This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Stochastic Binary Network for Universal Domain Adaptation

Saurabh Kumar Jain Sukhendu Das
Visualization and Perception Lab, Department of Computer Science
Engineering, Indian Institute of Technology, Madras, India
cs21s043@cse.iitm.ac.in, sdas@iitm.ac.in

Abstract

*Universal domain adaptation (UniDA) is the unsupervised domain adaptation with label shift. UniDA aims to classify unlabeled target samples into one of the "known" categories or into a single "unknown" category. Its main challenge lies in detecting private classes from both domains and performing alignment between the common classes. Current methods employ various techniques and loss functions to address these challenges. However, these methods commonly represent classifiers as point weight vectors, which are prone to overfitting by the source domain samples due to the lack of supervision from the target domain. Consequently, these classifiers struggle to separate target samples into known and unknown categories effectively. To address this, we introduce a novel framework called **St**ochastic Binary Network for **Un**iversal Domain Adaptation (STUN). STUN uses a Stochastic binary classifier for each class, whose weight is modeled as Gaussian distribution, enabling to sample an arbitrary number of classifiers while keeping the model size same as of two classifiers. Consistency between these sampled classifiers is used to derive the confidence scores for both source and target samples, which facilitates the alignment of common classes using weighted adversarial learning. Finally, we use deep discriminative clustering to formulate a loss function for solving the problem of fragmented feature distributions in the target domain. Extensive ablation studies and state-of-the-art results across three standard benchmark datasets show the efficacy of our framework.*

1. Introduction

Unsupervised domain adaptation (UDA) [45] mitigates domain shifts by transferring the knowledge from labeled source data ($i.e.$ source domain) to unlabeled target data ($i.e.$ target domain). Nevertheless, conventional UDA techniques [13,25,37] excel only when the source and target domains share the same label space. Recognizing this limitation, recent works introduced domain adaptation strategies that consider both domain and category shifts. Open-set domain adaptation (ODA) [29] assumes the presence of target-private classes, while Partial domain adaptation (PDA) [4] deals with the situation of source-private classes. However, these approaches are misaligned with real-world challenges, where the differences in label spaces between domains are unknown in advance. To address this, Universal Domain Adaptation (UniDA) [51] has emerged, aiming to accommodate various category shift scenarios and to classify target samples into either one of the correct known classes or an unknown class.

The primary hurdle for UniDA lies in accurately detecting unknown samples while effectively transferring domain knowledge from the source to the target domain. For detecting unknowns, some prior works [12, 51] use manually set thresholds to reject a specific portion of target samples. While these methods achieve significant advancements, they encounter two primary concerns: 1) Distinct datasets require distinct thresholds, leading to complexities in establishing generic detection criteria; 2) Lack of supervision from target samples causes detection criteria to be overfitted from source samples, which can be more severe in UniDA due to the coexistence of both domain and label shifts. The first concern is efficiently handled by threshold-free methods [18, 35], which allows the model to learn the classification boundaries for automatically detecting private samples. However, the second problem is still under-explored in UniDA literature, where methods still rely on the output of one or two classifiers for classifying target samples. This reliance often results in suboptimal performance due to the classifier's tendency to overfit on source samples. Ensemble learning [10] can be the simplest way to tackle this, where we can use distinct classifiers during training which can reduce prediction variance, consequently mitigating issues of overfitting. However, directly adding classifiers will not only increase the model size but also lead to longer training and testing times, which is an undesirable outcome. An intuitive approach to address this challenge involves treating the classifier weight as a random variable and define a distribution for it. Subsequently, we can draw

samples from this distribution to obtain various distinct yet plausible classifiers, without causing substantial increases in model size or training time. Motivated by this, we introduce a novel framework titled **St**ochastic Binary Network for **Uni**versal Domain Adaptation (STUN).

STUN utilizes stochastic classifiers [26] where classifier weights are represented by Gaussian distribution $\mathcal{N}(\mu, \Sigma)$. Mean vector (μ) and diagonal covariance matrix (Σ) serve as distribution parameters that can be fine-tuned during the training process. Concretely, STUN uses stochastic binary network (SBN) consisting of $|\mathcal{C}_s|$ stochastic binary classifiers, where $|\mathcal{C}_s|$ represents the number of classes in the source domain. We introduce $|\mathcal{C}_s|$ binary classifiers instead of a single multi-class classifier in our framework due to their capability of efficiently learning unknown space without any threshold [35]. Training with different sampled SBN enables us to learn a generalized mean vector (μ) that serves as the final classifier weight during inference. To avoid negative knowledge transfer due to label shift, we propose a novel confidence score estimation technique to find the confidence scores for source and target samples using consistency between the outputs of sampled stochastic binary networks. These scores enable common class alignment during adversarial learning [52] by giving higher weights to the samples belonging to the shared classes across domains. Although weighted adversarial learning encourages the separation of "known" and "unknown" data samples, the problem of fragmented distributions in the target domain can still exist. Hence, most domain adaptation works [18, 23, 24] use standard FixMatch loss [39] for constructing compact clusters. However, FixMatch does not consider the overall feature structure in the target domain and does not enforce any cluster size balance, which is essential in UniDA because of the absence of prior information about the target domain due to label shift. Hence, different from existing works, we incorporate deep discriminative clustering (DDC) [15] in our framework to formulate a loss function that enforces the *structural regularization* and enables learning of compact clusters in the target domain.

Our main contributions can be summarized as follows: (1) We approach the UniDA problem from a novel perspective, $i.e.$ treating classifier weight as a distribution rather than a point weight vector for learning a generalized classifier by implicitly solving the overfitting issue from source samples. (2) We propose a novel confidence score estimation technique to efficiently distinguish between common and private classes samples. (3) To the best of our knowledge, our work is the first attempt to utilize deep discriminative clustering framework in UniDA for efficiently solving the issue of fragmented distributions in the target domain. (4) State-of-the-art results on three benchmark datasets containing diverse category shifts under the open-set and universal domain adaptation scenarios demonstrate the superiority of our framework. Notably, in the most challenging UniDA scenario, STUN exhibits a boost of 8.5% in H-score on the large-scale VisDA [31] dataset.

2. Related works

Universal domain adaptation. Universal domain adaptation (UniDA) is a challenging domain adaptation technique that assumes no prior knowledge about the relationship between source and target label spaces. UAN [51] first introduces the problem of UniDA and proposes an uncertainty mechanism for solving it. CMU [12] further improves the uncertainty mechanism by using three quantities, namely entropy, confidence, and consistency together. ROS [3] uses the self-supervision task of image rotation for domain alignment and known/unknown separation. Further, DCC [23] tackle the UniDA from a new perspective by differentiating private samples into different clusters instead of treating them as a whole. Recently, CPR [18] used a reciprocal classifier [5, 6] for detecting unknown target samples and solving UniDA. But, all these works follow the common approach of relying on the predictions of one or two classifiers making their predictions unreliable and biased towards the source domain. We tackle this issue efficiently by introducing stochastic classifiers [26] in our framework.

Stochastic classifiers. Bayesian Neural Networks [27] treats weights as variables, and the process of training estimates a marginal distribution that best fits the provided data. Similarly, in stochastic classifiers [26], weights are modeled by a multivariate Gaussian distribution whose parameters are jointly optimized during training. Numerous vision problems have recently been solved using stochastic classifiers. For example, S3C [21] uses stochastic classifiers for few-shot class incremental learning [49], and STOCO [40] uses consensus of them for SSL [43]. In our work, we use a network of stochastic binary classifiers to solve universal domain adaptation by exploiting the consistency between them for common class alignment and private class detection.

3. Methodology

Notations: In UniDA, we are provided with a labeled source domain, denoted as $D_s = (x_i^s, y_i^s)_{i=1}^{N_s}$, consisting of N_s labeled source samples. Additionally, we have an unlabeled target domain, denoted as $D_t = (x_i^t)_{i=1}^{N_t}$, containing N_t unlabeled target samples. Similarly, $\bar{D}_t = (\mathcal{A}(x_i^t))_{i=1}^{N_t}$ denotes the strongly augmented data from the target domain, where $\mathcal{A}$ represents the RandAugment [7] as a strong augmentation technique. Let $\mathcal{C}_s$ and $\mathcal{C}_t$ denote the source and target label set. The label set shared across domains is denoted as $\mathcal{C} = \mathcal{C}_s \cap \mathcal{C}_t$ and the private label sets for the source and target domains are denoted as $\overline{\mathcal{C}}_s = \mathcal{C}_s - \mathcal{C}$ and $\overline{\mathcal{C}}_t = \mathcal{C}_t - \mathcal{C}$ respectively.

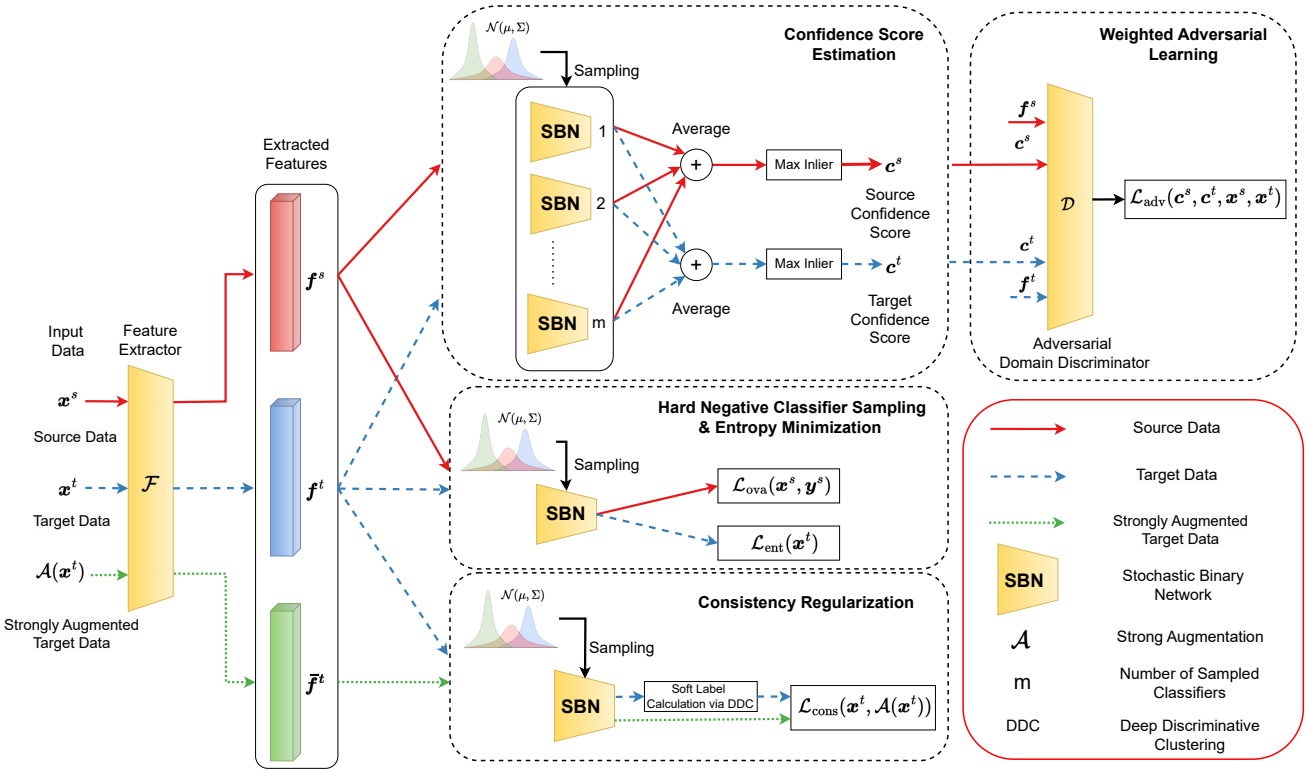


Figure 1. Illustration of the proposed STUN framework: Overall model consists of a feature extractor ($\mathcal{F}$), stochastic binary network (SBN), and an adversarial domain discriminator ($\mathcal{D}$). Hard negative classifier sampling (Sec. 3.2) is used for efficient training with source samples. The proposed confidence score estimation technique (Sec. 3.3) calculates robust confidence scores for source and target samples. Weighted adversarial learning is introduced for common class alignment between source and target domain (Sec. 3.4). Consistency regularization is used for learning compact feature distributions in the target domain via deep discriminative clustering (Sec. 3.5).

Basic framework: Fig. 1 provides a basic overview of our method, which comprises three main components: (a) Feature extractor ($\mathcal{F}$), which maps the input images x to features f: $f = \mathcal{F}(x)$; (b) Stochastic binary network (SBN) consisting of $|\mathcal{C}_s|$ binary classifiers sampled from a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$; (c) Adversarial domain discriminator ($\mathcal{D}$), which is used for common-class alignment.

3.1. Stochastic classifiers

Current UniDA works [3, 12, 18, 35, 38] use conventional classifiers, where weights are treated as point estimates and optimized during training. However, the classifier trained in this way can be easily overfitted by the source domain data, which results in suboptimal decision boundaries that cause negative transfer ($i.e.$ assigning private target samples to the source classes with high certainty). To rectify this, we have used stochastic classifiers in our framework.

In stochastic classifier [26], we treat the classifier weight vector as a random variable and use multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ with mean vector μ (after flattening the weight matrix) and diagonal covariance matrix Σ for modeling this. The mean (μ) can act as the final classifier weight during inference time, while the covariance matrix (Σ) represents inter-classifier discrepancy. So now, whenever we need a classifier, we can sample it from $\mathcal{N}(\mu, \Sigma)$, and the loss can be backpropagated to the learnable parameters μ and Σ. However, due to the non-differentiable nature of the sampling process, conventional end-to-training is not possible. Hence, we use reparametrisation trick [26] which can be written as: $\phi = \mu + \epsilon \odot \sigma$. Here, ϵ represents a sample drawn from $\mathcal{N}(\mu, \Sigma)$, σ represents diagonal of Σ and $\odot$ denotes the element-wise dot product between them. Now, we can do conventional end-to-training with ϕ as classifier weight.

Hence, our model can be trained with different classifiers in each iteration, enabling the learning of a generalizable weight vector (μ) that can be used to split target samples into known and unknown categories efficiently.

3.2. Hard negative classifier sampling

Unlike the tradition of using multi-class classifiers in UniDA, we introduce stochastic binary network (SBN) in

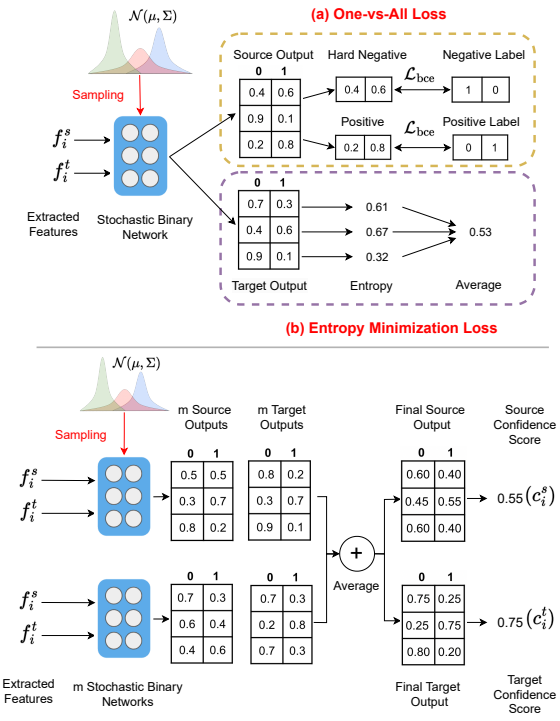


Figure 2. Overview of one-vs-all loss, entropy minimization loss, and confidence scores estimation: To illustrate these concepts, we consider a scenario with three classes and use $\mathcal{L}_{\text{bce}}$ to represent binary cross-entropy loss. In outputs of stochastic binary network, 0 and 1 index represents probability of outlier and inlier respectively. **Top:** (a) Shows the one-vs-all loss [Eq. (1)] calculation for source sample with label $y_i^s = 2$; (b) Shows the calculation of entropy minimization loss [Eq. (2)] for target samples. **Bottom:** Demonstrates the proposed confidence scores estimation technique for source and target samples [Sec. 3.3]. For ease of visualization, the number of sampled classifiers (m) is taken as 2.

our framework. SBN consists of $|\mathcal{C}_s|$ binary classifiers, one for each source domain class whose weights are sampled from a Gaussian distribution. Each binary classifier outputs a 2-dimensional vector denoting the probability of a sample being an outlier and inlier, respectively. For each source sample, the class corresponding to the given label is considered positive, and all other classes are considered negative. So, we train a binary classifier corresponding to the positive class to predict input as inlier. Now, a question arises of how to train classifiers belonging to negative classes? One simple approach is to train all classifiers of negative classes to predict input as an outlier. But, this approach suffers from 2 major drawbacks: (i) As shown in [28], decision boundary learned by classifiers trained in this way will not be effective with large number of classes, which can cause them to classify many unknown samples as known samples. (ii) training all $|\mathcal{C}_s|$ classifiers for each sample is both time-consuming and compute-intensive. Hence, we use hard negative classifier sampling (HNCS) [35] in our framework.

In HNCS, for each sample, only 2 classifiers belonging to the positive and the hard negative class will be trained. Hard negative class denotes the negative class which is very similar to the positive class. By training the corresponding classifier, we will be able to learn the discriminative decision boundaries among classes efficiently. Let, $p\left(\hat{y}^k|x\right)$ denote the output probability that the input x is an inlier for class k. Now, for the source samples, we introduce one-vs-all loss [35] which can be written as:

$$\mathcal{L}_{\text{ova}}\left(x^s, y^s\right) = -\log(p(\hat{y}^{y^s}|x^s)) - \min_{j \neq y^s} \log(1 - p(\hat{y}^j|x^s)) \tag{1}$$

In this, the first part computes the loss for a positive class, while the later part computes the loss for a hard negative class. For the target domain samples, we use open-set entropy minimization [35], in which we compute the entropy of all the binary classifiers of our stochastic binary network, and estimate the average and then train the model to minimize this. It can be written as follows:

$$\mathcal{L}_{ent}\left(x^t\right) = -\sum_{j=1}^{|C_s|} \left(p\left(\hat{y}^j \mid x^t\right) \log\left(p\left(\hat{y}^j \mid x^t\right)\right) + \left(1 - p\left(\hat{y}^j \mid x^t\right)\right) \log\left(1 - p\left(\hat{y}^j \mid x^t\right)\right)\right) \tag{2}$$

Minimizing Eq. (2) enhances the low-density separation in the target domain and helps in partially aligning known target samples to the source samples while keeping unknown samples as unknown. This unique capability is facilitated by using binary classifiers, which incorporate the concept of an "unknown" category, unlike multi-class classifiers that tend to forcibly align target samples to one of the source classes through entropy minimization. The calculation of $\mathcal{L}_{\text{ova}}$ and $\mathcal{L}_{\text{ent}}$ are depicted at the top of Fig. 2.

3.3. Confidence score estimation

Current UniDA works [3, 12, 18, 22, 23, 38, 51] use the output of one classifier for finding uncertainty scores (*e.g.* confidence, entropy and domain similarity) to exclude a specific portion of target samples. However, relying on only one classifier output results in unsatisfactory performance. Hence, motivated by the idea of co-training [2] and tri-training [53] some domain adaptation works [20, 35, 36, 48] use consensus of two or three diverse classifiers for getting more reliable results. But, one important question to ask is: why to stop at two, when the "wisdom of the crowd" principle suggests that more number of classifiers results into better performance in general. However, as shown in [26], the optimal number of classifiers is mostly task-specific. Moreover, simply adding classifiers into the model will not only increase computation time but also linearly increase the parameters of the model, which again causes the risk of overfitting. To tackle this, we use stochastic classifier [26] where

we can add approximately *infinite* number of classifiers into our model for making reliable decisions without increasing computation time and model parameters. To this end, we propose a novel confidence score estimation technique for target and source samples based on the consensus of m sampled stochastic binary networks.

Given a sample, confidence score will be calculated in four steps: (i) m stochastic binary networks (SBN) are sampled from distribution $\mathcal{N}(\mu, \Sigma)$. Due to the random and independent sampling, obtained SBN will be different enabling the robust confidence score estimation; (ii) Sample will be passed through each stochastic binary network to generate m independent $|\mathcal{C}_s| \times 2$ dimensional outputs; (iii) Average operation is performed to generate final output for a given sample; (iv) Confidence score will be calculated by taking maximum inlier probability of the final output. The confidence score for i-th source and target sample is denoted by c_i^s and c_i^t respectively. The bottom of Fig. 2 illustrates the calculation of these confidence scores with m=2.

The proposed technique is based on consistency among networks and gives a high score to the samples for which all m sampled networks are confident for the same class. Thus, the samples exhibiting larger scores will most likely belong to the common classes ($\mathcal{C}$) because the likelihood that all networks will commit the same mistake of considering the sample as inlier with high probability is very minimal.

3.4. Weighted adversarial learning

Adversarial learning [25, 41, 52] has been proven to be a powerful technique for reducing feature discrepancy and discovering invariant representations in domain adaptation. Nonetheless, due to the label shift in UniDA, incorporating adversarial learning directly into our framework would result in significant negative transfer, which could lead to the alignment of target private classes with the source classes. To rectify this, we introduce adversarial learning weighted by confidence scores (c.f. Sec. 3.3) for common class alignment. Weighted adversarial learning loss can be written as:

$$\begin{aligned}\mathcal{L}_{adv}\left(c^s, c^t, x^s, x^t\right) = &- \mathbb{E}_{x_i^s \sim D_s} c_i^s \log\left[\mathcal{D}\left(\mathcal{F}\left(x_i^s\right)\right)\right] \\ &- \mathbb{E}_{x_j^t \sim D_t} c_j^t \log\left[1 - \mathcal{D}\left(\mathcal{F}\left(x_j^t\right)\right)\right]\end{aligned} \tag{3}$$

It contains a feature extractor ($\mathcal{F}$) and adversarial domain discriminator ($\mathcal{D}$), where $\mathcal{D}$ aims to differentiate the source domain from the target domain, while the $\mathcal{F}$ tries to learn the domain invariant representations to fool the $\mathcal{D}$. Samples attributed to the private classes will exhibit lower confidence scores (c^s, c^t). Therefore, by incorporating these confidence scores into the adversarial learning, we can mitigate negative transfer resulting from samples belonging to the private classes. Finally, to optimize $\mathcal{F}$ and $\mathcal{D}$ in an end-to-end manner, we utilize the gradient reversal layer [14] to reverse the gradient between $\mathcal{F}$ and $\mathcal{D}$.

3.5. Consistency regularization via deep discriminative clustering

Consistency regularization [1, 39, 47] is a powerful solution for learning compact representations in semi-supervised learning [43], and FixMatch [39] is one of the representative of this. FixMatch promotes compact clusters by using the confident predictions of weakly augmented unlabeled data as the pseudo-labels for strongly augmented unlabeled data predictions. Let α and $\mathcal{A}$ denote the weak and strong augmentation, respectively. Similarly, p_i^t and $\bar{p}_i^t$ denote the predicted probability output for i^{th} unlabeled data using weak and strong augmentation, respectively. Using standard cross-entropy ($\mathcal{H}$), FixMatch loss on unlabeled data can be written as:

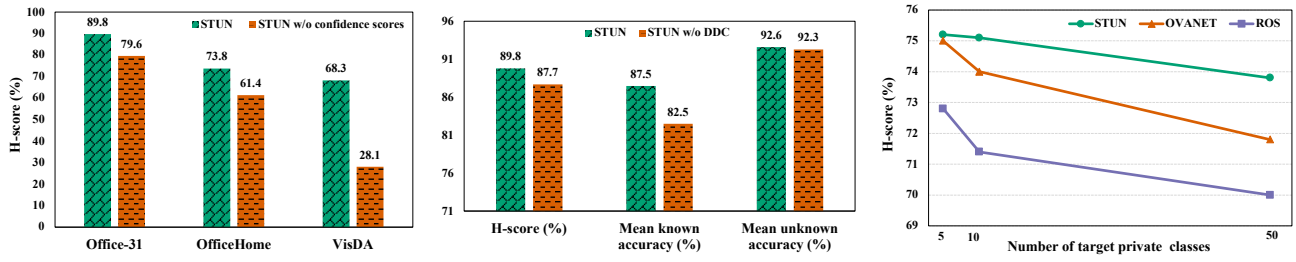
$$\mathcal{L}_{cons}(\alpha(x^t), \mathcal{A}(x^t)) = \frac{1}{N_t}\sum_{i=1}^{N_t} \mathrm{I}\left[\max p_i^t \geq \tau\right] \mathcal{H}(\bar{p}_i^t, \hat{y}_i), \tag{4}$$

where, $\hat{y}_i = \arg\max(p_i^t)$ and τ is the confidence threshold. However, we found two major problems on directly applying it in our framework. First, confident predictions are calculated using the output of only one classifier, which might lead to sub-optimal performance. Second, simply using hard labels will not enhance cluster size balance, which is crucial in UniDA due to the absence of prior information about the target distribution. The first problem we tackle by using robust confidence scores (c.f. Sec. 3.3) for selecting the target data. While, the second problem is addressed by using deep discriminative clustering [15, 19, 46] framework. It introduces auxiliary distribution ($i.e.$ soft label) by considering the overall feature structure in the target domain, which enforces *structural regularization* that leads to implicit discrimination in the target domain. Let, $P^t = \{p_i^t\}_{i=1}^{N_t}$ denotes the collective predicted probability outputs for target data. Then the auxiliary distribution $Q^t = \{q_i^t\}_{i=1}^{N_t}$ for the same can be obtained by optimizing the following objective:

$$\min_{Q^t} \mathrm{KL}(Q^t \mid\mid P^t) + \mathrm{KL}(\varrho^t \mid\mid u), \tag{5}$$

where, $\varrho^t = 1/N_t \sum_{i=1}^{N_t} q_i^t$ and u denotes the uniform distribution. The first term in Eq. (5) minimizes the KL divergence between probability distributions P^t and Q^t, which prevents the auxiliary distribution (Q^t) from deviating excessively from the original distribution (P^t). The second term minimizes the KL divergence between ϱ^t and π, which helps to achieve cluster size balance by avoiding degenerate solutions caused due to cluster merging. The closed form solution for Q^t is derived by [15] and can be written as:

$$q_{i,k}^t = \frac{p_{i,k}^t / \left(\sum_{i'=1}^{N_t} p_{i',k}^t\right)^{\frac{1}{2}}}{\sum_{k'=1}^{K} p_{i,k'}^t / \left(\sum_{i'=1}^{N_t} p_{i',k'}^t\right)^{\frac{1}{2}}}, \tag{6}$$



(a) (b) (c)

Figure 3. (a) Analysis of the importance of confidence scores in adversarial learning on all three datasets under UniDA setting. (b) Studying the effectiveness of deep discriminative clustering (DDC) using Office-31 dataset under UniDA setting. (c) Comparison with baselines on varying the number of target private classes ($\bar{\mathcal{C}}_t$) using UniDA setting for the OfficeHome dataset.

where, $p^t_{i,k}$ denotes the kth output of p^t_i for ith target instance. However, the above solution is valid for a multi-class classifier. Hence, we formulated an extended form of Eq. (6) for our stochastic binary network as:

$$q^t_{i,j,k} = \frac{p^t_{i,j,k} / \left(\sum_{i'=1}^{N_t} p^t_{i',j,k}\right)^{\frac{1}{2}}}{\sum_{j'=0}^{1} p^t_{i,j',k} / \left(\sum_{i'=1}^{N_t} p^t_{i',j',k}\right)^{\frac{1}{2}}}, \tag{7}$$

where, $p^t_{i,j,k}$ denotes the jth output of kth binary classifier for ith target instance, with i $\in \{1, N^t\}$, j $\in \{0, 1\}$ and k $\in \{0, |\mathcal{C}_s| - 1\}$. Finally, an improved formulation of Eq. (4) using generated auxiliary distribution (Q^t) as the soft labels for predictions of strongly augmented target data can be written as:

$$\mathcal{L}_{cons}(x^t, \mathcal{A}(x^t)) = \frac{1}{N_t} \sum_{i=1}^{N_t} \mathrm{I}\left[c^t_i \geq \tau\right] \bar{\mathcal{H}}(\bar{p}^t_i, q^t_i). \tag{8}$$

where, c^t_i is the confidence score for ith target instance and $\bar{\mathcal{H}}$ represents a modified version of standard cross-entropy that will calculate cross-entropy between the corresponding outputs of each binary classifiers of $\bar{p}^t_i$ and q^t_i and return their average. We exclude the target private samples in loss calculation by using confidence scores (c^t_i) with a threshold (τ). This accounts for their potential label inconsistency in neighboring data due to the use of a single unknown class for representing all the target private classes in both UniDA and ODA. The obtained loss $\mathcal{L}_{cons}$ Eq. (8) enables our framework to efficiently learn compact clusters by considering the overall structure of the target domain.

3.6. Overall objective

The model is jointly optimized using one-vs-all loss $\mathcal{L}_{\text{ova}}$ Eq. (1), entropy minimization loss $\mathcal{L}_{\text{ent}}$ Eq. (2), weighted adversarial learning loss $\mathcal{L}_{\text{adv}}$ Eq. (3) and consistency regularization loss $\mathcal{L}_{\text{cons}}$ Eq. (8). Thus, our overall objective is computed as follows:

$$\mathcal{L}_{\text{all}} = \mathcal{L}_{\text{ova}} + \lambda_1 \mathcal{L}_{\text{ent}} + \lambda_2 \mathcal{L}_{\text{adv}} + \lambda_3 \mathcal{L}_{\text{cons}}\,. \tag{9}$$

During the test phase, we utilize the trained μ as the final classifier weight. Now, if the maximum inlier probability of a target sample is greater than or equal to 0.5, it is classified as a known sample and is assigned the corresponding class. Otherwise, it is classified as an unknown sample.

4. Experiments

Datasets. We assess our model's performance on three widely used benchmark datasets. The Office-31 [33] dataset contains approximately 4700 images distributed across 31 categories from three domains: Amazon (A), DSLR (D), and Webcam (W). The OfficeHome [44] dataset is more extensive, with 15500 images spanning 65 categories across four domains: Art (A), Clipart (C), Product (P), and Real (R). Lastly, the VisDA [31] dataset is a challenging large-scale dataset with 12 categories, consisting of about 150K synthetic images in the source domain (S) and 50K real-world images in the target domain (R). We follow prior works [12, 23] to split datasets into common categories ($\mathcal{C}$), source private categories ($\bar{\mathcal{C}}_s$) and target private categories ($\bar{\mathcal{C}}_t$). We also show the category split ($\mathcal{C}/\bar{\mathcal{C}}_s/\bar{\mathcal{C}}_t$) of the dataset in all corresponding result tables.

Evaluation metric. Following prior works [12, 23, 35], we evaluate our method using H-score. H-score is the harmonic mean of accuracy on known classes (acc$_k$) and accuracy on unknown classes (acc$_u$). The H-score metric will be high when both known and unknown accuracies are high.

Implementation. We conduct our experiments on single NVIDIA GeForce RTX 3090 GPU using PyTorch [30]. Following previous works [12, 23, 35], we use ResNet50 [17] pretrained on ImageNet [9] as our feature extractor. We replace the last layer of ResNet50 with a new linear classification layer of stochastic classifier [26] and use m=10 for confidence score estimation. Due to the use of binary classifiers, the value of τ is simply set to 0.5. Additional experimental details and results are given in the supplementary.

Baselines. We compare our framework with state-of-the-art UniDA baselines namely: UAN [51] , CMU [12],

Table 1. H-score (%) results of each method in open-set domain adaptation (ODA) setting (best in **red** and second best in **blue**).

Method	Office-31 (10/0/11)							OfficeHome (25/0/40)												VisDA
	A2W	A2D	W2A	W2D	D2A	D2W	Avg	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	Avg	(6/0/6)
UAN	46.8	38.9	54.9	53.0	68.0	68.8	55.1	0.0	0.0	0.2	0.0	0.2	0.2	0.0	0.0	0.2	0.2	0.0	0.1	0.1	51.9
DCC	54.8	58.3	85.3	80.9	67.2	89.4	72.6	56.1	67.5	66.7	49.6	66.5	64.0	55.8	53.0	70.5	61.6	57.2	71.9	61.7	70.7
DANCE	78.8	84.9	68.3	88.9	79.1	78.8	79.8	61.9	61.3	63.7	64.2	58.6	62.6	67.4	61.0	65.5	65.9	61.3	64.2	63.0	67.5
ROS	71.7	65.8	82.0	98.2	87.2	94.8	83.3	60.1	69.3	76.5	58.9	65.2	68.6	60.6	56.3	74.4	68.8	60.4	75.7	66.2	50.1
OVANET	88.3	90.5	88.3	98.4	86.7	98.2	**91.7**	58.4	66.3	69.3	60.3	65.1	67.2	58.8	52.4	68.7	67.6	58.6	66.6	63.3	53.5
OVANET*	90.2	89.7	86.8	82.6	99.8	96.9	91.0	59.4	67.9	75.3	62.7	65.6	70.2	61.4	54.2	71.3	68.3	58.3	71.9	65.5	-
CPR	89.4	90.4	88.6	92.7	86.7	98.5	91.1	57.1	67.2	75.7	64.9	66.8	65.6	64.5	57.3	73.8	71.0	60.9	74.4	**66.6**	**79.4**
STUN	88.3	88.2	89.6	99.2	90.5	96.4	**92.0**	64.0	70.4	74.1	64.3	67.8	71.4	61.7	58.9	72.1	69.7	62.5	70.2	**67.3**	**80.0**

Table 2. H-score (%) results of each method in universal domain adaptation (UniDA) setting (best in **red** and second best in **blue**).

Method	Office-31 (10/10/11)							OfficeHome (10/5/50)												VisDA
	A2W	A2D	W2A	W2D	D2A	D2W	Avg	A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	Avg	(6/3/3)
UAN	58.6	59.7	60.3	71.4	60.1	70.6	63.5	51.6	51.7	54.3	61.7	57.6	61.9	50.4	47.6	61.5	62.9	52.6	65.2	56.6	30.5
CMU	67.3	68.1	72.2	80.4	71.4	79.3	73.1	56.0	56.9	59.2	67.0	64.3	67.8	54.7	51.1	66.4	68.2	57.9	69.7	61.6	34.6
I-UAN	79.4	71.5	83.0	80.7	81.0	81.5	79.5	54.1	63.1	65.2	70.5	68.3	73.2	61.9	51.8	63.8	69.8	55.6	70.7	64.0	-
ROS	71.3	71.4	79.2	95.3	81.0	94.6	82.1	54.0	77.6	85.3	62.1	71.0	76.4	68.8	52.4	83.2	71.6	57.8	79.2	70.0	50.1
DANCE	71.5	78.6	72.2	87.9	79.9	91.4	80.3	61.0	60.4	64.9	65.7	58.8	61.8	73.1	61.2	66.6	67.7	62.4	63.7	63.9	42.8
DCC	78.5	88.5	75.9	88.6	70.2	79.3	80.2	58.0	54.1	58.0	74.6	70.6	77.5	64.3	73.6	74.9	81.0	75.1	80.4	70.2	43.0
PCL	80.5	82.8	65.7	93.5	68.7	78.8	78.3	52.9	71.7	84.5	70.8	72.9	82.1	66.8	43.8	84.2	76.4	84.2	76.5	70.3	-
OVANET	79.4	85.8	84.0	94.3	80.1	95.4	86.5	62.8	75.6	78.6	70.7	68.8	75.0	71.3	58.6	80.5	76.1	64.1	78.9	71.8	53.1
SNAIL	80.6	82.4	86.4	94.2	84.2	96.5	87.4	55.9	57.9	63.1	52.5	55.4	56.4	66.8	53.5	61.1	64.3	53.8	63.2	58.6	**59.8**
OVANET*	80.9	85.4	82.5	97.5	82.5	92.3	86.9	62.0	77.7	86.3	70.0	70.1	79.3	70.0	58.8	82.5	76.8	64.0	80.5	**73.2**	-
CPR	81.4	84.4	91.3	96.8	85.5	93.4	**88.8**	59.0	77.1	83.7	69.7	68.1	75.4	74.6	56.1	78.9	80.5	63.0	81.0	72.3	58.2
STUN	83.9	89.5	89.0	95.8	86.1	94.7	**89.8**	64.3	77.8	81.3	70.1	70.0	75.8	75.3	63.5	81.6	78.9	65.6	81.0	**73.8**	**68.3**

Table 3. Analysis of the significance of stochastic classifiers (SC) using H-score (%) on Office-31 dataset under the UniDA setting.

Method	A2W	A2D	W2A	W2D	D2A	D2W	Avg
STUN w/o SC	80.0	88.9	87.8	95.1	82.1	93.1	87.8
STUN	**83.9**	**89.5**	**89.0**	**95.8**	**86.1**	**94.7**	**89.8**

I-UAN [50] , ROS [3], DANCE [34], OVANET [35], DCC [23], SPA [22], PCL [38], SNAIL [16] and CPR [18]. Since SPA is a plug-in method, we use OVANET* to represent OVANET+SPA. We are not comparing with recent arXiv works like UniAM [54] and Deng et al. [8] because they have utilized large-scale pre-trained models (*e.g.* CLIP [32] and ViT [11]), giving them an unfair advantage over our work and other ResNet50-based baselines.

4.1. Main results

Results in Table 1 show that our method consistently outperforms other baselines across all three datasets in the ODA setting. Especially in challenging domain adaptation scenarios such as A2C and R2C within the OfficeHome dataset, other methods struggle to effectively align common classes, resulting in lower H-scores. In contrast, our method consistently achieved higher scores even in these complex situations. Similarly, results in Table 2 demonstrate that our method achieves new state-of-the-art across all three datasets in the most challenging UniDA setting. On the large-scale VisDA dataset, our method gives 8.5% improvement in H-score. Collectively, superior results across datasets showcase the stronger capability of our framework in achieving common class alignment and private class detection under the varied ODA and UniDA settings.

4.2. Ablation studies

Importance of stochastic classifiers. Table 3 illustrates the significance of stochastic classifiers (SC) where a drop of 2 % in overall H-score is observed on replacing stochastic classifiers with conventional classifiers. Notably, this decline remains consistent across all adaptation scenarios and becomes more pronounced in complex adaptation settings (*e.g.,* A2W, D2A), exhibiting an almost 4% reduction.

Effect of $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{cons}}$. In Table 4, we study the individual contributions of $\mathcal{L}_{\text{adv}}$ Eq. (3) and $\mathcal{L}_{\text{cons}}$ Eq. (8) by removing one of them at a time. It is evident that the omission of either component leads to a reduction in the H-score.

Table 4. Effect of $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{cons}}$ on H-score (%) for Office-31.

Method	UniDA	ODA
STUN w/o $\mathcal{L}_{\text{cons}}$	81.0	88.8
STUN w/o $\mathcal{L}_{\text{adv}}$	87.5	90.8
STUN w/o $\mathcal{L}_{\text{cons}}, \mathcal{L}_{\text{adv}}$	78.1	87.8
STUN	**89.8**	**92.0**

This effect is more pronounced in the challenging UniDA context, where the removal of both $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{cons}}$ results in a substantial decline of over 10%. This phenomenon is attributed to the presence of both source and target private classes within UniDA, which elevates the chances of fragmented clusters and alignment between shared and private classes in the absence of $\mathcal{L}_{\text{cons}}$ and $\mathcal{L}_{\text{adv}}$, respectively.

Significance of weighted adversarial learning. We investigate the importance of weighted adversarial learning by removing confidence scores (*i.e.* c^s, c^t) from Eq. (3). As depicted in Fig. 3a, it leads to a performance decline across all datasets. Particularly in VisDA, which is a large-scale challenging dataset significant drop of 40% can be seen. These outcomes affirm the crucial role of generated confidence scores in mitigating negative transfer by giving higher weights to the samples belonging to the common classes and lower weights to those belonging to private classes.

Effectiveness of DDC. We introduce deep discriminative clustering (DDC) in our framework to generate better soft labels by considering the overall feature structure of the target domain. We verify its effectiveness by replacing soft labels generated by DDC (q_i^t) with soft labels produced by the classifier itself (p_i^t) in Eq. (8). By doing this, nearly 2% drop in H-score is observed from Fig. 3b. Importantly, 5% of decline is observed in the mean accuracy of known classes (acc_k), which occurs due to the relatively less compact clusters in the target domain, which causes various known samples to enter into the unknown class region. More details are in Sec. 3.3 of the supplementary.

Effect of varying degree of openness. In Fig. 3c, we examine the behavior of CPR on varying degree of openness. Following [35], we conduct the UniDA experiment within the OfficeHome dataset by varying the number of target private classes ($\overline{\mathcal{C}}_t$) while keeping the number of common classes ($\mathcal{C}$) and source private classes ($\overline{\mathcal{C}}_s$) constant. Remarkably, our framework consistently outperforms other baselines under the different values of $\overline{\mathcal{C}}_t$. Of particular note, only a smaller drop of 1.4% in H-score is observed on varying $\overline{\mathcal{C}}_t$ from 5 to 50, which underscores the robustness of our method on varying degrees of openness.

Varying the number of stochastic classifiers. We performed a study in Table 5 to examine the performance of STUN on varying the number of stochastic classifiers (m) in the proposed confidence score estimation technique (c.f. Sec. 3.3). The H-score increases with the growth of m when

Table 5. H-score (%) and time comparison (hours:minutes) upon varying the number of stochastic classifiers (m) on "A2D" setting of Office-31 dataset under UniDA scenario.

m	1	2	4	6	8	10	12	14
H-score	86.4	88.0	88.3	89.0	89.2	**89.5**	88.0	87.9
Time	1:25	1:25	1:26	1:27	1:27	1:28	1:29	1:29

(a) Ground truth labels (b) Predictions by STUN

Figure 4. Feature visualization on "W2D" of Office-31 dataset under ODA setting. Black markers represent samples of private classes, while color markers represent common class samples.

m$\leq$10, signifying that involving more classifiers enhances generalization ability through a more rigorous score estimation criterion. Conversely, a contrasting pattern is observed when m$>$10, implying that the estimation criteria become excessively strict, resulting in low confidence scores of many known samples. We also observed only a marginal increase in training time on varying m from 1 to 14, highlighting the capability of stochastic classifiers to build robust ensembles without significant training time escalation.

Feature visualization. We employ t-SNE [42] for visualizing the learned target features together with their actual labels and our predicted labels. As depicted in Fig. 4, learned features of samples belonging to private categories and common categories are well separated and features belonging to the same class exhibit compact clustering.

5. Conclusion

This paper proposes a novel framework STUN, which tackles UniDA problem from a new perspective by treating classifier weights as a distribution rather than a point estimate. This enables us to reduce overfitting by leveraging a potentially infinite number of classifiers during training without significantly increasing model size and training time. Additionally, we use consistency between these sampled networks to derive robust confidence scores, which allows us to achieve common class alignment via weighted adversarial learning. Finally, we utilize deep discriminative clustering framework to derive a loss function for achieving compact clusters in the target domain. Extensive experiments in open-set and universal DA scenarios across three benchmark datasets highlight the superiority and robustness of our framework. Remarkably, STUN significantly outperforms existing state-of-the-art by 8.5% on the large-scale VisDA dataset under the challenging UniDA scenario.

References

[1] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019. 5

[2] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998. 4

[3] Silvia Bucci, Mohammad Reza Loghmani, and Tatiana Tommasi. On the effectiveness of image rotation for open set domain adaptation. In *European Conference on Computer Vision*, 2020. 2, 3, 4, 7

[4] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Partial transfer learning with selective adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2724–2732, 2018. 1

[5] Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. Adversarial reciprocal points learning for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8065–8081, 2021. 2

[6] Guangyao Chen, Limeng Qiao, Yemin Shi, Peixi Peng, Jia Li, Tiejun Huang, Shiliang Pu, and Yonghong Tian. Learning open set network with discriminative reciprocal points. In *European Conference on Computer Vision*, 2020. 2

[7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 2

[8] Bin Deng and Kui Jia. Universal domain adaptation from foundation models. *arXiv preprint arXiv:2305.11092*, 2023. 7

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[10] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14:241–258, 2020. 1

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 7

[12] Bo Fu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Learning to detect open classes for universal domain adaptation. In *European Conference on Computer Vision*, 2020. 1, 2, 3, 4, 6

[13] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, 2015. 1

[14] 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), 2016. 5

[15] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5736–5745, 2017. 2, 5

[16] Zhongyi Han, Wan Su, Rundong He, and Yilong Yin. Snail: Semi-separated uncertainty adversarial learning for universal domain adaptation. In *Asian Conference on Machine Learning*, pages 436–451, 2023. 7

[17] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2015. 6

[18] Sungsu Hur, Inkyu Shin, Kwanyong Park, Sanghyun Woo, and In So Kweon. Learning classifiers of prototypes and reciprocal points for universal domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 531–540, 2023. 1, 2, 3, 4, 7

[19] Mohammed Jabi, Marco Pedersoli, Amar Mitiche, and Ismail Ben Ayed. Deep clustering: On the link between discriminative models and k-means. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(6):1887–1896, 2019. 5

[20] Saurabh Kumar Jain and Sukhendu Das. Marrs: Modern backbones assisted co-training for rapid and robust semi-supervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 4579–4588, 2023. 4

[21] Jayateja Kalla and Soma Biswas. S3c: Self-supervised stochastic classifiers for few-shot class-incremental learning. In *European Conference on Computer Vision*, 2022. 2

[22] Jogendra Nath Kundu, Suvaansh Bhambri, Akshay R Kulkarni, Hiran Sarkar, Varun Jampani, et al. Subsidiary prototype alignment for universal domain adaptation. In *Advances in Neural Information Processing Systems*, 2022. 4, 7

[23] Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9752–9761, 2021. 2, 4, 6, 7

[24] Kai Li, Chang Liu, Handong Zhao, Yulun Zhang, and Yun Fu. Ecacl: A holistic framework for semi-supervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8578–8587, 2021. 2

[25] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, 2018. 1, 5

[26] Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi-Zhe Song, and Tao Xiang. Stochastic classifiers for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9111–9120, 2020. 2, 3, 4, 6

[27] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012. 2

[28] Shreyas Padhy, Zachary Nado, Jie Jessie Ren, Jeremiah Zhe Liu, Jasper Snoek, and Balaji Lakshminarayanan. Revisiting one-vs-all classifiers for predictive uncertainty and out-of-distribution detection in neural networks. *ArXiv*, abs/2007.05134, 2020. 4

[29] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 754–763, 2017. 1

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zach DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[31] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *ArXiv*, abs/1710.06924, 2017. 2, 6

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021. 7

[33] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision*, 2010. 6

[34] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. In *Advances in Neural Information Processing Systems*, 2020. 7

[35] Kuniaki Saito and Kate Saenko. Ovanet: One-vs-all network for universal domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8980–8989, 2021. 1, 2, 3, 4, 6, 7, 8

[36] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 2988–2997, 2017. 4

[37] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018. 1

[38] Xinxin Shan, Tai Ma, and Ying Wen. Prediction of common labels for universal domain adaptation. *Neural Networks*, 2023. 3, 4, 7

[39] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, 2020. 2, 5

[40] Hui Tang, Lin Sun, and Kui Jia. Stochastic consensus: Enhancing semi-supervised learning with consistency of stochastic classifiers. In *European Conference on Computer Vision*, 2022. 2

[41] 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. 5

[42] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *The Journal of Machine Learning Research*, 9(11), 2008. 8

[43] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020. 2, 5

[44] 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 5385–5394, 2017. 6

[45] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020. 1

[46] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pages 478–487, 2016. 5

[47] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, 2020. 5

[48] Luyu Yang, Yan Wang, Mingfei Gao, Abhinav Shrivastava, Kilian Q Weinberger, Wei-Lun Chao, and Ser-Nam Lim. Deep co-training with task decomposition for semi-supervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8906–8916, 2021. 4

[49] Qing Yang, Yudi Gu, and Dongsheng Wu. Survey of incremental learning. In *2019 Chinese Control and Decision Conference (ccdc)*, pages 399–404. IEEE, 2019. 2

[50] Yueming Yin, Zhen Yang, Xiaofu Wu, and Haifeng Hu. Pseudo-margin-based universal domain adaptation. *Knowledge-Based Systems*, 229:107315, 2021. 7

[51] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2715–2724, 2019. 1, 2, 4, 6

[52] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5031–5040, 2019. 2, 5

[53] Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541, 2005. 4

[54] Didi Zhu, Yincuan Li, Junkun Yuan, Zexi Li, Yunfeng Shao, Kun Kuang, and Chao Wu. Universal domain adaptation via compressive attention matching. *arXiv preprint arXiv:2304.11862*, 2023. 7