, our final

fine-tuning model can be formulated as:

$$\min_{\Phi_t, F_t} \max_{D} -\mathcal{L}_{MI}(X_t; \Phi_t, F_t) + \lambda \mathcal{L}_{adv}(\hat{X}_s, X_t; \Phi_t, D). \tag{10}$$

which can be solved using the standard min-max gradient descent algorithm through a gradient-reverse layer [6].

4 Experiments

We conducted two sets of experiments on the standard benchmarks. In the first set of experiments we tested the proposed surrogate source data generation method by deploying two existing UDA methods. In the second set of experiments, we compared our SSFT-SSD framework[1] with the state-of-the-art source-free unsupervised domain adaptation methods.

4.1 Experimental Setting

Datasets. We conducted experiments on the following benchmark datasets: *Office-31* [24], *Office-Home* [29], and *VisDA-2017* [23]. *Office-31* is a standard small-sized visual domain adaptation benchmark which contains images of 31 categories from three domains: Amazon (**A**), DSLR (**D**) and Webcam (**W**). *Office-Home* is a medium-sized dataset with images belonging to 65 categories from four distinct domains: Artistic images (**Ar**), Clip Art (**Cl**), Product images (**Pr**), and Real-World images (**Rw**). *VisDA-2017* is a large-scale synthetic-to-real dataset with images in 12 categories from two domains: *Synthetic* and *Real*.

Implementation Details. We used the same network architecture as the previous methods for fairness: on Office-31 and Office-Home, ResNet-50 [10] is used as the backbone network, while ResNet-101 [10] is used on the VisDA-2017 dataset. Following [6], the fully-connected (FC) layer in the ResNet network is replaced with a bottleneck and one FC layer, where the bottleneck layer is composed of one FC layer with 256 units and an one-dimensional Batch Normalization (BN) layer. The batch size is set as 64. For the source data generation, the maximum step number K is set to 10, the stepsize η is set to 10, γ and ρ are set to 0.01, and the maximum iteration number M is set to 3. For the UDA training with generated source data, we adopt the mini-batch SGD algorithm with momentum 0.9. Following [6], the learning rate is adjusted per batch iteration according to $\eta_i = \eta_0(1+\alpha\, i)^{-\beta}$, where $\alpha = 0.001, \beta = 0.75$ and i is the iteration index. The initial learning rate η_0 is set as 0.0001 for the pre-trained backbone module and 0.001 for the bottleneck and FC layers.

4.2 Results of UDA with Generated Surrogate Source Data

To investigate the effectiveness of the surrogate source data generation method, we conducted experiments to compare the performance of the classic UDA methods using the generated source data with their performance obtained when using the original source data. In principle, the proposed source data generation method can be used by any UDA methods to solve the SFUDA problem. In this experiment, we deployed two standard UDA methods, DANN [7] and CDAN [21]. For each UDA method, we compared their results with the original source data (denoted as "+original"), the surrogate data generated with local fitting

[1] Code is available at: https://github.com/cnyanhao/SSFTSSD.

Table 1: Accuracy (%) of UDA methods on Office-31 dataset (ResNet-50)

Methods	A → D	A → W	D → A	D → W	W → A	W → D	Avg.
Source-only	80.5	74.7	63.0	95.7	62.3	97.8	79.0
DANN+original	83.6	91.4	73.3	97.9	70.4	100.0	86.1
DANN+local	85.1	83.0	70.5	97.5	69.4	99.4	84.2
DANN+global	85.9	82.9	71.6	97.7	69.9	99.6	84.6
CDAN+original	89.9	93.8	73.4	98.5	70.4	100.0	87.7
CDAN+local	85.1	84.4	72.2	97.7	70.6	99.6	84.9
CDAN+global	85.7	84.3	72.5	98.1	71.8	99.8	85.4

Table 2: Accuracy (%) of UDA methods on Office-Home dataset (ResNet-50)

Methods	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg.
Source-only	43.8	66.8	74.9	50.9	61.9	63.6	51.7	37.1	72.7	64.4	43.0	76.6	59.0
DANN+original	53.8	62.6	74.0	55.8	67.3	67.3	55.8	55.1	77.9	71.1	60.7	81.1	65.2
DANN+local	47.7	72.1	77.1	55.6	67.0	68.2	56.8	41.0	75.9	68.1	46.3	79.2	62.9
DANN+global	48.4	72.4	77.2	55.6	68.1	68.2	56.9	41.9	76.3	67.7	46.3	79.1	63.2
CDAN+original	55.2	72.4	77.6	62.0	69.7	70.9	62.4	54.3	80.5	75.5	61.0	83.8	68.8
CDAN+local	46.9	73.1	78.5	57.8	70.1	70.3	58.3	40.9	77.4	68.8	45.3	80.2	64.0
CDAN+global	46.8	73.2	78.6	57.7	70.7	70.2	58.6	40.3	77.4	69.5	45.8	79.9	64.1

(denoted as "+local") and global fitting (denoted as "+global") respectively. We also compared them with the baseline result yielded by directly applying the source model, denoted as "Source-only". The comparison results on the three datasets, Office-31, Office-Home and VisDA-2017, are reported in Table 1, 2 and 3 respectively.

We can see that with our proposed surrogate source data generation, both global fitting and local fitting outperform Source-only with large performance gains, which validates that the generated source data are effective in aligning the cross-domain distributions and improving the adaptation performance in the target domain. Moreover, both "+local" and "+global" achieve comparable performance with "+original". Note the performance of DANN+original and CDAN+original can be treated as the upper bound performance of these methods under the source-free setting. These comparison results validated that our proposed surrogate source data generation methods can generate useful data which are able to approximate the original source training data from the perspective of domain adaptation and be combined with standard UDA methods to tackle the SFUDA problem with inaccessible source data. Between the local and global fitting methods, the surrogate source data generated using global fitting can achieve better performance than those generated using local fitting. This verifies that the optimality condition constraints incorporated in global fitting do help generate source data that are more consistent with the source model. Despite the difference, the results do not show much of the potential drawback of the local fitting procedure. The possible reason is that the source and target domains in these datasets share similar label distributions, and using the target domain data as the initial values for surrogate source data generation can naturally help promote the data diversity and distribution fitting in the source domain, which originally can be drawbacks of the local fitting method.

4.3 Results of the SFUDA Methods

Previous results have shown that the generated source data are useful and can be combined with traditional UDA methods to help align cross-domain distributions. In the second set of experiments, we further investigate our proposed Semi-Supervised Fine-Tuning frame-

Table 3: Accuracy (%) of UDA methods on VisDA-2017 dataset (ResNet-101)

Methods	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
Source-only	83.1	23.8	57.4	74.5	67.3	1.4	88.9	9.3	74.3	40.5	76.2	3.2	50.0
DANN+original	93.5	74.3	83.4	50.7	87.2	90.2	89.9	76.1	88.1	91.4	89.7	39.8	79.5
DANN+local	89.8	69.5	78.9	70.3	92.1	49.0	91.4	77.0	77.0	52.7	78.1	37.1	71.9
DANN+global	91.6	70.9	77.4	70.7	92.9	53.9	91.1	74.5	77.4	61.1	75.6	36.4	72.8
CDAN+original	94.0	69.2	78.9	57.0	89.8	94.9	91.9	80.3	86.8	84.9	85.0	48.5	80.1
CDAN+local	93.1	61.4	68.3	74.3	85.8	57.5	94.0	71.1	88.5	63.2	84.7	29.2	72.6
CDAN+global	93.0	61.0	80.2	73.3	92.5	34.7	93.0	76.7	90.0	82.5	79.9	33.2	74.1

Table 4: Accuracy (%) of SFUDA methods on Office-31 dataset (ResNet-50)

Methods	source training	A → D	A → W	D → A	D → W	W → A	W → D	Avg.
Source-only	✗	80.5	74.7	63.0	95.7	62.3	97.8	79.0
SDDA	✗	85.3	82.5	66.4	**99.0**	67.7	99.8	83.5
3C-GAN	✗	92.7	93.7	**75.3**	98.5	**77.8**	99.8	**89.6**
SSFT-SSD(Ours)	✗	**95.2**	**95.0**	72.7	98.7	73.5	**100.0**	89.2
SHOT-IM	✓	88.8	90.8	73.6	98.4	71.7	99.9	87.2
SHOT	✓	93.1	90.9	74.5	98.8	74.8	99.9	88.7

work with the generated Surrogate Source Data (SSFT-SSD) under the source-free UDA (SFUDA) setting, by comparing its results with the reported results of some state-of-the-art SFUDA methods, including SDDA [15], PPDA [17], 3C-GAN [18], SHOT-IM and SHOT [19]. The SHOT models [19] take additional advantage of deploying extra efforts in training the source models to facilitate source-free unsupervised domain adaptation. Here we simply use their results as references. The proposed SSFT-SSD framework can be combined with any UDA methods. Following our methodology section, here we adopted the CDAN [21]. The comparison results on the three datasets are reported in Table 4, 5, and 6 respectively.

Table 4 presents the results on the six domain adaptation tasks of Office-31. By combining with the results reported in Table 1, we can see that both 3C-GAN and SHOT outperform the standard UDA methods, DANN+original and CDAN+original. The reason lies in that these SFUDA methods have largely adopted semi-supervised learning terms to exploit the unlabeled data in the target domain. This is the motivation that we develop our SSFT-SSD to integrate the strengths of UDA and semi-supervised learning through surrogate data generation. We can see that our SSFT-SSD method further improves our previous best results obtained with CDAN+global in Table 1. SSFT-SSD outperforms the Source-only on all tasks and increases the average accuracy by 10.2 percentage points. Comparing to the state-of-the-art SFUDA methods, among the six tasks, SSFT-SSD produces the best results on three tasks, while 3C-GAN produces the best results on two tasks and SDDA on one task. In terms of average performance over all tasks, SSFT-SSD performs very similar to 3C-GAN while being much simpler without 3C-GAN's multiple terms, and outperforms SDDA with a notable gain of 5.7 percentage points. Although the SHOT methods deploy additional efforts in source model training, our proposed model still outperforms them.

Table 5 reports the results on the 12 domain adaptation tasks of the Office-Home dataset. Again, our SSFT-SSD further improves the previous CDAN+global and outperforms the upper bound CDAN+original produced in Table 2. This demonstrates the efficacy of the semi-supervised term we adopted. Compared with Source-only, our proposed SSFT-SSD improves the classification accuracy on all tasks and increases the average accuracy by 10.8 percentage points. We also compared with the results of the state-of-the-art method PPDA

Table 5: Accuracy (%) of SFUDA methods on Office-Home dataset (ResNet-50)

Methods	source training	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg.
Source-only	✗	43.8	66.8	74.9	50.9	61.9	63.6	51.7	37.1	72.7	64.4	43.0	76.6	59.0
PPDA	✗	48.5	71.3	75.6	63.9	69.0	72.1	62.4	43.5	76.0	70.4	50.1	76.1	64.9
SSFT-SSD(Ours)	✗	**51.7**	**76.0**	**79.9**	**66.8**	**75.8**	**77.2**	**63.9**	**52.1**	**80.6**	**73.5**	**57.1**	**83.0**	**69.8**
SHOT-IM	✓	52.8	72.9	78.4	65.4	73.8	74.1	64.6	50.8	78.9	72.7	53.5	81.2	68.3
SHOT	✓	56.9	78.1	81.0	67.9	78.4	78.1	67.0	54.6	81.8	73.4	58.1	84.5	71.6

Table 6: Accuracy (%) of UDA methods on VisDA-2017 dataset (ResNet-101)

Methods	source training	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
Source-only	✗	83.1	23.8	57.4	74.5	67.3	1.4	**88.9**	9.3	74.3	40.5	76.2	3.2	50.0
PPDA	✗	81.5	79.4	**80.3**	61.8	92.3	91.9	84.5	82.7	86.5	58.4	74.2	43.5	76.4
3C-GAN	✗	94.8	73.4	68.8	**74.8**	93.1	**95.4**	88.6	**84.7**	89.1	84.7	83.5	48.1	81.6
SSFT-SSD(Ours)	✗	**95.4**	**86.5**	79.3	51.5	**92.9**	94.5	82.1	79.7	**90.0**	**87.1**	**87.8**	**57.9**	**82.1**
SHOT-IM	✓	89.9	80.1	79.1	50.9	88.0	90.5	78.2	78.5	89.3	80.2	85.8	44.9	77.9
SHOT	✓	92.6	81.1	80.1	58.5	89.7	86.1	81.5	77.8	89.5	84.9	84.3	49.3	79.6

[17]: SSFT-SSD outperforms PPDA consistently across all the 12 tasks with notable performance gains. For the favorably trained SHOT models [19], SSFT-SSD can still outperform SHOT-IM in terms of average performance, while being slightly inferior to SHOT.

Table 6 presents the results on VisDA-2017. We can see that our SSFT-SSD method further improves our previous best results obtained with CDAN+global in Table 3 on this dataset as well. SSFT-SSD outperforms the Source-only and increases the average accuracy by 32.1 percentage points. Compared to the state-of-the-art SFUDA methods, SSFT-SSD produces the best results on seven classes while 3C-GAN produces the best results on three classes among the twelve classes. In terms of average performance over all tasks, SSFT-SSD achieves the state-of-the-art result, and outperforms PPDA with a remarkable gain of 5.7 percentage points. and outperforms the complex model 3C-GAN with 0.5 percentage points. Moreover, SSFT-SSD notably outperforms both SHOT-IM and SHOT on this dataset.

These results suggest that our proposed novel surrogate source data generation method is very effective in coping with the inaccessibility of the source training data, and can enable the successful deployment of standard UDA strategies in the SFUDA setting.

5 Conclusion

In this paper, we proposed a surrogate source training data generation method to enable the reuse of existing UDA methods in the source-free unsupervised domain adaptation (SFUDA) setting. Instead of introducing additional generative models, the method simply fixes the given model parameters and generates the input data by minimizing the supervised training loss from the optimization perspective. To avoid local fittings to individual instances, we further enforce a necessary optimality condition for the prediction model as constraints and solve the resulted problem using an ADMM algorithm. Our experimental results show that the standard UDA methods with the generated surrogate source data can yield similar adaptation performance as with the original source data. We further extend the deployment of UDA with the generated surrogate data into a semi-supervised fine-tuning adaptation framework by incorporating a mutual information term. The experimental results on several benchmark datasets show this framework yields the state-of-the-art SFUDA performance.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[2] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *ICLR*, 2020.

[3] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[4] Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. Homm: Higher-order moment matching for unsupervised domain adaptation. In *AAAI*, 2020.

[5] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *ICLR*, 2018.

[6] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

[7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[9] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, 2005.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[11] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, 2019.

[12] Youngeun Kim, Donghyeon Cho, and Sungeun Hong. Towards privacy-preserving domain adaptation. *IEEE Signal Processing Letters*, 27:1675–1679, 2020.

[13] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *CVPR*, 2020.

[14] Jogendra Nath Kundu, Naveen Venkat, Ambareesh Revanur, R Venkatesh Babu, et al. Towards inheritable models for open-set domain adaptation. In *CVPR*, 2020.

[15] Vinod K Kurmi, Venkatesh K Subramanian, and Vinay P Namboodiri. Domain impression: A source data free domain adaptation method. In *WACV*, 2021.

[16] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, 2019.

[17] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013.

[18] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, 2020.

[19] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020.

[20] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

[21] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018.

[22] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[23] Xingchao Peng, Ben Usman, Neela Kaushik, Dequan Wang, Judy Hoffman, and Kate Saenko. Visda: A synthetic-to-real benchmark for visual domain adaptation. In *CVPR Workshops*, 2018.

[24] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[25] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, 2018.

[26] Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-T approach to unsupervised domain adaptation. In *ICLR*, 2018.

[27] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.

[28] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018.

[29] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

[30] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, 2019.