Generative Alignment of Posterior Probabilities for Source-free Domain Adaptation

Sachin Chhabra
Arizona State University,
Tempe, AZ, USA
schhabr6@asu.edu

Hemanth Venkateswara
Georgia State University,
Atlanta, GA, USA
hvenkateswara@gsu.edu

Baoxin Li
Arizona State University,
Tempe, AZ, USA
baoxin.li@asu.edu

Abstract

Existing domain adaptation literature comprises multiple techniques that align the labeled source and unlabeled target domains at different stages, and predict the target labels. In a source-free domain adaptation setting, the source data is not available for alignment. We present a source-free generative paradigm that captures the relations between the source categories and enforces them onto the unlabeled target data, thereby circumventing the need for source data without introducing any new hyper-parameters. The adaptation is performed through the adversarial alignment of the posterior probabilities of the source and target categories. The proposed approach demonstrates competitive performance against other source-free domain adaptation techniques and can also be used for source-present settings.

1. Introduction

A classifier model trained on one dataset (source), generally underperforms when tested on data from a different dataset (target). This is due to the distribution difference between the two datasets [3]. Unsupervised domain adaptation aims to adapt the classifier to the target dataset without using target labels. It has been studied extensively over the past few years [39, 42].

Unsupervised domain adaptation requires both the labeled source dataset and the unlabeled target dataset when learning a classifier for the target. Access to a source dataset may not be available owing to security and privacy constraints. Source-free domain adaptation assumes only the presence of a pre-trained source classifier and the unlabeled target dataset [21]. Unsupervised domain adaptation uses source-target alignment approaches like feature alignment [10, 30, 37, 8, 35] and pixel alignment [13, 4, 33] to perform domain adaptation. These approaches are generally not possible in the source-free setting.

In this paper, we propose a novel approach that models a generative paradigm governed by a joint probability $p(\boldsymbol{x}, \boldsymbol{y})$ where $\boldsymbol{x}$ is the visible data and $\boldsymbol{y}$ are latent (labels) variables. We propose an approximation $q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ to the unknown posterior probabilities $p(\boldsymbol{y}|\boldsymbol{x})$. We demonstrate that a good approximation of the posterior probability $q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) \approx p(\boldsymbol{y}|\boldsymbol{x})$ can be learned by aligning the predicted posterior distribution $q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ with the class prior $p(\boldsymbol{y})$. We present arguments to establish that the generative paradigm is equivalent to the source-free domain alignment setting when we align the source and target posterior probabilities using adversarial alignment. Specifically, we circumvent the need for source data by generating the source category distribution $\hat{p}_s(\boldsymbol{y})$ using a conditional generative adversarial framework. Domain adaptation is achieved by enforcing the target posterior distribution to align with the source category distribution using adversarial alignment. In place of the traditional image or feature alignment, we proceed with alignment in the label space.

We present a **G**enerative model for the **A**lignment of **P**osterior probabilities (GAP) of the source and target to perform source-free and unsupervised domain adaptation. Some of the highlights of the GAP model are: (1) GAP does not introduce any new hyper-parameters and hence, does not require any additional hyperparameter tuning; (2) GAP is robust to variations in batch size (3) GAP can effectively exploit source data (when present) to enforce inter-class relationships through knowledge distillation. [12].

2. Related Work

2.1. Unsupervised Domain Adaptation

Most unsupervised domain adaptation methods attempt to reduce the misalignment between the source and target domains while training a common classifier for classifying the source and target data. Domain alignment is achieved by aligning the source and target features extracted using a deep neural network. Adversarial alignment using

Generative Adversarial Networks is the most popular approach for feature alignment [10, 37, 23]. Otherwise, distance metrics like Maximum Mean Discrepancy [38], Wasserstein distance [35, 20], and Moment matching [47] are used for feature alignment. Pixel-based approaches translate images from one domain into another and then train a common classifier for the domains [13, 4]. A combination of these approaches has also been proposed in [24, 13, 9, 46].

2.2. Source-free Domain Adaptation

Unsupervised domain adaptation methods require access to source and target data at the same time to perform domain alignment. Source-free domain adaptation estimates labels for the target data without using source data.

Popular approaches include a combination of entropy minimization and diversity maximization in addition to other objective functions during training. Entropy minimization ensures that the posterior probabilities predicting the label for the target have low entropy. While minimizing posterior entropy is necessary, it can result in a trivial solution where all the target samples are assigned to one class alone. Therefore entropy minimization is accompanied by diversity maximization, which ensures that the average posterior probability for a random subset of target samples is uniformly distributed. [21] used entropy minimization+diversity maximization loss along with a pseudo-labeling procedure to adapt the network to the target domain. [16] adjusted the deep features of the target data to match the stored source batch-norm parameters along with entropy minimization+diversity maximization loss. [1] used virtual adversarial training [27] and K-means over the target pseudo labels along with entropy minimization+diversity maximization loss.

Other approaches like [14] generate images translates the target image to source style using batch norm statistics. [18] generates target like image with low entropy on the source-trained model.

3. Method

3.1. Problem Statement

Let $D_s = \{\boldsymbol{x}_s^{(i)}, \boldsymbol{y}_s^{(i)}\}_{i=1}^{n_s}$ be the source dataset where n_s represents the number of labelled training samples and $\boldsymbol{y}_s$ is the one-hot representation of the source label with K categories. The unlabeled target dataset is $D_t = \{\boldsymbol{x}_t^{(i)}\}_{i=1}^{n_t}$ with n_t samples. The datasets D_s and D_t are drawn from distributions p_s and p_t with $p_s(\boldsymbol{x}, \boldsymbol{y}) \neq p_t(\boldsymbol{x}, \boldsymbol{y})$, but they share the same label space with identical K categories. The goal of unsupervised domain adaptation is to estimate the labels $\{\hat{\boldsymbol{y}}_t^{(i)}\}_{i=1}^{n_t}$ corresponding to elements in D_t. Source-free domain adaptation is a more restricted setup where the source and target data are not accessible at the same time. In source-free domain adaptation, once the source classifier $f_{\boldsymbol{\theta}}(.)$ has been trained using D_s, we loose access to D_s. We propose to predict target labels using D_t and the source classifier $f_{\boldsymbol{\theta}}(.)$.

3.2. Generative Model

We propose a generative paradigm to estimate the labels for the target dataset. Let the images from the target dataset be sampled from $\boldsymbol{x} \in \mathcal{X}$, where $\mathcal{X}$ is the space of images. The corresponding labels are one-hot binary vectors of the type $\boldsymbol{y} \in \{0,1\}^K$, where $\sum_k y_k = 1$. K is the number of distinct categories in the target dataset. The images and labels are sampled from an unknown target distribution $p_t(\boldsymbol{x}, \boldsymbol{y})$. Given a target sample $\boldsymbol{x}$, we intend to estimate the posterior $p_t(\boldsymbol{y}|\boldsymbol{x})$ using which we arrive at the label $\boldsymbol{y}$. We propose to approximate $p_t(.)$ using a parametric model $q_{\boldsymbol{\theta}}(.)$. In essence we seek to estimate parameter $\boldsymbol{\theta}$ such that $p_t(\boldsymbol{y}|\boldsymbol{x}) \approx q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$.

In order to estimate $\boldsymbol{\theta}$, we begin with estimating the reverse Kullback-Leibler (KL) divergence $\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})||p_t(\boldsymbol{y}|\boldsymbol{x}))$,

$$
\begin{aligned}
&\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})||p_t(\boldsymbol{y}|\boldsymbol{x})) = \\
&\quad = \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})}\big[\log q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) - \log p_t(\boldsymbol{y}|\boldsymbol{x})\big] \\
&\quad = \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})}\big[\log q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) - \log p_t(\boldsymbol{x}|\boldsymbol{y}) - \log p_t(\boldsymbol{y}) \\
&\qquad\qquad + \log p_t(\boldsymbol{x})\big] \\
&\quad = \mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})||p_t(\boldsymbol{y})) - \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})}\big[\log p_t(\boldsymbol{x}|\boldsymbol{y})\big] \\
&\qquad\qquad + \log p_t(\boldsymbol{x}) \\
&\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})||p_t(\boldsymbol{y}|\boldsymbol{x})) \leq \mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})||p_t(\boldsymbol{y})) \\
&\qquad\qquad - \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})}\big[\log p_t(\boldsymbol{x}|\boldsymbol{y})\big]. \quad (1)
\end{aligned}
$$

In deriving Eq.1 we have used $p_t(\boldsymbol{y}|\boldsymbol{x}) = \frac{p_t(\boldsymbol{x}|\boldsymbol{y})p_t(\boldsymbol{y})}{p_t(\boldsymbol{x})}$ and $\mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})}[\log p_t(\boldsymbol{x})] = \log p_t(\boldsymbol{x}) \leq 0$. The R.H.S in Eq.1 is an upper bound for the KL-divergence between the distributions $q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ and $p_t(\boldsymbol{y}|\boldsymbol{x})$. The value of $\boldsymbol{\theta}$ which will minimize the R.H.S will align the two distributions. The 1st term on the R.H.S is the measure of alignment between the unknown prior distribution $p_t(\boldsymbol{y})$ and the generative model $q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$. The second term can be viewed as the expected reconstruction error converting from $\boldsymbol{y}$ to $\boldsymbol{x}$.

3.3. Model Assumptions

A few assumptions are made to further simplify the model. $p_t(\boldsymbol{y})$ is unknown, but the source and the target have the same label space. We assume $p_s(\boldsymbol{y}) \approx p_t(\boldsymbol{y})$, which is a reasonable assumption along the lines of assuming covariate-shift ($p_s(\boldsymbol{y}|\boldsymbol{x}) \approx p_t(\boldsymbol{y}|\boldsymbol{x})$) [3] or concept-shift ($p_s(\boldsymbol{x}) \approx p_t(\boldsymbol{x})$) [41, 32]. We model $q_{\boldsymbol{\theta}}(.)$ using a neural network which takes $\boldsymbol{x}$ as input and yields the posterior

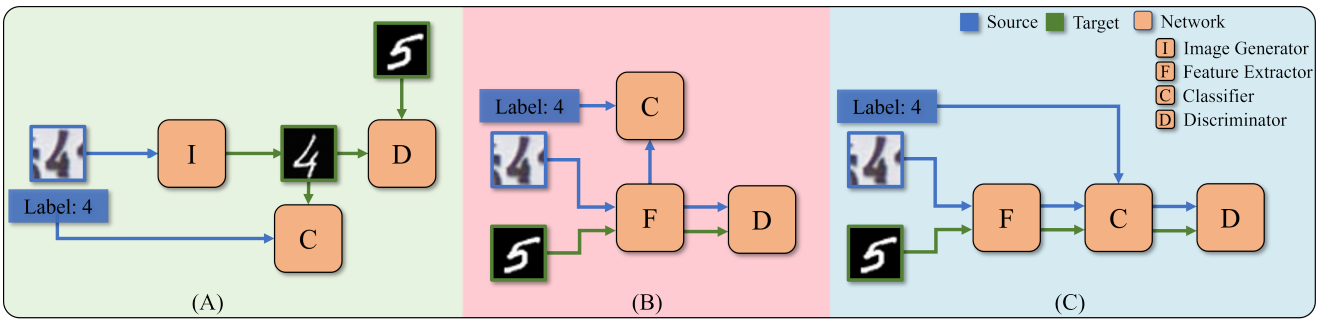

Figure 1. Variations in domain alignment. **A.** Pixel-level alignment: Adversarial image translation is applied to translate source images into target images before using a common classifier. **B.** Feature-level alignment: The source and target features of a deep neural network are aligned before applying a common classifier. The figure depicts an adversarial feature alignment architecture. **C.** Generative Alignment of Posterior probabilities (GAP): We propose to align the posterior probabilities of the source and target classifiers using an adversarial framework. Although the image indicates the presence of a source, the GAP model can be used for both source-free and unsupervised domain adaptation.

$q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ as output. The proposed model has issues of *identifiability*. A model is identifiable when $p(\boldsymbol{x})$ has a unique decomposition in $\sum_{\boldsymbol{y}} p(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y})$ [5]. If a swapping of labels can yield the same marginal $p(\boldsymbol{x})$, the problem setup is not identifiable and the resulting solution is not consistent (different assignment of labels under different initial conditions). The 2nd term on the R.H.S of Eq. 1 can be viewed as a reconstruction term that can ensure identifiability. However, $p_t(\boldsymbol{x}|\boldsymbol{y})$ is unknown. We address both the problems by dropping the 2nd term and initializing the parameters in the network $q_{\boldsymbol{\theta}}(.)$ with the parameters of a pretrained source classifier. This is a robust initialization mechanism that will bias the model $q_{\theta}(.)$ to yield a consistent mapping between $\boldsymbol{x}$ and $\boldsymbol{y}$. Under these assumptions, we minimize $\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})||p_s(\boldsymbol{y}))$ in an attempt to align the distributions $q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ and $p_t(\boldsymbol{y}|\boldsymbol{x})$.

3.4. Source Replicator

In the absence of the source data, we cannot align the probabilities for the source and target directly. Therefore, we aim to replicate source prior probability vectors for target alignment. We perform this additional step of source replication after training a source network. Specifically, we design a conditional generative adversarial framework [26] - a generator $G_c(.;\boldsymbol{\theta}_c)$ that takes fake label ($\boldsymbol{y}_f \in \mathcal{Y}$, where $\mathcal{Y} = \{0,1\}^K$) and noise ($\boldsymbol{z} \in \mathcal{N}(0,I)$) as input to generate a probability vector of K-dimensions (using softmax activation at the last layer). The conditional discriminator $D_c(.;\boldsymbol{\phi}_c)$ is trained to discriminate between generated probabilities and source probabilities from the pre-trained source classifier $G(.,\theta)$. The conditional framework is preferred over a vanilla GAN primarily to avoid partial mode collapse and also to have control over the prior class distribution. We refer to the conditional generator as the Source Replicator in this paper. On account of its stability, we train the conditional GAN using the least squared loss function [25],

$$\min_{\boldsymbol{\phi}_c} \frac{1}{2}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D_s}[(D_c(G(\boldsymbol{x};\boldsymbol{\theta}),\boldsymbol{y};\boldsymbol{\phi}_c)-1)^2] + \frac{1}{2}\mathbb{E}_{\substack{\boldsymbol{y}_f\sim\mathcal{Y}\\ \boldsymbol{z}\in\mathcal{N}(0,I)}}[(D_c(G_c(\boldsymbol{z},\boldsymbol{y}_f;\boldsymbol{\theta}_c),\boldsymbol{y}_f;\boldsymbol{\phi}_c))^2],$$

$$\min_{\boldsymbol{\theta}_c} \frac{1}{2}\mathbb{E}_{\substack{\boldsymbol{y}_f\sim\mathcal{Y}\\ \boldsymbol{z}\in\mathcal{N}(0,I)}}[(D_c(G_c(\boldsymbol{z},\boldsymbol{y}_f;\boldsymbol{\theta}_c),\boldsymbol{y}_f;\boldsymbol{\phi}_c)-1)^2]. \quad (2)$$

The conditional framework captures the statistical variations of the source predictions and their inter-class relations. We use the trained conditional generator G_c to replicate (generate) source probabilities $p_s(\boldsymbol{y})$ when minimizing $\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})||p_s(\boldsymbol{y}))$. Figure 2A is the pre-trained source classifier. Figure 2B depicts the training of the Source Replicator G_c.

3.5. Source-free Domain Alignment

Domain adaptation approaches rely on the source data to perform source-target alignment either in the pixel space through image translation [13, 4] or in the feature space [38, 10] (see Figure 1). The proposed model can be interpreted as the alignment of posterior probabilities $p_s(\boldsymbol{y}|\boldsymbol{x})$ and $p_t(\boldsymbol{y}|\boldsymbol{x})$, where we align the source and target in the final stage of the classification process. As the input image $\boldsymbol{x}$ propagates through the classification framework (neural network), it is transformed from an image to a feature vector and finally to a probability vector while decreasing its information content and complexity. We hypothesize that source-target alignment in the high-dimensional pixel space and feature space is complex and less effective compared to the alignment of probability vectors. Also, the feature space is constantly changing as the network trains whereas, the probability space has fewer variations.

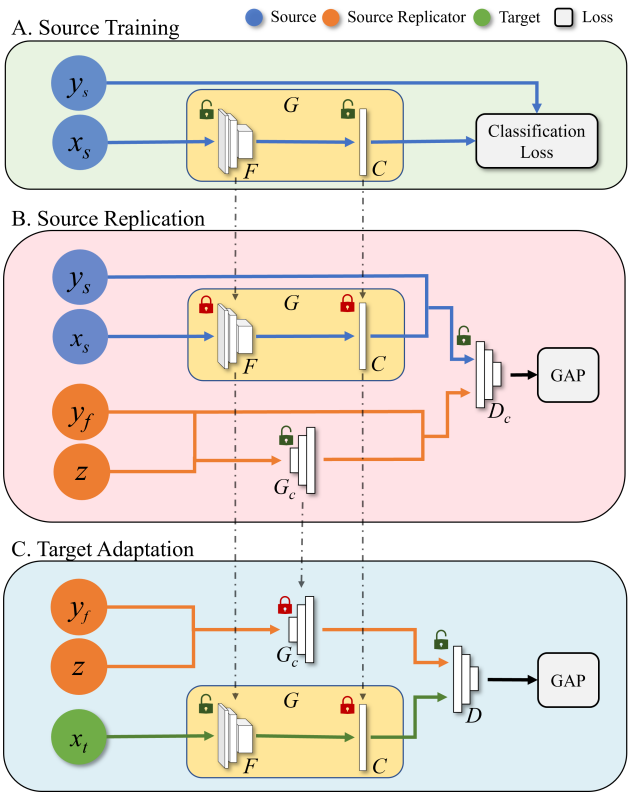

Figure 2. Model diagram for the GAP. **(A)** The feature extractor F and Classifier C are trained using the source data. **(B)** Source replicator (conditional generator) G_c is trained to replicate source predictions. $\boldsymbol{y}_f$ represents the fake label. **(C)** During the target adaptation, the C component is frozen to retain the source classification boundaries. The feature extractor F and classifier C represent the Generator G, and are initialized using the source network. The Source replication is performed using Eq.2 and the target adaptation uses Eq.3.

We implement a Generative Adversarial Network (GAN) to align the distributions $q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ and $p_s(\boldsymbol{y})$ [11]. The Generator network $G(.;\boldsymbol{\theta})$ models $q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$, where $\boldsymbol{\theta}$ are the parameters of $G(.;\boldsymbol{\theta})$, $\boldsymbol{x}$ is the target input image and $\boldsymbol{y}$ is the predicted label. The source prior $p_s(\boldsymbol{y})$ is modeled by the Source Replicator $G_c(.)$. We initialize the Generator $G(.;\boldsymbol{\theta})$ with the parameters of a pre-trained source classifier. The similarity between the source and target domains is exploited to provide a near-optimal initialization for $\boldsymbol{\theta}$. The Discriminator network $D(.;\boldsymbol{\phi})$, with parameters $\boldsymbol{\phi}$, is trained to distinguish between the Generator output $G(\boldsymbol{x};\boldsymbol{\theta}) \rightarrow q_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ and outputs of the Source Replicator, $G_c(\boldsymbol{z},\boldsymbol{y}_f;\boldsymbol{\theta}_c)$. Figure 2C depicts target adaptation where only the feature extractor (F) in the pre-trained classifier $G(.)$ is trained. The parameters of the classifier (C) are frozen to anchor the classifier and ensure the target alignment does not drift away to yield a trivial solution in the absence of source data for alignment. The GAN objective is based on the mean-squared loss function [25],

$$\min_{\boldsymbol{\phi}} \frac{1}{2}\mathbb{E}_{\substack{\boldsymbol{y}\sim\mathcal{Y} \\ \boldsymbol{z}\in\mathcal{N}(0,I)}}[(D(G_c(\boldsymbol{z},\boldsymbol{y};\boldsymbol{\theta}_c);\boldsymbol{\phi})-1)^2] + \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim D_t}[(D(G(\boldsymbol{x};\boldsymbol{\theta});\boldsymbol{\phi}))^2],$$
$$\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim D_t}[(D(G(\boldsymbol{x};\boldsymbol{\theta});\boldsymbol{\phi})-1)^2]. \quad (3)$$

where the input to the Source Replicator $\boldsymbol{y} \in \mathcal{Y}$ is a 1-of-K binary vector, where $p(\boldsymbol{y}|\boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{\bar{y}_k}$. Following [21], we assume the mixing components in $\boldsymbol{\pi}$ to be a uniform prior with $\pi_k = K^{-1}\ \forall k$. The uniform prior can be replaced with the source prior distribution. However, we found empirically that uniform prior performs better than the latter (see supplementary material).

In Figure 2C, the feature extractor (F) and classifier (C) together form the Generator G. To account for smoother posterior probabilities, we apply label smoothing with a constant $\epsilon = 0.1$ [36] while training the source network. Label smoothing results in better generalization accuracy and has been adopted in many source-free domain adaptation approaches [21, 16, 2]. The parameters of the Generator $G(.;\boldsymbol{\theta})$ are initialized using a pre-trained source network. Following [21], we only train the feature extractor (F) and fix the classifier (C) during the target adaptation phase.

Method	S→M	M→U	U→M	Avg
Source	67.1	82.2	69.6	73.0
Source + LS	70.2	79.7	88.0	79.3
SFIT [14]	90.4	84.7	82.3	85.8
SDDA [18]	76.3	88.5	-	-
SHOT-IM [21]	99.0	97.6	97.7	98.2
SHOT-Full [21]	98.9	**98.0**	97.9	98.3
GAP (Ours)	**99.1**	97.6	**98.4**	**98.4**
Target-Supervised (Oracle)	99.4	98.0	99.4	98.8

Table 1. Comparison of source-free classification accuracies on the digits dataset. Bold numbers represent the highest accuracy and the underline denotes the second highest.

4. Experiments

4.1. Datasets

We perform our experiments on the standard source-free domain adaptation settings. For digits experiments, we use SVHN → MNIST, MNIST → USPS and USPS → MNIST combinations [29, 19, 15]. For object recognition, we use Office-31 [34], Office-Home [40] and VisDa-C [31] datasets.

Source	Ar			Cl			Pr			Rw			Avg
Target	Cl	Pr	Rw	Ar	Pr	Rw	Ar	Cl	Rw	Ar	Cl	Pr	
Source	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
Source + LS	44.6	67.3	74.8	52.7	62.7	64.8	53.0	40.6	73.2	65.3	45.4	78.0	60.2
Rob. Adapt [1]	-	-	-	-	-	-	-	-	-	-	-	-	65.1
SFDA [17]	48.4	73.4	76.9	64.3	69.8	71.7	62.7	45.3	76.6	69.8	50.5	79.0	65.7
SHOT-IM [21]	55.4	76.6	80.4	66.9	74.3	75.4	65.6	54.8	80.7	73.7	58.4	83.4	70.5
SHOT-full [21]	**57.1**	**78.1**	**81.5**	**68.0**	**78.2**	78.1	**67.4**	**54.9**	82.2	73.3	58.8	**84.3**	**71.8**
GAP (Ours)	55.4	73.4	80.8	67.2	75.5	**78.3**	65.5	54.0	**82.4**	**74.3**	**59.4**	84.0	70.8

Table 2. Comparison of source-free classification accuracies on the Office-Home dataset (ResNet-50). Bold numbers represent the highest accuracy and the underline denotes the second highest. LS stands for label smoothing.

Method	Plane	Bcycl	Bus	Car	Horse	Knife	Mcycl	Person	Plant	Sktbrd	Train	Truck	Avg
Source	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
Source + LS	60.9	21.6	50.9	**67.6**	65.8	6.3	82.2	23.2	57.3	30.6	84.6	8.0	46.6
SFIT [14]	-	-	-	-	-	-	-	-	-	-	-	-	63.5
Rob. Adapt [1]	-	-	-	-	-	-	-	-	-	-	-	-	74.9
SFDA [17]	86.9	81.7	**84.6**	63.9	93.1	91.4	**86.6**	71.9	84.5	58.2	74.5	42.7	76.7
SHOT-IM [21]	93.7	86.4	78.7	50.7	91.0	93.5	79.0	78.3	89.2	85.4	**87.9**	51.1	80.4
SHOT-Full [21]	**94.3**	**88.5**	80.1	57.3	93.1	94.9	80.7	**80.3**	**91.5**	**89.1**	86.3	**58.2**	**82.9**
GAP (Ours)	94.2	84.8	82.5	57.2	**93.8**	**95.1**	86.5	78.2	83.1	87.8	86.3	53.5	81.9

Table 3. Comparison of source-free classification accuracies on on the VisDA dataset (ResNet-101). Bold numbers represent the highest accuracy and the underline denotes the second highest. LS stands for label smoothing.

Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg
Source	68.9	68.4	62.5	96.7	60.7	99.4	76.1
Source + LS	80.8	76.9	60.3	95.3	63.6	98.7	79.3
SDDA [18]	85.3	82.5	66.4	**99.0**	67.7	99.8	83.5
Rob. Adapt [1]	-	-	-	-	-	-	87.0
SFDA [17]	92.2	91.1	71.0	98.2	71.2	99.5	87.2
SHOT-IM [21]	90.6	**91.2**	72.5	98.3	71.4	**99.9**	87.3
SHOT-full [21]	**94.0**	90.1	**74.7**	98.4	**74.3**	**99.9**	**88.6**
GAP (Ours)	90.6	90.9	74.5	98.7	73.9	99.8	88.1

Table 4. Comparison of source-free classification accuracies on the Office-31 dataset (ResNet-50). Bold numbers represent the highest accuracy and the underline denotes the second highest. LS stands for label smoothing.

4.2. Training Setup

We follow the training protocol and use the network architectures from [21]. For the digits dataset, the networks are trained from scratch. We use ResNet-50 and ResNet-101 as the backbone network for Office datasets and VisDa datasets respectively. The networks G_c and D_c consist of four fully connected layers of size 500 and discriminator D is a vanilla discriminator composed of two hidden layers of size 100. All the networks are trained on a batch size of 64 using an SGD optimizer with a momentum of 0.9. We use $1e^{-2}$ learning rate for office and $1e^{-3}$ for the VisDa dataset and decay it as $\eta = \eta_0(1+10p)^{-0.75}$ where p is the training progress. The learning rate for pre-trained layers is reduced by a factor of 10.

4.3. Results

The results for digits, Office-Home, VisDa and Office-31 are in Table 1, 2, 3 and 4 respectively. We compare our approach with source-free domain adaptation methods like SFDA [17], SDDA [18], SHOT [21]. Our method achieves comparable performance against all the baseline methods. SHOT-full outperforms our approach but it uses pseudo labeling loss along with SHOT-IM loss functions- entropy minimization and diversity maximization. Our method is much simpler, does not use any auxiliary loss or requires hyper-parameter tuning.

5. Analysis

5.1. Ablation Study

We perform an ablation study on our loss function to understand the contribution of its different components. First, we remove the label smoothing from source model training. In the second experiment, we replace the mean-squared error (MSE) loss with the binary cross-entropy (BCE) loss. We perform these experiments on Office-31, Office-home, digits datasets and the results are in Table 5. As per the

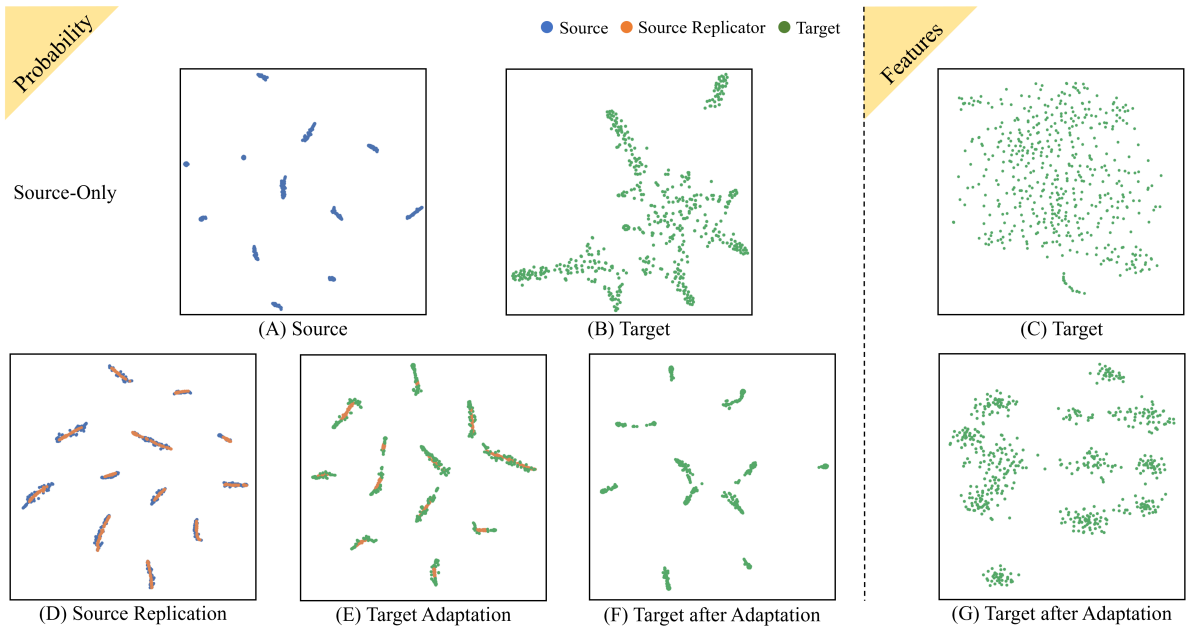

Figure 3. t-SNE visualization for VisDa in a source-free setting. Different colors represents different domains: Blue: Source; Orange: Source Replicator; Green: Target. The probability distributions are showcased in the left section and the penultimate layer features on the right. The first row is before adaptation and the second row is our approach. (A) Source probability distribution on source trained model. (B) Target probability distribution on source trained model. (C) Target features using the source-trained model. (D) Source and Source Replicator probability distribution. (E) Source Replicator and Target probability distribution after adaptation. (F) Target probability distribution after adaptation. (G) Target features after adaptation. Best viewed in color.

LS	Alignment	Office-31	Office-Home	Digits
✗	NA	76.1	46.1	73.0
✗	BCE	84.1	67.0	98.1
✗	MSE	86.3	69.8	98.1
✓	NA	79.3	60.2	79.3
✓	BCE	85.7	68.8	97.2
✓	MSE	**88.1**	**70.8**	**98.4**

Table 5. Ablation study of our method on Office-31 and OfficeHome source-free setting. 'LS' denotes label smoothing used for source-model training. 'Alignment' denotes the alignment loss used for target adaptation. NA: No alignment was performed/Source-only performance. BCE: Binary Cross-Entropy; MSE: Mean-squared-error

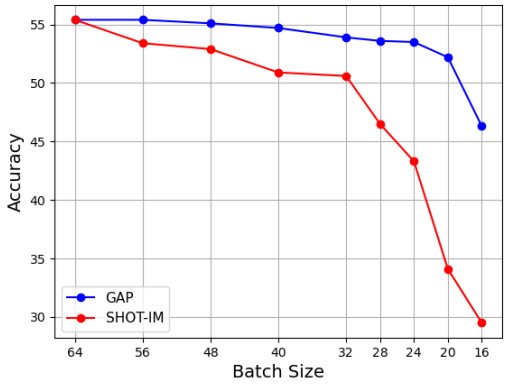

Figure 4. Comparison of our approach with SHOT-IM (Entropy minimization + Diversity maximization loss) for different batch sizes on Art → Clipart (Office-Home) source-free setting.

results, label smoothing provides benefits in generalization during source training and target domain adaptation. The MSE loss function results in slightly superior performance, mainly due to its stability and also observed that BCE loss experiences partial mode collapse frequently.

5.2. Impact of batch size

When there are a large number of categories, a mini-batch cannot contain samples from all the classes. We compare GAP with SHOT-IM (Entropy minimization + diversity maximization loss) for different batch sizes. Figure 4 shows the results of this experiment on Art → Clipart (Office-Home) source-free setting. When the batch size decreases to $\frac{1}{4}$ of its starting value, our approach has a significantly lesser decrease (approximately 10%) in performance compared to SHOT-IM (25%). GAP does not show fluctuations until the batch size reaches 20 whereas the performance of SHOT-IM drops significantly with the reduction

Method	A→D	A→W	D→A	D→W	W→A	W→D	Mean
CDAN+E [23]	92.9	94.1	71.0	98.6	69.3	**100.0**	87.7
BSP+CDAN [7]	93.0	93.3	73.6	98.2	72.6	**100.0**	88.5
ALDA [6]	**94.0**	95.6	72.2	97.7	72.5	**100.0**	88.7
SymNets [49]	93.9	90.8	74.6	**98.8**	72.5	**100.0**	88.4
TADA[45]	91.6	94.3	72.9	98.7	73.0	99.8	88.4
MADA [30]	87.8	90.0	70.3	97.4	66.4	99.6	85.2
MDD [48]	93.5	94.5	74.6	98.4	72.2	**100.0**	88.9
CDAN+TransNorm [44]	**94.0**	**95.7**	73.4	98.7	74.2	**100.0**	**89.3**
Source	68.9	68.4	62.5	96.7	60.7	99.4	76.1
DANN [10]	79.7	82.0	68.2	96.9	67.4	99.1	82.2
GAP (Ours)	84.7	89.3	72.9	97.2	72.8	99.2	86.0
GAP + KD (Ours)	88.8	91.8	75.4	97.6	73.5	99.8	87.8
Source + LS	80.8	76.9	60.3	95.3	63.6	98.7	79.3
DANN + LS [10]	78.7	86.3	69.0	94.7	71.5	99.4	83.3
GAP + LS (Ours)	92.2	92.6	**78.0**	98.2	74.0	**100.0**	89.2
GAP + LS + KD (Ours)	90.0	93.6	74.5	97.7	**74.4**	**100.0**	88.4

Table 6. Comparison of classification accuracies and abalation study on the Office-31 dataset with source present (ResNet-50). LS stands for source network trained with label smoothing and KD denotes knowledge distillation. Bold numbers represent the highest accuracy and the underline denotes the second highest.

of batch size. Due to this, all approaches using entropy minimization and diversity maximization losses are susceptible to small batch sizes. We overcome this issue by training the Discriminator multiple times before updating the Generator. This way the Discriminator is unaffected by the mini-batch bias.

5.3. t-SNE Visualization

We show t-SNE plots for visualizing the output probability space and the penultimate layer features for the VisDa dataset in Figure 3. (A) and (B) denote the output probabilities of the source and target data respectively using a model trained on the source. (C) shows the target features using the source-trained model. (D) exhibits GAP can replicate the source probability variations. (E) We train the model to have the same variations with target data. (F) and (G) are the target outputs and features after adaptation. Based on these plots, we observe that GAP ends up clustering the penultimate layer features as well.

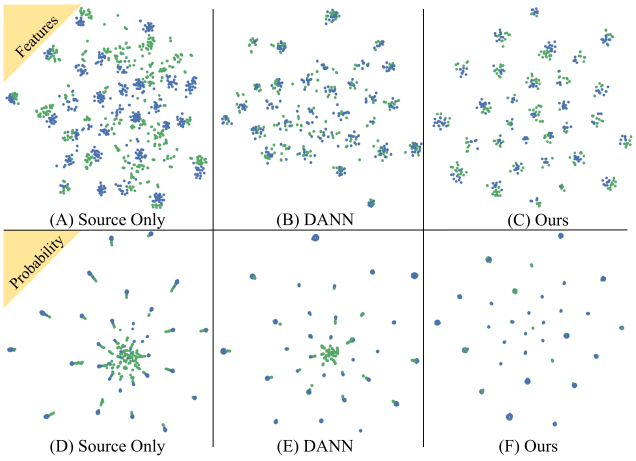

Figure 5. t-SNE visualization of the penultimate layer features (top row) and the the output probability space (bottom row) for Office31 - Amazon → Webcam with source present setting. Blue denotes Source domain (Amazon) and Green denotes Target domain (Webcam). Best viewed in color.

6. Domain Adaptation with Source Data

We evaluate GAP for regular unsupervised domain adaptation. In this scenario, we do not need to train a source replicator. Instead, we can align the target with actual source probabilities and train the network on source data as well. We use the popular Gradient Reversal layer and adapt it to our approach for these experiments. Specifically, the Discriminator is trained to discriminate between source and target output probabilities using the mean-squared loss function. The network is trained using reversed gradients from the Discriminator. In this scenario, we do not freeze the classification layer. We use Office-31 and Office-Home datasets for these experiments.

The experiment results are in Table 6 for Office-31 and Table 7 for Office-Home dataset. We compare GAP with domain adaptation methods like CDANN [7], SymNets [49], TADA [45] in the top section. In our approach, we consider both the cases of training the network - without label smoothing (midsection) and with label smoothing (bottom section). In the mid and bottom sections, the first row denotes the performance of source-only models. The

Source	Ar			Cl			Pr			Rw			Mean
Target	Cl	Pr	Rw	Ar	Pr	Rw	Ar	Cl	Rw	Ar	Cl	Pr	
CDAN+E [23]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
CDAN+BSP [7]	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	59.3	81.9	66.3
ALDA [6]	53.7	70.1	76.4	60.2	72.6	71.5	56.8	51.9	77.1	70.2	56.3	82.1	66.6
SymNets [49]	47.7	72.9	78.5	64.2	71.3	74.2	64.2	48.8	79.5	74.5	52.6	82.7	67.6
TADA [45]	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
CDAN+TransNorm [44]	50.2	71.4	77.4	59.3	72.7	73.1	61.0	53.1	79.5	71.9	59.0	82.9	67.6
MDD [48]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	**60.2**	82.3	68.1
Source	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN [10]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
GAP (Ours)	54.5	71.1	78.5	61.6	73.7	71.7	61.6	54.2	80.4	72.5	57.4	83.8	68.4
GAP + KD (Ours)	56.5	72.6	79.2	64.5	74.3	74.0	64.8	**57.7**	82.3	74.2	57.4	84.7	70.2
Source + LS	44.6	67.3	74.8	52.7	62.7	64.8	53.0	40.6	73.2	65.3	45.4	78.0	60.2
LS + DANN [10]	50.9	59.3	73.0	54.6	67.2	69.8	55.2	53.7	76.0	68.4	58.7	79.3	63.8
GAP + LS (Ours)	**57.5**	**74.3**	78.7	**65.6**	74.5	75.2	65.4	57.1	81.6	**75.4**	59.8	**85.6**	**70.9**
GAP + LS + KD (Ours)	55.6	73.3	78.1	64.9	72.6	74.5	65.0	57.6	81.4	73.7	59.6	84.6	70.1

Table 7. Comparison of classification accuracies on the Office-Home dataset with source present (ResNet-50). LS is for source network trained with Label Smoothing and KD denotes Knowledge Distillation. Bold numbers represent the highest accuracy and the underline denotes the second highest.

second-row results are when we align the deep features using the vanilla DANN loss function for baseline comparison [10]. The third row shows the performance of GAP. In the last row, we combine our approach with knowledge distillation [12]. We soften the probabilities using a temperature equal to 2 to align the inter-class relationships between the domains. We do not use temperature for the source-free experiments as the logits space between the source and target is not the same. Hence, using a temperature hurts the performance.

GAP outperforms all the listed baselines for source-present scenarios on both datasets. The compared approaches use advanced techniques like attention [45] or complex architectures like SymNets [49], whereas the GAP is a simple model based on adversarial alignment. Knowledge distillation boosts the performance when no label smoothing is used. Using knowledge distillation with label smoothing results in a negative transfer as label smoothing places deep features of the classes at equal distance from each other and disturbs the inter-class relationships [28]. We also show the t-SNE plots of the penultimate features and the probability outputs for Office-31 Amazon to Webcam in Figure 5. GAP aligns the probabilities better than DANN and results in similar feature alignment [10] without explicitly aligning it.

7. Limitations

Although GAP has several advantages and yields good performance, we discuss the scenarios where it can underperform. GAP is based on $p_s(\boldsymbol{y}) \approx p_t(\boldsymbol{y})$ assumption, and our experiments on multiple datasets that it is a reasonable assumption and works for the majority of the cases. However, this can affect the performance negatively when the target labels follow a long-tail distribution and would affect the baseline methods as well. On the other hand, if the distribution were known, it could be used as a prior to enhance the alignment. GAP assumes the source and target have identical label spaces. In its current form GAP cannot be extended to *OpenSet* and *Partial* domain adaptation cases where the label spaces of the source and target are not identical. Exploring these settings for our method can be an interesting future direction.

8. Conclusions

In this paper, we present a model for the Generative Alignment of Posterior probabilities (GAP) for source-free domain adaptation. GAP uses a Source Replicator (probability generator) that mimics the variations in the posterior probabilities of the source classes and then aligns target posterior probabilities to it through adversarial alignment. Through extensive experiments, we show the approach is robust to smaller batch sizes, does not introduce any new hyper-parameters and yields comparable performance to the compared baselines for source-free and source-present domain adaptation scenarios.

Acknowledgements

The work was supported in part by a grant from ONR. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of ONR.

References

[1] Peshal Agarwal, Danda Pani Paudel, Jan-Nico Zaech, and Luc Van Gool. Unsupervised robust domain adaptation without source data. *CoRR*, abs/2103.14577, 2021.

[2] Sk Miraj Ahmed, Dripta S Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10103–10112, 2021.

[3] 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(1-2):151–175, 2010.

[4] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, pages 3722–3731, 2017.

[5] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning. 2006. *Cambridge, Massachusettes: The MIT Press View Article*, 2006.

[6] Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. Adversarial-learned loss for domain adaptation. In *AAAI*, pages 3521–3528, 2020.

[7] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International Conference on Machine Learning*, pages 1081–1090, 2019.

[8] Sachin Chhabra, Prabal Bijoy Dutta, Baoxin Li, and Hemanth Venkateswara. Glocal alignment for unsupervised domain adaptation. In *Multimedia Understanding with Less Labeling on Multimedia Understanding with Less Labeling*, pages 45–51. 2021.

[9] Sachin Chhabra, Hemanth Venkateswara, and Baoxin Li. Iterative image translation for unsupervised domain adaptation. In *1st Workshop on Multimedia Understanding with Less Labeling, MULL 2021, co-located with ACM MM 2021*, pages 37–44. Association for Computing Machinery, Inc, 2021.

[10] 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.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[12] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[13] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018.

[14] Yunzhong Hou and Liang Zheng. Source free domain adaptation with image translation. *CoRR*, abs/2008.07514, 2020.

[15] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[16] Masato Ishii and Masashi Sugiyama. Source-free domain adaptation via distributional alignment by matching batch normalization statistics. *CoRR*, abs/2101.10842, 2021.

[17] Youngeun Kim, Donghyeon Cho, Priyadarshini Panda, and Sungeun Hong. Progressive domain adaptation from a source pre-trained model. *arXiv preprint arXiv:2007.01524*, 2020.

[18] Vinod K Kurmi, Venkatesh K Subramanian, and Vinay P Namboodiri. Domain impression: A source data free domain adaptation method. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 615–625, 2021.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10285–10295, 2019.

[21] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020.

[22] Rui Lix, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020.

[23] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in neural information processing systems*, pages 1640–1650, 2018.

[24] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017.

[25] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[26] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[27] 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.

[28] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in Neural Information Processing Systems*, 32:4694–4703, 2019.

[29] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[30] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[31] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *CoRR*, abs/1710.06924, 2017.

[32] Anton Popovič. The concept "shift of expression" in translation analysis. In *The nature of translation*, pages 78–88. De Gruyter Mouton, 2011.

[33] Paolo Russo, Fabio M Carlucci, Tatiana Tommasi, and Barbara Caputo. From source to target and back: Symmetric bi-directional adaptive GAN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8099–8108, 2018.

[34] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[35] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[37] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[38] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014.

[39] Hemanth Venkateswara, Shayok Chakraborty, and Sethuraman Panchanathan. Deep-learning systems for domain adaptation in computer vision: Learning transferable feature representations. *IEEE Signal Processing Magazine*, 34(6):117–129, 2017.

[40] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[41] Peter Vorburger and Abraham Bernstein. Entropy-based concept shift detection. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 1113–1118. IEEE, 2006.

[42] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[43] Rui Wang, Zuxuan Wu, Zejia Weng, Jingjing Chen, Guo-Jun Qi, and Yu-Gang Jiang. Cross-domain contrastive learning for unsupervised domain adaptation. *IEEE Transactions on Multimedia*, 2022.

[44] Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Transferable normalization: Towards improving transferability of deep neural networks. *Advances in neural information processing systems*, 32, 2019.

[45] Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. Transferable attention for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5345–5352, 2019.

[46] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6502–6509, 2020.

[47] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (CMD) for domain-invariant representation learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[48] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413. PMLR, 2019.

[49] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5031–5040, 2019.