Learning Unbiased Transferability for Domain Adaptation by Uncertainty Modeling

Jian Hu[1][*], Haowen Zhong[2][*], Fei Yang[2],
Shaogang Gong[1], Guile Wu, Junchi Yan[3]

[1] Queen Mary University of London
[2] Zhejiang Lab
[3] Shanghai Jiao Tong University
{jian.hu,s.gong}@qmul.ac.uk, guile.wu@outlook.com
{zhonghw,yangf}@zhejianglab.com, yanjunchi@sjtu.edu.cn

Abstract. Domain adaptation (DA) aims to transfer knowledge learned from a labeled source domain to an unlabeled or a less labeled but related target domain. Ideally, the source and target distributions should be aligned to each other equally to achieve unbiased knowledge transfer. However, due to the significant imbalance between the amount of annotated data in the source and target domains, usually only the target distribution is aligned to the source domain, leading to adapting unnecessary source specific knowledge to the target domain, i.e., biased domain adaptation. To resolve this problem, in this work, we delve into the transferability estimation problem in domain adaptation and propose a non-intrusive Unbiased Transferability Estimation Plug-in (UTEP) by modeling the uncertainty of a discriminator in adversarial-based DA methods to optimize unbiased transfer. We theoretically analyze the effectiveness of the proposed approach to unbiased transferability learning in DA. Furthermore, to alleviate the impact of imbalanced annotated data, we utilize the estimated uncertainty for pseudo label selection of unlabeled samples in the target domain, which helps achieve better marginal and conditional distribution alignments between domains. Extensive experimental results on a high variety of DA benchmark datasets show that the proposed approach can be readily incorporated into various adversarial-based DA methods, achieving state-of-the-art performance.

Keywords: Unbiased transferability estimation, domain adaptation, pseudo labeling

1 Introduction

With the rise of deep neural networks, convolutional neural networks (CNNs) [28] and transformers [47], supervised learning based on deep neural networks has shown promising performance in various vision and language tasks. Conventional supervised learning methods mostly assume that the training (source) domain
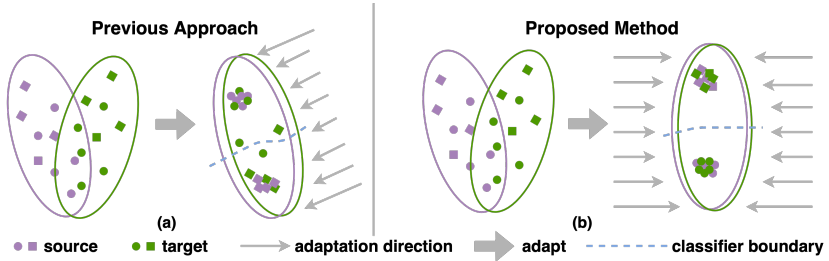
---

[*] Equal contribution

Fig. 1: **Motivation for Unbiased Transferability Estimation.** The left and right parts in (**a**) (**b**) are distributions of source and target domains before and after adaptation respectively. (**a**) Previous methods [16,41] focus on aligning target samples to source well-clustered classes (distribution), leading to misalignment near the classifier boundary and transferring unnecessary source specific knowledge to the target, and the decision boundaries remain ambiguous for some target samples near the boundaries. (**b**) Our method enforces source and target distributions to align to each other equally, to help unbiased DA to achieve better marginal and conditional distribution alignment. The target decision boundary is further optimized with uncertainty-based pseudo label selection.

and the test (target) domain are subject to the i.i.d. (independent and identically distributed) hypothesis. As a consequence, they usually show poor generalization in the test domain in the existence of a domain gap [21]. Moreover, deep neural networks have ravenous appetite for a large amount of labeled data, which explores extra linkage between labelled data in a source domain and unlabeled data in a target domain. Domain adaption aims to resolve this quandary by leveraging previously labeled datasets [45, 55] to realize effective knowledge transfer across domains. DA can further divide into unsupervised domain adaptation (UDA) and semi-supervised domain adaptation (SSDA) based on the accessibility of labeled data in the target domain.

Transferability indicates the ability of representations to bridge the discrepancy across domains [8]. Contemporary deep DA methods mainly focus on exploring cross-domain transferability to narrow the gap between domains [51,61]. There are two typical approaches: 1) High-order moment matching [40,68], which reduces the distribution discrepancy between domains by minimizing the distance between high-dimensional features; 2) Adversarial learning [16,23,54], in which a domain discriminator and a feature extractor play a two-player game to align distributions between domains. Although existing DA methods have shown promising performance on resolving domain shift, there remain two major problems. First, most existing methods tend to promote the alignment of domains to encourage transferability, which is evaluated by measuring the distribution discrepancy (estimated transferability) [2,39]. However, even if the distribution discrepancy is completely eliminated, the deviation of the estimated and real transferability still exists, leading to negative transfer (see Figure 1(a)). Second, existing approaches only consider aligning the overall distributions between do-

Table 1: A comparison of uncertainty modeling methods in DA.

Methods	online learning	non-intrusive design	unbiased transfer	uncertainty usage
CPCS [46]	✓	✗	✗	weighting for calibration
PACET [37]	✓	✗	✗	sample selection
BUM [57]	✓	✗	✗	weighting
TransCal [56]	✗	✓	✗	post-hoc calibration
CADA [33]	✓	✗	✗	weighting for attention
UTEP(Ours)	✓	✓	✓	weighting + regularization + sample selection

mains and tend to assign larger weights to samples that are not sufficiently transferred [39, 57]. However, since the number of annotations in the source and target domains is extremely imbalanced, the transferability is usually estimated by the target sample score on the source domain classifier, resulting in a strong bias towards the source domain (see Figure 1(a)). In other words, under the supervision of source labeled samples, the source domain distribution is well learned and hardly affected by the target one, while the target domain distribution tends to match the source domain instead of encouraging the source to align the target domain. As a consequence, the target domain is not only learning the domain invariant knowledge (features) but also learning domain specific knowledge of the source domain, resulting in *biased domain adaptation*. Therefore, to address these two problems, it is essential to learn unbiased transferability to facilitate knowledge transfer across domains in domain adaptation.

In this work, we focus on the transferability problem in domain adaptation and present a non-intrusive Unbiased Transferability Estimation Plug-in (UTEP) model by uncertainty variance modeling. Specifically, inspired by [56] which uses a post-hoc strategy to re-calibrate existing DA methods, we theoretically analyze the transferability of adversarial-based DA methods and propose to model the uncertainty variance of a discriminator to minimize biased transfer between domains (see Figure 1(b)). Different from [56], UTEP incorporates unbiased transferability estimation into an online model learning process rather than post-processing, which helps mitigate negative transfer and achieve better alignment between domains for unbiased domain adaptation. Meanwhile, as the estimated uncertainty also reveals the reliability of unlabeled samples from the target domain, we use it for pseudo label selection to achieve better marginal and conditional distribution alignment between source and target domains. The proposed UTEP is plug-and-play and can be easily incorporated into various adversarial based DA methods. An overview of UTEP is depicted in Figure 2. The **contributions** of this work are in three-fold:

1) To address the transferability bias due to the imbalance between the quantity of labeled data from the source and target domains, unlike prior works [46,56] that model the classifier to obtain better calibrated prediction, which still has a strong bias toward source domain, we propose a novel non-intrusive Unbiased Transferability Estimation Plug-in (UTEP) model for DA by uncertainty variance modeling of a discriminator. 2) To quantify and minimize the bias of

transferability, we theoretically analyze the cause of the bias and show that our method can alleviate it by lowering its upper bound. To our best knowledge, this is the first work to explore the unbiased knowledge transfer rather than model calibration via uncertainty modeling in domain adaptation. 3) We plug our technique into various adversarial based DA methods [16,33,66] and show its superiority over the state-of-the-art methods in both UDA and SSDA settings.

2   Related Works

Deep Domain Adaptation. Domain adaptation aims to transfer knowledge between a labeled source domain and an unlabeled or a less labeled target domain. The key challenge of DA is the existence of domain shift [2], the data bias between source and target domains. There have been many distance-based and divergence-based methods proposed in recent years for measuring and resolving the domain shift proposed in DA. These measurement dimensions include Maximum Mean Discrepancy between the feature embeddings of different domains [24,41,68,69], the optimal transmission distance across domains [10], high-dimensional discriminative adversarial learning [16,26,44,54,65], and so on. Although these methods are capable of aligning distribution between domains, they largely ignore the deviation between the estimated transferability and the real one. Our work focuses on learning unbiased transferability by modeling uncertainty variance of a discriminator in adversarial-based domain adaptation.

Uncertainty Estimation. Uncertainty estimation can be used to either measure the uncertainty caused by noise (known as aleatoric uncertainty) or learn the uncertainty of a model (known as epistemic uncertainty) [12,32]. In vision tasks, it is often more challenging and practical to model the epistemic uncertainty, which can be learned by Bayesian neural network [6,42]. Besides, some approximate reasoning methods [48,56] can be used to model the uncertainty based on abundant observation data. Our work employs the MCDropout [15], an efficient method for acquiring Bayesian uncertainty, to model uncertainty variance of a discriminator in adversarial-based domain adaptation.

Uncertainty Modeling in Domain Adaptation. Uncertainty modeling can be used to perform cross-domain calibration and improve the reliability of pseudo labels in DA. CPCS [46] incorporates importance weighting into temperature scaling to cope with the cross-domain calibration. TransCal [56] uses post-hoc transferable calibration to realize more accurate calibration with lower bias and variance. PACET [37] employs uncertainty modeling to describe the cross-domain distribution differences and other intra-domain relationships. BUM [57] uses a Bayesian neural network to quantify prediction uncertainty of a classifier with the aid of a discriminator. CADA [33] utilizes certainty based attention to identify adaptive region on pixel level with Bayesian classifier and discriminator. Besides, uncertainty can also be used to facilitate cross-domain object detection [18] and segmentation [67]. Existing approaches mostly focus more on modeling the classifier to obtain better calibrated prediction. However, due to the extremely unbalanced amount of annotated data in domains, the classifier
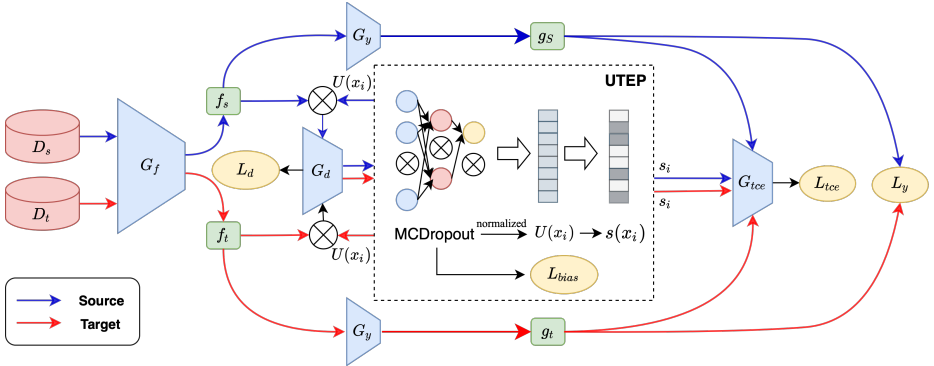
Fig. 2: The proposed UTEP framework. G_f is the feature extractor. G_d is the domain discriminator. G_y is the source classifier. G_{tce} is the pseudo label cluster. f_s and f_t are feature representations defined by $f = G_f(x)$, while g_s and g_t are the classifier prediction obtained from G_y. UTEP is the unbiased transferability estimation plug-in. This module can not only evaluate the unbiased transferability by minimizing $\mathcal{L}_{bias}$ and adding weight $\mu(x)$ on the discriminator with discriminator loss L_d, but also can generate $s(x)$ to evaluate the reliability of pseudo labels to facilitate learning with L_{tce} and L_y.

output is strongly biased to the source, while the discriminator is unbiased. Our work focuses on learning unbiased transferability by uncertainty modeling the discriminator in theory. Table 1 compares representative uncertainty methods.

3   Methodology

3.1   Preliminaries

Problem Definition. Let x_i denote an input of the network, y_i is the corresponding class label of x_i, and d_i is the domain label of x_i. We set $d_i = 1$ when x_i is from the source domain and $d_i = 0$ when x_i is from the target one. In unsupervised domain adaptation (UDA), the labeled source domain is defined as $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s source samples and the unlabeled target domain is defined as $\mathcal{D}_t = \mathcal{D}_{tu} = \{x_i^t\}_{i=1}^{n_{tu}}$ with n_{tu} samples (here, $n_t = n_{tu}$). $\mathcal{D}_s$ and $\mathcal{D}_t$ are sampled from the source $p_s(x)$ and the target $p_t(x)$ distributions. In semi-supervised domain adaptation (SSDA), in addition to the labeled source domain $\mathcal{D}_s$ and the unlabeled target domain $\mathcal{D}_{tu}$, there is an extra target labeled domain $\mathcal{D}_{tl} = \{(x_i^{tl}, y_i^{tl})\}_{i=1}^{n_{tl}}$ with n_{tl} samples, so the whole target domain can be defined as $\mathcal{D}_t = \mathcal{D}_{tl} \cup \mathcal{D}_{tu}$ with $n_t = n_{tl} + n_{tu}$. In our paper, the source label space equals to the target one, there are C categories in both source and target domains.

Classical Adversarial Based UDA. In the classical adversarial based domain adaptation framework, DANN [16], there is a feature extractor and a domain discriminator, where the discriminator tries to find out which domain the features come from while the feature extractor tries to confuse the discriminator. When

the discriminator is fully confused, we consider the source classifier can generalize well to the target domains. In this case, the output of the discriminator should be 0.5, $P(d = 0|x) = P(d = 1|x) = 0.5$. Overall, the source classifier G_y, the domain discriminator G_d and the feature extractor G_f can be jointly learned by:

$$
\min_{G_y} \min_{G_f} \max_{G_d} \mathcal{L}_{adv}(G_d, G_f, G_y) = \mathbb{E}_{(x,y) \sim p_s} [\mathcal{L}_y(G_y(G_f(x)), y)]
$$
$$
- \alpha_{adv} \left(\mathbb{E}_{x \sim p_s}[\log(G_d(G_f(x)))] + \mathbb{E}_{x \sim p_t}[\log(1 - G_d(G_f(x)))] \right), \tag{1}
$$

where $\mathcal{L}_y$ is the cross-entropy loss for source classifier. α_{adv} is a trade-off between classifier loss $\mathcal{L}_y$ and discriminator loss $\mathcal{L}_d$. By default, we set $\alpha_{adv} = 1$.

3.2   Unbiased Transferability Estimation

Transferability in DA. The challenge of DA is to use the source classifier to minimize the target classification error $\mathbb{E}_{(x,y) \sim p_t}[\mathcal{L}_y(G_y(G_f(x)), y)]$ when target labels are unavailable. Since the existence of domain shift, the source classifier cannot work properly on target samples. To better account for the shift between source and target distributions, density ratio $w(x) = \frac{p_t(x)}{p_s(x)}$ [56,62] can be used as a metric of transferability to measure the discrepancy between domains. In general, the shift will be eliminated when $w(x) = \frac{p_t(x)}{p_s(x)} = 1$. Then, the target classification error can be estimated by the source distribution $p_s(x)$ as:

$$
\mathbb{E}_{(x,y) \sim p_t} [\mathcal{L}_y(G_y(G_f(x)), y))] = \int_{D_t} \mathcal{L}_y(G_y(G_f(x)), y) \, p_t(x) dx
$$
$$
= \int_{D_s} \frac{p_t(x)}{p_s(x)} \mathcal{L}_y(G_y(G_f(x)), y) p_s(x) dx = \mathbb{E}_{(x,y) \sim p_s} [w(x)\mathcal{L}_y(G_y(G_f(x)), y))] . \tag{2}
$$

However, the density ratio $w(x)$ is often not accessible in DA, we follow [56] and use estimated $\hat{w}(x)$ to approximate the real $w(x)$. Specifically, LogReg [5,49] is used to estimate the density ratio by Bayesian formula:

$$
\hat{w}(x) = \frac{p_t(x)}{p_s(x)} = \frac{m(x|d=0)}{m(x|d=1)} = \frac{P(d=1)P(d=0|x)}{P(d=0)P(d=1|x)} = \frac{n_s}{n_t} \cdot \frac{P(d=0|x)}{P(d=1|x)} = \frac{P(d=0|x)}{P(d=1|x)}, \tag{3}
$$

where m is a distribution over $(x, d) \sim X \times (0, 1)$, and $d \sim Bernoulli(0.5)$ is a Bernoulli variable representing which domain x belongs to. Here, $\frac{n_s}{n_t}$ is a constant regarding to sample sizes and the source or the target dataset is randomly up-sampled to ensure $n_s = n_t$. In this way, $\hat{w}(x)$ only depends on $\frac{P(d=0|x)}{P(d=1|x)}$.

  In DA, $w(x)$ can be treated as the real transferability, while $\hat{w}(x)$ can be considered as the estimated transferability. Ideally, when $w(x) = \hat{w}(x)$, domain adaptation is unbiased. Previous DA methods [40,68] mainly focus on aligning distributions between source and target domains to encourage $\hat{w}(x) = 1$ but largely ignore the deviation between the estimated $\hat{w}(x)$ and the real $w(x)$, which yields *biased domain adaptation.*
Learning Unbiased Transferability. There have been some DA methods [56,57] employing the classifier output for uncertainty modeling to encourage transferability/-calibration. However, due to the lack of class labels in the target domain, using predictions of unlabeled target samples from the source domain classifier merely describes the alignment of target samples to the source distribution, while not encouraging the source distribution to align to the target one. On the other hand, domain labels are

always known for both source and target samples during model learning, which provides abundant label information for measuring the bias between the estimated $\hat{w}(x)$ and the real $w(x)$. In light of this, to account for the deviation between the estimated $\hat{w}(x)$ and the real $w(x)$, *we propose to model the transferability by uncertainty variance estimation of a discriminator in adversarial-based DA.*

Specifically, different from previous classifier-based uncertainty modeling methods, we introduce a novel unbiased transferability estimation method by lowering the variance of the discriminator output $\mathbb{V}ar_{x \sim p_{ds}}(\hat{P}_d(d = 1|x))$ and $\mathbb{V}ar_{x \sim p_{dt}}(\hat{P}_d(d = 0|x))$. Here, p_{ds} and p_{dt} are the source and target discriminator distributions, an instance is sampled from p_d which equals to be sampled from p_{ds} or p_{dt}. We use MCDropout [15,57] to compute $\mathbb{V}ar_{x \sim p_d}(\hat{P}_d(d|x))$ as:

$$\mathbb{V}ar_{x \sim p_d}(\hat{P}_d(d|x)) \approx \frac{1}{K} \sum_{k=1}^{K} \left((G_d(G_g(x)))_k - \left(\frac{1}{K} \sum_{k=1}^{K} (G_d(G_g(x)))_k \right) \right)^2, \quad (4)$$

where K is the number of times performing stochastic forward passes through the discriminator network. We set $u(x) = \mathbb{V}ar_{x \sim p_{ds}}(\hat{P}_d(d = 1|x)) + \mathbb{V}ar_{x \sim p_{dt}}(\hat{P}_d(d = 0|x))$, the modeled uncertainty. Here, we set $\mathcal{U} = [u(x_1), ...u(x_i), ...]$, where $u(x_i)$ measures the uncertainty of the ith sample. Since the uncertainty can also be seen as a measurement of the distance between x and the general distribution (initial aligned distribution of source and target samples, which be called $p_{s,t}$ in the following). When a sample x is close to $p_{s,t}$, $u(x)$ is minimized and x possesses better transferability. From this perspective, $u(x)$ can also be seen as a transferability metric. We normalize $u(x)$ by Min-Max Normalization as follows:

$$\mu(x) = \frac{u(x) - \min(\mathcal{U})}{\max(\mathcal{U})}, \quad (5)$$

Then, we set normalized unbiased transferability $\mu(x)$ as the transferability weight in the adversarial learning process (Eq. (1)) to facilitate unbiased DA model learning as:

$$\min_{G_y} \min_{G_f} \max_{G_d} \mathcal{L}_{adv}(G_d, G_f, G_y) = \mathbb{E}_{(x,y) \sim p_s}[\mathcal{L}_y(G_y(G_f(x)), y)]$$
$$-\mathbb{E}_{x \sim p_s}[(1 + \mu(x)) \log(G_d(G_f(x)))] - \mathbb{E}_{x \sim p_t}[(1 + \mu(x)) \log((1 - G_d(G_f(x))))]. \quad (6)$$

Furthermore, based on Eq. (4), we define the L2 regularized $\mathcal{U}$ as the bias loss and minimize the bias loss as:

$$\min \mathcal{L}_{bias} = \min ||\mathcal{U}||_2^2 = \min_{G_d, G_f} \sum_{i=1}^{n_t+n_s} \left(\mathbb{V}ar_{x_i \sim p_d}(\hat{P}_d(d|x_i)) \right)^2. \quad (7)$$

With each mini-batch, we perform K times stochastic forward passes through the discriminator to estimate the variance in Eq. (7).

Theoretical Analysis. In this section, we theoretically analyze the cause of the bias in transferability and discuss how to lower the upper bound of the bias. With Eqs. (2) and (3), we show how to measure the target classification error and the estimated transferability. Following [56], we can use the difference between the estimated target classification error and the real one to measure the bias of transferability, as:

$$\left| \mathbb{E}_{(x,y) \sim p_t}[\mathcal{L}_y^{\hat{w}(x)}(G_y(G_f(x)), y))] - \mathbb{E}_{(x,y) \sim p_t}[\mathcal{L}_y^{w(x)}(G_y(G_f(x)), y))] \right|$$
$$= \left| \mathbb{E}_{(x,y) \sim p_s}[(\hat{w}(x) - w(x))\mathcal{L}_y(G_y(G_f(x)), y))] \right| \quad (8)$$
$$\leq \frac{1}{2} \left(\mathbb{E}_{x \sim p_s}[(\hat{w}(x) - w(x))^2] + \mathbb{E}_{(x,y) \sim p_s}[(\mathcal{L}_y(G_y(G_f(x)), y))^2] \right).$$

In the above inequality, since the second term is bounded by supervised learning in the labeled source domain, we only need to focus on the first term. We use a discriminator to alleviate the deviation between the estimated $\hat{w}(x)$ and the real $w(x)$. From our unbiased transferability perspective, we further formalize the transferability based on discriminator as $W(x) = \frac{B_t(x|d=0)}{B_s(x|d=1)}$. Here, distribution B is a distribution over $(x, d) \in X(0, 1)$. In this case, $d \sim Bernoulli(0.5)$, if $d = 1$, $x \sim p_s$ or $x \sim p_t$. Furthermore, as the unbiased transferability is derived in the discriminator label space, $W(x)$ and $\hat{W}(x)$ are assumed to be the real and estimated transferability in this space. Assume we have upper bound $N \geq 0$ for $W(x)$ subject to $N \geq W(x) \geq 0$ according to the bounded importance weight assumption [9]. Combined with upper bound N, we have $\frac{1}{N+1} \leq P_d(d = 1|x) \leq 1$ and $P_d(x) = B(d = 1|x) = \frac{1}{1+W(x)}$. Then, the first term of Eq. (8) is bounded by:

$$\mathbb{E}_{x \sim p_s}\left[(\hat{w}(x_i) - w(x))^2\right] = \mathbb{E}_{x \sim p_{ds}}\left[(\hat{W}(x) - W(x))^2 * \frac{p_s(x)}{p_{ds}(x)}\right]$$

$$\leq 2\mathbb{E}_{x \sim p_{ds}}[(\hat{W}(x_i) - W(x))^2] = 2\mathbb{E}_{x \sim p_{ds}}\left[\left(\frac{P_d(d = 1|x) - \hat{P}_d(d = 1|x)}{P_d(d = 1|x)\hat{P}_d(d = 1|x)}\right)^2\right]$$

$$\leq 2(N + 1)^4 \mathbb{E}_{x \sim p_{ds}}\left[\left(P_d(d = 1|x) - \hat{P}_d(d = 1|x)\right)^2\right]. \tag{9}$$

The first row changes the probability from source label space to source discriminator domain label space. Then, the deviation between the real and the estimated transferability is calculated in the discriminator label space. The second inequality in Eq. (9) can be further rewritten as:

$$2(N + 1)^4 \mathbb{E}_{x \sim p_{ds}}\left[\left(P_d(d = 1|x) - \hat{P}_d(d = 1|x)\right)^2\right]$$

$$= 2(N + 1)^4 \left(\mathbb{E}_{x \sim p_{ds}}[(P_d(d = 1|x))^2] - (\mathbb{E}_{x \sim p_{ds}}[P_d(d = 1|x)])^2\right.$$

$$+ (\mathbb{E}_{x \sim p_{ds}}[P_d(d = 1|x)])^2 + \mathbb{E}_{x \sim p_{ds}}[(\hat{P}_d(d = 1|x))^2]$$

$$- (\mathbb{E}_{x \sim p_{ds}}[\hat{P}_d(d = 1|x)])^2 + (\mathbb{E}_{x \sim p_{ds}}[\hat{P}_d(d = 1|x)])^2$$

$$- 2\mathbb{E}_{x \sim p_{ds}}[\hat{P}_d(d = 1|x)P_d(d = 1|x)]\right)$$

$$= 2(N + 1)^4 \left(\mathbb{V}ar_{x \sim p_{ds}}(P_d(d = 1|x)) + \mathbb{V}ar_{x \sim p_{ds}}(\hat{P}_d(d = 1|x))\right.$$

$$\left. + \left(\mathbb{E}_{x \sim p_{ds}}[P_d(d = 1|x)] - \mathbb{E}_{x \sim p_{ds}}[\hat{P}_d(d = 1|x)]\right)^2\right), \tag{10}$$

where the variances of outputs under real and estimated probability distributions are $\mathbb{V}ar_{x \sim p_{ds}}(P_d(d = 1|x))$ and $\mathbb{V}ar_{x \sim p_{ds}}(\hat{P}_d(d = 1|x))$ respectively. Ideally, when the source and target domains are aligned, the output of discriminator should be 0.5, then, $\mathbb{V}ar_{x \sim p_{ds}}(P_d(d = 1|x)) = 0$. Hence, the bias of transferability can be formulated as:

$$\mathbb{E}_{x \sim p_{ds}}\left[(\hat{w}(x) - w(x))^2\right] \leq 2(N + 1)^4 \left[\mathbb{V}ar_{x \sim p_{ds}}\left(\hat{P}_d(d = 1|x)\right)\right.$$

$$\left. + \left(\mathbb{E}_{x \sim p_{ds}}[P_d(d = 1|x)] - \mathbb{E}_{x \sim p_{ds}}[\hat{P}_d(d = 1|x)]\right)^2\right]. \tag{11}$$

The second term of Eq. (11) is constrained since the domain adaptation process encourages $\mathbb{E}_{x \sim p_{ds}}(P_d(d = 1|x))$ to approximate to $\mathbb{E}_{x \sim p_{ds}}(\hat{P}_d(d = 1|x))$. Therefore, to

learn unbiased transferability, we can minimize $\mathbb{V}ar_{x \sim p_{ds}}(\hat{P}_d(d = 1|x))$. Besides, since we need to use the estimated uncertainty of unlabeled samples from the target domain for pseudo label selection (see Sec. 3.3), we also minimize $\mathbb{V}ar_{x \sim p_{dt}}(\hat{P}_d(d = 0|x))$ and use it as a part of the transferability weight. Thus, we set:

$$u(x) = \mathbb{V}ar_{x \sim p_{ds}}(\hat{P}_d(d = 1|x)) + \mathbb{V}ar_{x \sim p_{dt}}(\hat{P}_d(d = 0|x)). \tag{12}$$

In this way, for both the source and target samples, we lower the variance of the discriminator outputs with Eq. (4), and use Eq. (7) to lower the upper bound of the deviation between estimated transferability and the real one to realize unbiased domain adaptation. The details of the theoretical analysis are in the supplementary materials.

3.3   Unbiased Domain Adaptation

Pseudo Label Selection. Originally, pseudo label is introduced to solve the problem of label shortage in unlabeled domain in semi-supervised learning. However, recent works [7,38] implies that when domain shift exists, the pseudo label selection strategies tailored for semi-supervised learning is difficult to be effective. Different from the classifier-based pseudo label evaluation methods [14,50,60], we evaluate pseudo label reliability under domain shift by the normalized unbiased transferability $\mu(x)$ of domain discriminator. Intuitively, the lower $\mu(x)$ indicates the better transferability of x, which is more reliable for pseudo labeling. Denote $s(x)$ as the selected weight for x:

$$s(x) = 1 - \mu(x). \tag{13}$$

Then, pseudo labels are generated with those preliminary refined unlabeled samples of the target domain based on the predefined thresholds. Specifically, suppose $g(x)$ is the C-ways source classifier probability prediction output for sample x, $g(x) = G_y(G_f(x)) = [g^{[1]}(x), ...g^{[c]}(x), ..., g^{[C]}(x)]$. Here, $g^{[c]}(x)$ is the probability of class c for the sample. Only when the $g^{[c]}(x)$ is higher than the threshold β, where $\beta \in (0, 1)$, the positive pseudo label for the sample is selected. Hence, $h(x) = [h^{[1]}(x), ..., h^{[C]}(x)]$ is a binary vector representing the selected positive pseudo label for the sample. When $g^{[c]}(x)$ is selected, $h^{[c]}(x)$ is 1, or $h^{[c]}(x)$ is 0. Here, $h^{[c]}(x)$ is obtained by:

$$h^{[c]}(x) = \mathbb{1}[g^{[c]}(x) \geq \beta]. \tag{14}$$

Then, the selected $g^{[c]}(x)$ is treated as a soft positive pseudo label, so the positive cross-entropy loss is defined as:

$$\mathcal{L}_{pce}(g, h) = -\mathbb{E}_{x \sim p_d} s(x) \sum_{c=1}^{C} h^{[c]}(x) \left[g^{[c]}(x) \cdot \log(g^{[c]}(x)) \right]. \tag{15}$$

Similarly, we pick out those unlikely categories with high probability as negative pseudo labels to further dismiss the interference of noises on training. Only when $g^{[c]}(x) \leq \gamma$, where $\gamma \in (0, 1)$, the negative pseudo label is generated for the sample. Thus, we have:

$$l^{[c]}(x) = \mathbb{1}[g^{[c]}(x) \leq \gamma], \tag{16}$$

where $l(x)$ is corresponding to $h(x)$, representing the selected negative pseudo label for the sample. Then, the selected $g^{[c]}(x)$ is seen as a soft negative pseudo-label, so a negative cross-entropy loss is defined as:

$$\mathcal{L}_{nce}(g, l) = -\mathbb{E}_{x \sim p_d} s(x) \sum_{c=1}^{C} l^{[c]}(x) \left[(1 - g^{[c]}(x)) \cdot \log(1 - g^{[c]}(x)) \right]. \tag{17}$$

Thus, the pseudo label cluster learning loss is modeled as:

$$\mathcal{L}_{tce}(g, h, l) = \mathcal{L}_{pce}(g, h) + \alpha_{nce}\mathcal{L}_{nce}(g, l). \tag{18}$$

By default, $\alpha_{nce} = 1$. Note that this strategy is similar to UPS [50] which presents uncertainty-aware pseudo labeling for semi-supervised learning, but we conjecture that the uncertainty of labels mostly comes from the domain gap between labeled and unlabeled data. Hence, unlike UPS, uncertainty is derived by a variance of the discriminator output and used for pseudo label selection to facilitate unbiased domain adaptation. Such a strategy is not only suitable for DA and SSDA, but also for semi-supervised learning problems. Furthermore, we conducted a comparative experiment in Table 5 to show the effectiveness of our uncertainty-aware pseudo labeling.

Unbiased Domain Adaptation. The proposed UTEP for DA includes three parts, namely unbiased domain alignment with adversarial learning, transferability bias regularization, and pseudo label cluster learning. The overall loss of UTEP is defined as:

$$\mathcal{L} = \mathcal{L}_{adv} + \alpha_{bias}\mathcal{L}_{bias} + \alpha_{tce}\mathcal{L}_{tce}, \tag{19}$$

where $\mathcal{L}_{adv}$ focuses on domain alignment with the adversarial loss and the source classification loss. $\mathcal{L}_{bias}$ is the unbiased transferability estimation loss which is obtained per batch during model training. $\mathcal{L}_{tce}$ is the pseudo label cluster loss. α_{bias} and α_{tce} are the hyper-parameters.

4 Experiments

4.1 Datasets and Protocols

Datasets. To evaluate the effectiveness of the proposed UTEP approach, we conduct extensive experiments on three popular DA datasets: **Office-31**, **Office-Home** and **VisDA-2017**. **Office-31** is the most popular DA dataset, containing 31 classes and 4600 images from three domains, namely Webcam(**W**), Amazon(**A**) and Dslr(**D**). **Office-Home** includes 15,500 images from 65 categories collected from four domains, namely Artistic Images (**Ar**), Clip Art (**Cl**), Product (**Pr**) and Real-World (**Rw**). **VisDA-2017** is a more challenging **S**imulation-to-**R**eal dataset with more than 280K images in 12 categories.

Protocols. Our experiments are performed under two different settings, namely UDA and SSDA, and are carried out using PyTorch. Our codes are based on [30] and released on Github. We use mini-batch SGD to fine-tune an ImageNet pretrained model (ResNet-50 [20] and ResNet-101 for different DA tasks) as the feature encoder with the learning rate as 0.001 and to learn the new layers (bottleneck layer and classification layer) from scratch with the learning rate is 0.01. In the UDA setting, each training batch consists of 32 source samples and 32 target samples. In the SSDA setting, each training batch consists of 16 source samples, 16 labeled target samples and 32 unlabeled target samples. We conduct the experiment with ResNet-34 as the feature encoder. It is worth noting that only 1% target samples are selected out as the labeled target domain for training. More implementation details are in the supplementary material.

4.2 Experimental Results

Unsupervised DA. To verify the universality of our UTEP, we incorporate it into various adversarial-based UDA methods, including three classical methods (DANN [16],

Table 2: Classification accuracy of UDA on *Office-31* with ResNet-50 as backbone model. Best in **bold** and the second best in **bold with underline**.

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
MinEnt [17]	89.4	97.5	**100.0**	90.7	67.1	65.0	85.0
ResNet [20]	75.8	95.5	99.0	79.3	63.6	63.8	79.5
GTA [16]	89.5	97.9	99.8	87.7	72.8	71.4	86.5
CDAN+E [39]	94.2	98.6	**100.0**	94.5	72.8	72.2	88.7
SAFN [59]	90.1	98.6	99.8	90.7	73.0	70.2	87.1
CAN [31]	94.5	99.1	99.8	95.0	**78.0**	**77.0**	**90.6**
MCC [29]	94.0	98.5	**100.0**	92.1	74.9	75.3	89.1
BNM [11]	94.0	98.5	**100.0**	92.2	74.9	75.3	89.2
GSDA [25]	95.7	99.1	**100.0**	94.8	73.5	74.9	89.7
SRDC [53]	95.7	**99.2**	100.0	**95.8**	76.7	**77.1**	90.8
VAK [22]	91.8	98.7	**99.9**	89.9	73.9	72.0	87.7
DANN [16]	91.4	97.9	100.0	83.6	73.3	70.4	86.1
DANN+UTEP	92.7	98.5	100.0	90.4	73.8	72.7	88.0(+1.9%)
MDD [66]	94.5	98.4	100.0	93.4	74.6	72.2	88.9
MDD+UTEP	94.7	99.0	100.0	94.4	77.0	74.5	89.9(+1.0%)
CADA [33]	**97.0**	99.3	100.0	**95.6**	71.5	73.0	89.5
CADA+UTEP	**97.2**	**99.4**	100.0	95.3	73.5	74.3	90.0(+0.5%)
TransPar [19]	95.5	98.9	100.0	94.2	77.7	72.8	89.9
TransPar+UTEP	95.7	**99.4**	100.0	95.2	**78.6**	75.6	**90.8**(+0.9%)

Table 3: Accuracy of UDA on *Office-Home* with ResNet-50.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
MinEnt [17]	51.0	71.9	77.1	61.2	69.1	70.1	59.3	48.7	77.0	70.4	53.0	81.0	65.8
ResNet [20]	41.1	65.9	73.7	53.1	60.1	63.3	52.2	36.7	71.8	64.8	42.6	75.2	58.4
CDAN+E [39]	54.6	74.1	78.1	63.0	72.2	74.1	61.6	52.3	79.1	72.3	57.3	82.8	68.5
DAN [63]	45.6	67.7	73.9	57.7	63.8	66.0	54.9	40.0	74.5	66.2	49.1	77.9	61.4
SAFN [59]	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
TransCal [56]	49.4	68.4	75.5	57.6	70.1	70.4	51.1	50.3	72.4	68.9	54.4	81.2	64.1
ATM [36]	52.4	72.6	78.0	61.1	72.0	72.6	59.5	52.0	79.1	73.3	58.9	83.4	67.9
CKD [43]	54.2	74.1	77.5	64.6	72.2	71.0	64.5	53.4	78.7	72.6	58.4	82.8	68.7
GSDA [27]	**61.3**	76.1	79.4	65.4	73.3	74.3	65.0	53.2	80.0	72.2	60.6	83.1	70.3
SRDC [53]	52.3	76.3	**81.0**	**69.5**	76.2	78.0	**68.7**	53.8	**81.7**	76.3	57.1	**85.0**	**71.3**
DCC [35]	**58.0**	54.1	58.0	**74.6**	70.6	**77.5**	64.3	**73.6**	74.9	80.9	**75.1**	80.4	70.2
DANN [16]	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+UTEP	50.7	68.9	77.1	58.7	72.3	71.7	59.3	52.9	79.8	73.5	59.9	83.6	67.3(+2.1%)
MDD [66]	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
MDD+UTEP	57.2	75.9	79.6	63.4	72.8	73.7	64.6	**55.4**	79.8	74.0	61.1	84.2	70.1(+2.0%)
CADA [33]	56.9	**76.4**	80.7	61.3	75.2	75.2	63.2	54.5	80.7	73.9	61.5	84.1	70.2
CADA+UTEP	57.1	**76.5**	81.1	61.7	**75.4**	75.2	63.9	54.9	**80.9**	74.2	**61.8**	84.1	**70.6**(+0.4%)
TransPar [19]	55.3	75.9	79.2	63.2	72.4	71.8	61.2	55.1	80.0	74.5	60.9	83.7	69.8
TransPar+UTEP	57.4	76.1	80.2	64.2	73.2	73.7	**64.8**	**55.4**	80.9	**74.7**	61.1	**84.6**	**70.6**(+0.8%)

MDD [66] and CADA [33]) and a state-of-the-art method (TransPar [19]). Tables 2, 3 and 4 show experimental results on the small-sized Office-31, the medium-sized Office-Home and the more challenging VisDA-2017, respectively. Overall, our UTEP module can improve the performance of various adversarial-based baseline methods, achieving state-of-the-art performance among UDA methods. On Office-31, for average accuracy, UTEP improves DANN (by 1.9%), MDD (by 1.0%), CADA (by 0.5%) and TransPar (by 0.9%) and TransPar+UTEP achieves 90.8% which is on par with the state-of-the-art. On Office-Home, in terms of average accuracy, UTEP significantly improves DANN, MDD and TransPar by approximately 2.0% and improves CADA by 0.4%, while CADA+UTEP and TransPar+UTEP achieve 70.6% which are the second best

Table 4: Classification accuracy of UDA on *VisDA-2107* using ResNet-101.

Method	plane	bicycle	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Mean
MinEnt [17]	88.6	29.5	82.5	75.8	88.7	16.0	93.2	63.4	94.2	40.1	87.3	12.1	64.3
ResNet [20]	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
SAFN [59]	94.2	56.2	81.3	**69.8**	93.0	81.0	**93.0**	74.1	91.7	55.0	**90.6**	18.1	75.0
MixMatch [4]	93.9	71.8	93.5	**82.1**	**95.3**	0.7	90.8	38.1	94.2	96.0	86.3	2.2	70.4
BNM [11]	91.1	69.0	76.7	64.3	89.8	61.2	90.8	74.8	90.9	66.6	88.1	46.1	75.8
MCC [29]	92.2	**82.9**	76.8	66.6	90.9	78.5	87.9	73.8	90.1	76.1	87.1	41.0	78.7
DWL [58]	90.7	**80.2**	86.1	67.6	92.4	81.5	86.8	78.0	90.6	57.1	85.6	28.7	77.1
VAK [22]	**94.3**	79.0	**84.9**	63.6	92.6	92.0	88.4	79.1	92.2	79.8	87.6	43.0	81.4
DANN [16]	90.0	58.9	76.9	56.1	80.3	60.9	89.1	72.5	84.3	73.8	**89.3**	35.8	72.3
DANN+UTEP	93.8	70.7	**85.5**	62.7	91.0	90.7	88.6	76.2	87.2	85.4	84.7	33.8	79.2(**+6.9%**)
MDD [66]	94.2	71.6	84.3	65.4	91.5	94.9	92.0	**80.3**	90.8	**90.0**	82.4	42.0	81.4
MDD+UTEP	**94.7**	75.4	83.2	60.1	**93.7**	95.3	**93.1**	82.6	**94.3**	**89.8**	84.6	41.1	**82.3(+0.9%)**
CADA [33]	85.1	61.2	84.4	69.0	89.8	**97.1**	93.0	75.8	90.2	84.1	81.4	45.4	79.7
CADA+UTEP	88.9	76.9	82.6	65.6	90.8	**96.9**	90.6	78.5	86.3	86.4	83.8	**48.8**	81.3(**+1.6%**)
TransPar [19]	84.2	57.7	85.3	62.7	90.0	92.9	92.3	73.9	**95.9**	86.9	84.2	44.7	79.4
TransPar+UTEP	90.0	74.8	82.6	66.2	91.1	95.8	91.3	77.5	89.0	88.3	82.6	**47.2**	**81.5**(**+2.1%**)

Table 5: Classification accuracy of SSDA with 1% target labeled data on *Office-Home* with ResNet-34 as backbone model (left) and accuracy of 3-shot SSL on *Office-Home* with ResNet-50 as backbone (right).

Methods	SSDA setting				Methods	3-shot SSL setting				
	Ar→Rw	Rw→Pr	Pr→Cl	Avg		Ar	Cl	Pr	Rw	Avg
ADR [64]	70.6	76.6	49.5	65.6	ResNet [20]	48.7	42.1	68.9	66.6	56.6
IRM [1]	71.1	77.6	51.5	66.7	MixMatch [4]	52.2	41.9	73.1	69.1	59.1
MME [52]	72.1	78.1	52.8	67.7	MinEnt [17]	51.7	44.5	72.4	68.9	59.4
CDAN [39]	73.0	79.2	53.1	68.4	MCC [29]	58.9	**47.7**	77.4	74.3	**64.6**
LIRR [34]	**73.6**	**80.2**	**53.8**	**69.2**	BNM [11]	59.0	**46.0**	**76.5**	71.5	63.2
DANN [16]	72.2	78.1	52.5	67.6	DANN [16]	57.4	41.6	74.5	73.4	61.7
DANN+UPS [50]	73.2	79.6	52.8	68.5	DANN+UPS [50]	**59.1**	42.6	75.3	**76.2**	63.3
DANN+UTEP	**74.4**	**82.1**	**53.3**	**69.9 (+2.3%)**	DANN+UTEP	**60.2**	43.6	**77.9**	**78.8**	**65.1(+3.4%)**

results. On VisDA-2017, we can see a notable improvement of UTEP to DANN (by 6.9%) in average accuracy while the improvement of UTEP to MDD, CADA and TransPar are also significant, yielding state-of-the-art performance. In our analysis, the improvement of UTEP to various adversarial-based DA methods can be attributed to learning unbiased transferability by uncertainty variance modeling and uncertainty-based pseudo label selection.

Semi-supervised DA. On SSDA, we follow the setting of the state-of-the-art LIRR [34] and conduct experiments on Office-Home. We randomly select 1% of target samples as the labeled target domain for training and evaluate SSDA methods on unlabeled samples from the target domain. As shown in the left of Table 5, our method significantly improves the mean accuracy of DANN (by 2.3%) and achieves compelling performance against the state-of-the-art SSDA methods. This also examines the efficacy of UTEP for DA under the semi-supervised learning scenario.

4.3   Further Analysis and Discussion

Visualization. In Figure 3(a)-(d), we use t-SNE [13] to visualize feature embeddings of vanilla ResNet, DANN, DANN+UPS and DANN+UTEP on task **Pr→Ar** (65 classes) on Office-Home. From Figures 3(a) to 3(c), we can see that the vanilla ResNet has
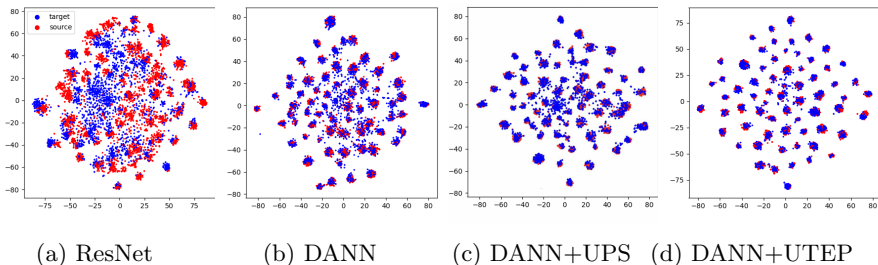
(a) ResNet   (b) DANN   (c) DANN+UPS (d) DANN+UTEP

Fig. 3: Visualization results for baseline and ours of UDA w/ ResNet-50.

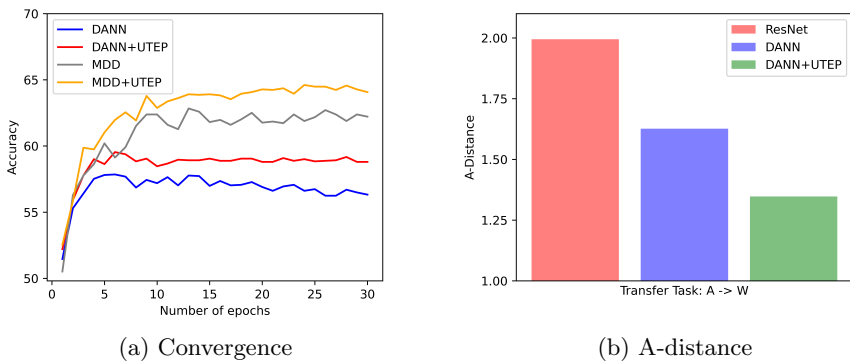(a) Convergence   (b) A-distance

Fig. 4: (a) Comparison of the training convergence between the two baseline UDA methods(DANN and MDD) and variants equipped with our UTEP with ResNet-50 as backbone; (b) Comparison of the distribution discrepancy measured by A-distance between vanilla ResNet-50, DANN and DANN with our UTEP.

large domain discrepancy, while DANN significantly decreases the domain discrepancy. However, the decision boundaries are still not clear in DANN due to the misalignment between the source and target distributions, while DANN+UPS can make the target decision boundary much clearer, but there are still many samples near the boundary are hard to distinguish. By contrast, as shown in Figure 3(d), DANN+UTEP further resolves this issue and achieves both better marginal and conditional distribution alignments thanks to the effective learning of unbiased transferability.

Convergence and Distribution Discrepancy. In Figure 4(a), we compare the convergence of DANN, DANN+UTEP, MDD and MDD+UTEP on task $\mathbf{Pr}{\rightarrow}\mathbf{Ar}$ on Office-Home. We can see that our UTEP can significantly improve the accuracy of DANN and MDD while achieving more stable convergence. Figure 4(b) compares A-distance, a widely used measure for distribution discrepancy in DA [3], of vanilla ResNet, DANN and DANN+UTEP on task $\mathbf{Pr}{\rightarrow}\mathbf{Ar}$ on Office-Home. It is clearly shown that A-distance of DANN+UTEP on task $\mathbf{A}{\rightarrow}\mathbf{W}$ is smaller than that of vanilla ResNet and DANN, indicating better distribution alignment and higher accuracies of DANN+UTEP.

Table 6: Ablation study of variants with baseline DANN [16]. SIW and TIW denote using importance weights for samples from source and target domains respectively (Eq. (6)); SBL and TBL represent using the bias loss for samples from source and target domains (Eq. (7)); PCE and NCE indicate using the positive and negative cross-entropy in pseudo label cluster learning (Eq. (18)).

DANN [16]'s variant						settings on Office-31						
SIW	TIW	SBL	TBL	PCE	NCE	A→W	D→W	W→D	A→D	D→A	W→A	Avg
✗	✗	✗	✗	✗	✗	91.4	97.9	**100.0**	83.6	73.3	70.4	86.1
✓	✓	✓	✓	✗	✗	92.0	98.2	**100.0**	85.7	73.0	71.1	86.7
✓	✓	✗	✗	✓	✓	92.2	98.2	**100.0**	88.5	72.9	71.5	87.2
✓	✓	✓	✓	✓	✗	92.3	98.3	**100.0**	87.9	73.3	72.0	87.3
✓	✗	✓	✗	✓	✓	91.8	98.2	**100.0**	88.2	73.5	70.9	87.1
✓	✓	✓	✗	✓	✓	92.0	98.2	**100.0**	89.0	73.3	71.5	87.3
✓	✓	✓	✓	✓	✓	**92.7**	**98.5**	**100.0**	**90.4**	**73.8**	**72.7**	**88.0**

Component Effectiveness Evaluation. In Table 6, we study the effectiveness of each component in the proposed UTEP using DANN as the baseline method. In Table 6, the results in the first and seventh rows refer to DANN and DANN+UTEP, respectively. It is shown that all the variants perform better than DANN only, showing the effectiveness of each component in the proposed UTEP. The results in the second row are those of DANN using uncertainty modeling for sample weighting and bias loss regularization, which are superior to those of DANN only but inferior to those of DANN+UTEP. The results in the third row shows that without the bias loss, the performance of DANN+UTEP decreases. The results in the fourth, fifth and sixth rows further show that using only parts of the proposed components will also lead to performance degradation.

Evaluation on Semi-Supervised Learning. To further study the efficacy of the proposed UTEP, especially compared with semi-supervised pseudo labeling methods (such as UPS [50]), we conduct 3-shot semi-supervised learning (SSL) experiments on Office-Home in the right part of Table 5. In the SSL setting, only three samples of each class are selected as the labeled domain while the rest are used as the unlabeled domain. For fair comparison between UTEP and UPS, we incorporate both UTEP and UPS into DANN for experiments. From the right part of Table 5, we can see that both UTEP and UPS can improve the performance of DANN, which shows the effectiveness of uncertainty-aware pseudo label selection. Here, DANN+UTEP outperforms DANN+UPS, which further shows the effectiveness of learning unbiased transferability between labeled and unlabeled data for SSL.

5   Conclusion

In this work, we present the Unbiased Transferability Estimation Plug-in (UTEP) to learn unbiased transferability in domain adaptation. The key idea is to model the variance uncertainty of a discriminator in adversarial-based DA method and further exploit the uncertainty for pseudo label selection to achieve better marginal and conditional distribution alignment. Experiments on DA benchmarks show its effectiveness for improving various DA methods with state-of-the-art performance.

References

1. Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant risk minimization. arXiv preprint arXiv:1907.02893 (2019) 12
2. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine learning (2010) 2, 4
3. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. Advances in neural information processing systems **19** (2006) 13
4. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: A holistic approach to semi-supervised learning. Advances in Neural Information Processing Systems **32** (2019) 12
5. Bickel, S., Scheffer, T.: Dirichlet-enhanced spam filtering based on biased samples. In: NIPS (2007) 6
6. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural network. In: ICML (2015) 4
7. Chen, C., Xie, W., Huang, W., Rong, Y., Ding, X., Huang, Y., Xu, T., Huang, J.: Progressive feature alignment for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 627–636 (2019) 9
8. Chen, X., Wang, S., Long, M., Wang, J.: Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In: International conference on machine learning. pp. 1081–1090. PMLR (2019) 2
9. Cortes, C., Mansour, Y., Mohri, M.: Learning bounds for importance weighting. In: NIPS (2010) 8
10. Courty, N., Flamary, R., Tuia, D., Rakotomamonjy, A.: Optimal transport for domain adaptation. IEEE transactions on pattern analysis and machine intelligence **39**(9), 1853–1865 (2016) 4
11. Cui, S., Wang, S., Zhuo, J., Li, L., Huang, Q., Tian, Q.: Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In: CVPR (2020) 11, 12
12. Der Kiureghian, A., Ditlevsen, O.: Aleatory or epistemic? does it matter? Structural safety **31**(2), 105–112 (2009) 4
13. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. In: International conference on machine learning. pp. 647–655. PMLR (2014) 12
14. French, G., Mackiewicz, M., Fisher, M.: Self-ensembling for visual domain adaptation. In: International Conference on Learning Representations (ICLR) (2018) 9
15. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: ICML (2016) 4, 7
16. Ganin, Y., Ustinova, E., Ajakan, H.., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.S.: Domain-adversarial training of neural networks. Journal of Machine Learning Research (2016) 2, 4, 5, 10, 11, 12, 14

17. Grandvalet, Y., Bengio, Y., et al.: Semi-supervised learning by entropy minimization. CAP **367**, 281–296 (2005) 11, 12
18. Guan, D., Huang, J., Xiao, A., Lu, S., Cao, Y.: Uncertainty-aware unsupervised domain adaptation in object detection. IEEE Transactions on Multimedia (2021) 4
19. Han, Z., Sun, H., Yin, Y.: Learning transferable parameters for unsupervised domain adaptation. arXiv preprint arXiv:2108.06129 (2021) 11, 12
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 10, 11, 12
21. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: International conference on machine learning. pp. 1989–1998. PMLR (2018) 2
22. Hou, Y., Zheng, L.: Visualizing adapted knowledge in domain transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13824–13833 (June 2021) 11, 12
23. Hu, J., Tuo, H., Wang, C., Qiao, L., Zhong, H., Jing, Z.: Multi-weight partial domain adaptation. In: BMVC (2019) 2
24. Hu, J., Tuo, H., Wang, C., Qiao, L., Zhong, H., Yan, J., Jing, Z., Leung, H.: Discriminative partial domain adversarial network. In: ECCV. pp. 632–648. Springer (2020) 4
25. Hu, J., Tuo, H., Wang, C., Zhong, H., Pan, H., Jing, Z.: Unsupervised satellite image classification based on partial transfer learning. Aerospace Systems **3**(1), 21–28 (2020) 11
26. Hu, J., Tuo, H., Zhang, S., Wang, C., Zhong, H., Zou, Z., Jing, Z., Leung, H., Zou, R.: Self-adaptive partial domain adaptation. arXiv preprint arXiv:2109.08829 (2021) 4
27. Hu, L., Kan, M., Shan, S., Chen, X.: Unsupervised domain adaptation with hierarchical gradient synchronization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4043–4052 (2020) 11
28. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv preprint arXiv:1602.07360 (2016) 1
29. Jin, Y., Wang, X., Long, M., Wang, J.: Minimum class confusion for versatile domain adaptation. In: ECCV (2020) 11, 12
30. Junguang Jiang, Baixu Chen, B.F.M.L.: Transfer-learning-library. https://github.com/thuml/Transfer-Learning-Library (2020) 10
31. Kang, G., Jiang, L., Yang, Y., Hauptmann, A.G.: Contrastive adaptation network for unsupervised domain adaptation. In: CVPR. pp. 4893–4902 (2019) 11
32. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? arXiv preprint arXiv:1703.04977 (2017) 4
33. Kurmi, V.K., Kumar, S., Namboodiri, V.P.: Attending to discriminative certainty for domain adaptation. In: CVPR. pp. 491–500 (2019) 3, 4, 11, 12
34. Li, B., Wang, Y., Zhang, S., Li, D., Keutzer, K., Darrell, T., Zhao, H.: Learning invariant representations and risks for semi-supervised domain adaptation. In: CVPR (2021) 12
35. Li, G., Kang, G., Zhu, Y., Wei, Y., Yang, Y.: Domain consensus clustering for universal domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9757–9766 (June 2021) 11

36. Li, J., Chen, E., Ding, Z., Zhu, L., Shen, H.T.: Maximum density divergence for domain adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020) 11
37. Liang, J., He, R., Sun, Z., Tan, T.: Exploring uncertainty in pseudo-label guided unsupervised domain adaptation. Pattern Recognition (2019) 3, 4
38. Liu, H., Wang, J., Long, M.: Cycle self-training for domain adaptation. arXiv preprint arXiv:2103.03571 (2021) 9
39. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: NeurIPS (2018) 2, 3, 11, 12
40. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: ICML (2015) 2, 6
41. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. arXiv preprint arXiv:1602.04433 (2016) 2, 4
42. Louizos, C., Welling, M.: Multiplicative normalizing flows for variational bayesian neural networks. In: ICML (2017) 4
43. Luo, Y.W., Ren, C.X.: Conditional bures metric for domain adaptation. In: CVPR (2021) 11
44. Luo, Z., Zou, Y., Hoffman, J., Fei-Fei, L.: Label efficient learning of transferable representations across domains and tasks. arXiv preprint arXiv:1712.00123 (2017) 4
45. Pan, Y., Yao, T., Li, Y., Wang, Y., Ngo, C.W., Mei, T.: Transferrable prototypical networks for unsupervised domain adaptation. In: CVPR (2019) 2
46. Park, S., Bastani, O., Weimer, J., Lee, I.: Calibrated prediction with covariate shift via unsupervised domain adaptation. In: AISTATS (2020) 3, 4
47. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., Tran, D.: Image transformer. In: International Conference on Machine Learning. pp. 4055–4064. PMLR (2018) 1
48. Pawlowski, N., Brock, A., Lee, M.C.H., Rajchl, M., Glocker, B.: Implicit weight uncertainty in neural networks. arXiv preprint arXiv:1711.01297 (2017) 4
49. Qin, J.: Inferences for case-control and semiparametric two-sample density ratio models. Biometrika **85**(3), 619–630 (1998) 6
50. Rizve, M.N., Duarte, K., Rawat, Y.S., Shah, M.: In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In: ICLR (2020) 9, 10, 12, 14
51. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV (2010) 2
52. Saito, K., Kim, D., Sclaroff, S., Darrell, T., Saenko, K.: Semi-supervised domain adaptation via minimax entropy. In: ICCV (2019) 12
53. Tang, H., Chen, K., Jia, K.: Unsupervised domain adaptation via structurally regularized deep clustering. In: CVPR. pp. 8725–8735 (2020) 11
54. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR (2017) 2, 4
55. Wang, X., Li, L., Ye, W., Long, M., Wang, J.: Transferable attention for domain adaptation. In: AAAI (2019) 2
56. Wang, X., Long, M., Wang, J., Jordan, M.: Transferable calibration with lower bias and variance in domain adaptation. Advances in Neural Information Processing Systems pp. 19212–19223 (2020) 3, 4, 6, 7, 11
57. Wen, J., Zheng, N., Yuan, J., Gong, Z., Chen, C.: Bayesian uncertainty matching for unsupervised domain adaptation. arXiv preprint arXiv:1906.09693 (2019) 3, 4, 6, 7

58. Xiao, N., Zhang, L.: Dynamic weighted learning for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15242–15251 (June 2021) 12
59. Xu, R., Li, G., Yang, J., Lin, L.: Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In: ICCV. pp. 1426–1435 (2019) 11, 12
60. Yang, X., Hou, L., Zhou, Y., Wang, W., Yan, J.: Dense label encoding for boundary discontinuity free rotation detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15819–15829 (2021) 9
61. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? arXiv preprint arXiv:1411.1792 (2014) 2
62. You, K., Wang, X., Long, M., Jordan, M.: Towards accurate model selection in deep unsupervised domain adaptation. In: International Conference on Machine Learning. pp. 7124–7133. PMLR (2019) 6
63. Zhang, J., Ding, Z., Li, W., Ogunbona, P.: Importance weighted adversarial nets for partial domain adaptation. In: CVPR (2018) 11
64. Zhang, S., Wu, G., Costeira, J.P., Moura, J.M.: Understanding traffic density from large-scale web camera data. In: CVPR (2017) 12
65. Zhang, S., Tuo, H., Hu, J., Jing, Z.: Domain adaptive yolo for one-stage cross-domain detection. arXiv preprint arXiv:2106.13939 (2021) 4
66. Zhang, Y., Liu, T., Long, M., Jordan, M.: Bridging theory and algorithm for domain adaptation. In: International Conference on Machine Learning. pp. 7404–7413. PMLR (2019) 4, 11, 12
67. Zheng, Z., Yang, Y.: Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. International Journal of Computer Vision (2021) 4
68. Zhong, H., Tuo, H., Wang, C., Ren, X., Hu, J., Qiao, L.: Source-constraint adversarial domain adaptation. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 2486–2490. IEEE (2019) 2, 4, 6
69. Zou, Z., Qu, X., Zhou, P., Xu, S., Ye, J.: Coarse to fine: Domain adaptive crowd counting via adversarial scoring network (2021) 4