SNAIL: Semi-Separated Uncertainty Adversarial Learning for Universal Domain Adaptation

Zhongyi Han HANZHONGYICN@GMAIL.COM
Wan Su SUWAN@MAIL.SDU.EDU.CN
Rundong He RUNDONG_HE@MAIL.SDU.EDU.CN
Yilong Yin YLYIN@SDU.EDU.CN
School of Software, Shandong University

Editors: Emtiyaz Khan and Mehmet Gonen

Abstract

Universal domain adaptation (UniDA) is a new sub-topic of unsupervised domain adaptation. It handles the problem that the source or target domain possibly has open-class samples. The inborn challenge is to detect the open-class samples in the test phase. Pioneering studies could be viewed as dependent-detector-based methods. They cleverly design efficient uncertainty metrics (*e.g.*, confidence, entropy, distance) based on the outputs of domain adaptation models (predictor) to detect open-class samples. However, they have a pain point in setting extremely-sensitive and task-dependent thresholds on the uncertainty metrics to filter open-class samples. To bypass this pain point, we propose a semi-separated-detector-based method, Semi-Separated Uncertainty Adversarial Learning (SNAIL). We build a semi-separated uncertainty decision-maker to enable sensitive-threshold-free detection. It receives multiple uncertainty metrics as attributes and separately learns the thresholds of uncertainty metrics in a multi-level decision rule. For some challenging tasks, the uncertainty margins between common and open classes are subtle, leading to difficulty learning optimal decision rules. We present the uncertainty separation loss to enlarge the uncertainty margin. Further, forcibly aligning the distributions could incorrectly align the open classes to common classes. Thanks to the open-class detection strategy, we design the conditional-weighted adversarial loss that adversarially and selectively matches the feature distributions to defeat the distribution misalignment problem. Extensive experiments show that SNAIL remarkably outperforms the state-of-the-art domain adaptation methods, with over 25% improvements in open-class detection accuracy for some tasks.

Keywords: domain adaptation; transfer learning; semi-separated detector; new class

1. Introduction

As one of the booming branches of artificial intelligence, machine learning has been continuously connected to various data analysis and processing scenarios, actively promoting the intelligent process of application fields, generating substantial economic and social benefits (Shalev-Shwartz and Ben-David, 2014). Machine learning techniques already demonstrate more stable performance in closed and static task environments where the identical distribution assumption is well satisfied (LeCun et al., 2015; He et al., 2016). However, this assumption does not hold in one of the open and dynamic task environments where the data distribution does change over time (Han et al., 2020). Unsupervised domain adaptation is a

superb strategy that enables an algorithm to transfer from a labeled source domain to an unlabeled domain under the distribution shift (Han et al., 2021; Ben-David et al., 2007).

Further, the distribution shift problem often co-occurs with another problematic situation in the open and dynamic task environment: the openness of the label set. The label set of source or target domain perhaps holds open classes, further degrading the robustness of machine learning techniques. For example, before the emergence of COVID-19, the standard domain adaptation algorithm for pneumonia detection could only identify common-type pneumonia cross domains (*e.g.*, different hospitals, various types of devices). After COVID-19 suddenly emerges in clinical but has not yet been detected, the target domain possibly contains COVID-19 samples. At this time, one cannot guarantee the algorithm's robustness. Thus it is urgent to develop new domain adaptation algorithms to detect COVID-19 timely. To overcome this latest issue, we advance the new topic called *universal domain adaptation* (UniDA) to remove the label set constraints (You et al., 2019).

The new problem setting introduces two entangled challenges: (1) How to detect the open-class samples without prior knowledge about the label set? Since open-class samples usually have larger uncertainties than common class samples, most of the pioneering UniDA works are dependent-detector-based methods that depend on the output of the domain adaptation models (*e.g.*, confidence metric, entropy metric, similarity) to detect open classes (You et al., 2019; Fu et al., 2020). However, they require extremely-sensitive and task-dependent thresholds acted on the above uncertainty metrics to split open and common classes. The quality of a dependent detector is also highly influenced by the quality of the model output (Hendrickx et al., 2021). (2) How to match the distributions across domains under the unexpected category shift? Forcibly aligning the distributions could incorrectly align the open classes to common classes, leading to catastrophic misalignment (Saito et al., 2020). Solving the first challenge is the prerequisite for the second challenge. Only when we accurately identify the open classes will reduce its impact on distribution alignment.

In this paper, we propose Semi-separated uNcertainty AdversarIal Learning (SNAIL), an effective and comprehensive solution that enables sensitive-threshold-free open-class detection and weak distribution alignment. The key insight is that, rather than designing a dependent detector, a separated detector is agnostic of domain adaptation models (predictor) and can be trained independently of the predictor. The separated detector could be an extra decision-maker that can separately decide the classification boundary between open and common classes without sensitive thresholds. However, not sharing information between the predictor and detector results often in sub-optimal detection performance. To balance the independence and optimal performance, we propose a semi-separated uncertainty decision-maker that depends on the outputs of predictors as attributes and separately learns the thresholds of them to classify each sample into an open or common class. For some challenging tasks, the boundary between open and common classes is ambiguous and difficult to learn. Thus we propose the uncertainty separation loss that draws an explicit uncertainty boundary with a large uncertainty margin. We further propose the conditional-weighted adversarial loss that adversarially and selectively matches common class feature distributions by leveraging the uncertainty boundary to avoid the catastrophic misalignment problem. We ultimately integrate our methods into an end-to-end solution, successfully mitigating the negative influence of the category gap and the distribution shift.

The main contributions of this paper are:

(1) We propose the semi-separated uncertainty decision-maker to resolve the common pain point regarding the manual thresholds of uncertainty metrics. We establish the clear separation between the detector and the predictor and make it transparent why a prediction was or was not made. The proposal can also blend into other UniDA frameworks.

(2) We propose two novel learning objectives: uncertainty separation loss and conditional-weighted adversarial loss, designed for category gap and distribution shift, respectively. Their novelty lies in the uncertainty boundary with large margins that help filter out open-class samples to promote common-class adaptation when conducting adversarially training.

(3) We empirically validate the proposal on four UniDA benchmarks, on which SNAIL outperforms pioneering methods on multiple evaluation metrics, especially in open-class detection. Taking Figure 1 as an example, the open-class detection performance (*e.g.*, H-score) of our method remarkably outperforms the state-of-the-art method (CMU) by almost 25% H-score on the VisDA dataset.

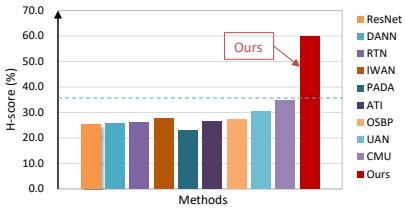


Figure 1: Open-class detection performance on VisDA.

2. Related Work

This section outlines closed-set domain adaptation, partial domain adaptation, open-set domain adaptation, and universal domain adaptation. Figure 2 shows their differences.

Closed-set domain adaptation has been gaining momentum in the past decade to cope with the distribution shift (Han et al., 2021). Previous works have fulfilled in-depth research in theory, which have firmly established the algorithm foundation (Han et al., 2020; Ben-David et al., 2007; Zhang et al., 2019). From the way of aligning distribution shift, previous works can be assorted into importance estimation (Sugiyama et al., 2008), moment matching (Huang et al., 2006; Long et al., 2015), pseudo labeling (Sener et al., 2016; Saito et al., 2017), and adversarial training (Ganin and Lempitsky, 2015; Long et al., 2018). Previous works implicitly assume the label set is closed, which is difficult to hold in practice.

Partial domain adaptation assumes the label set of the target domain is a subset of the source domain. This problem occurs when a source domain has a large annotated dataset, such as ImageNet (Deng et al., 2009), Global Financial Development Database (Cihak et al., 2012). There are two strategies to deal with this problem. One intuition strategy down-weights or picks source examples that are dissimilar to target samples (Cao et al., 2018a,b). Another strategy uses a domain discriminator to estimate the domain similarity between private source examples and target samples (Cao et al., 2019; Zhang et al., 2018). The drawback of partial domain adaptation is the limited application scope.

Open-set domain adaptation deals with that both source and target domains have common and open classes, but the prior knowledge about the emergence of open classes is known (Jing et al., 2021; Pan et al., 2020). One of the pioneering works in this line is proposed by Busto and Gall (2017) that learned to map from the source domain to the target domain by jointly solving the assignment problem defined by a binary linear program. Motivated by this, Saito et al. (2018) utilized adversarial training to learn domain-invariant

features that align the target samples of common classes with the source domain and detect the target samples of open classes. Baktashmotlagh et al. (2019) learned low-dimensional features to factorize the target data into shared and private parts. The limitation of open-set domain adaptation is that the prior knowledge is known.

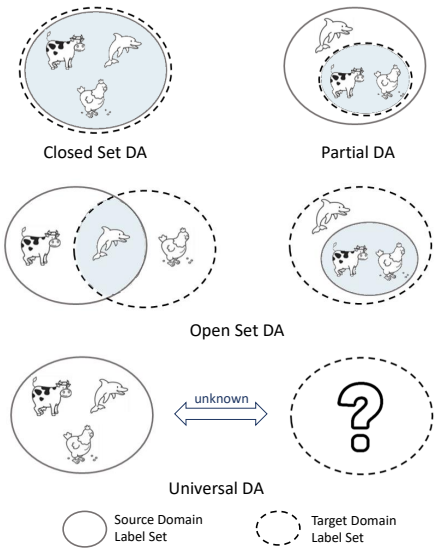


Figure 2: Label set differences.

Universal domain adaptation relaxes the prior knowledge and removes all constraints on the label sets. It thus has a broader scope of application but poses a challenge to detect open-class samples. You et al. (2019) opened this problem and proposed the novel universal adaptation network that can measure domain similarity and entropy to evaluate the transferability of target samples. Fu et al. (2020) further designed multiple classifiers to calibrate the uncertainties. However, they use a soft-weighted loss to align the distributions across domains that could incorrectly move the open-class features to common classes. To avoid this problem, Saito et al. (2020) proposed a domain adaptative neighborhood clustering to align the feature distributions between two domains weakly. Lifshitz and Wolf (2021) proposed to score the target samples and apply a threshold to select common class samples. The common bottleneck of previous studies is that they require manual thresholds for prediction uncertainty metrics, causing them to drop robustness.

3. Methodology

3.1. Learning Setup

In universal domain adaptation, we denote by $\mathbf{x}$ the input, $\mathbf{y}$ the label, and d the Bernoulli variable indicating to which domain $\mathbf{x}$ belongs. A sample of n_s labeled training examples $(\mathbf{x}_s^i, \mathbf{y}_s^i)_{i=1}^{n_s}$ $(d = 0)$ is drawn according to the source distribution p defined on $\mathcal{X}^s \times \mathcal{Y}^s$. $\mathcal{X}^s$ is the input set and $\mathcal{Y}^s$ is the label set. Meanwhile, a sample of n_t unlabeled test examples $(\mathbf{x}_t^i)_{i=1}^{n_t}$ $(d = 1)$ is drawn according to the target distribution q defined on $\mathcal{X}^t \times \mathcal{Y}^t$. The crucial difference is that both source label set $\mathcal{Y}^s$ and target label set $\mathcal{Y}^t$ contains common label set and private label set. $\mathcal{Y} = \mathcal{Y}^s \cap \mathcal{Y}^t$ is the common set shared by source domain and target domain. $\bar{\mathcal{Y}}^s = \mathcal{Y}^s \setminus \mathcal{Y}$ and $\bar{\mathcal{Y}}^t = \mathcal{Y}^t \setminus \mathcal{Y}$ are the private label sets held by both domains, respectively. Note that the target label set is unavailable at training and only used to define the UniDA problem. The objective is to detect target open-class data in $\bar{\mathcal{Y}}^t$ and predict the accurate label for target data in $\mathcal{Y}$.

3.2. Technical Challenges and Overview

As pointed out in the pioneering work (You et al., 2019), the key technical challenges are the **category gap** and the **domain gap** between two domains without any prior knowledge about category information. The challenges hinder the naive application of existing closed,

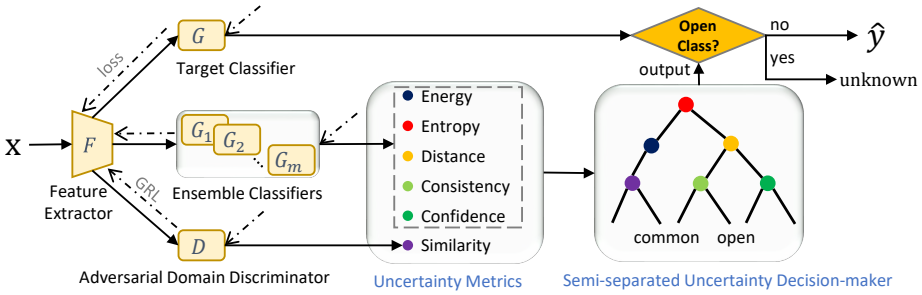


Figure 3: The framework of the semi-separated umcertainty adversarial learning.

partial, or open-set domain adaptation methods on the UniDA problem. A natural idea to detect target open-class data in UniDA is to define a threshold to separate the quantities of uncertainty metrics termed w^t (*e.g.*, confidence, entropy). With a threshold z_0, the class c of an input sample $\mathbf{x}$ can be predicted by thresholding class probability vector $\hat{\mathbf{y}}$ w.r.t. z_0:

$$c = \begin{cases} \text{unknown} & w^t(\mathbf{x}) < z_0 \\ \arg\max(\hat{\mathbf{y}}) & w^t(\mathbf{x}) \geq z_0 \end{cases} \tag{1}$$

which either detects the target sample $\mathbf{x}$ as an "unknown" class or classifies it to one of the common classes. While the thresholding strategy is intuitive, the threshold parameters are susceptible to set because we need cross-validation to develop validated thresholds manually. Setting a fixed threshold of z_0 by cross-validation is impractical in domain adaptation because we cannot access the annotated data of the target domain. Moreover, since the quantities of uncertainty metrics have substantial differences across different transfer tasks, the fixed (task-independent) thresholds will cause biased decisions, while the task-dependent thresholds are time-consuming. To avoid the manual setting of susceptible threshold parameters, we propose the *semi-separated uncertainty decision-maker (SUD)* that can adaptatively determine the thresholds of multiple uncertainty metrics and cut them into multi-level decision rules (elaborated in Sec. 3.3). Further, it is in the expectation that open-class data have larger uncertainty quantities than common class data in the target domain. However, the uncertainty metrics are less discriminative due to the underlying domain gap and the task difficulty. To ensure the uncertainty quantities are distinguishable, we propose the novel *uncertainty separation (US) loss* to keep the target open-class samples far from the common class samples (elaborated in Sec. 3.4). Moreover, depending on the semi-separated uncertainty decision-maker, we realize the *conditional-weighted adversarial (CA) loss* to avoid incorrectly aligning the open classes to common classes (elaborated in Sec. 3.5).

As shown in Figure 3, the architecture of Semi-separated uNcertainty AdversarIal Learning (SNAIL) includes four essential parts: the feature extractor F to learn the domain-invariant feature representation, the target classifier G to predict the labels of inputs, the ensemble classifiers $G_1, ..., G_m$ to estimate and calibrate multiple uncertainty metrics (Fu et al., 2020; Venkat et al., 2020), and the adversarial domain discriminator D to adversarially match the feature distributions and quantify the similarity z_{simi} of $\mathbf{x}$ to the source domain.

3.3. Semi-separated Uncertainty Decision-maker

To avoid the manual threshold z_0 in Eq. (1), we propose the semi-separated uncertainty decision-make. Compared with previous dependent-detector methods, the semi-separated uncertainty decision-maker adaptatively learns optimal thresholds to detect open-class samples. Given each input target sample $\mathbf{x}$, its quantities of multiple uncertainty metrics $\mathbf{v}(\mathbf{x}) = (z_1, \cdots, z_u)$, and its categorical prediction $\hat{\mathbf{y}}$ over the source label set $\mathcal{Y}^s$, the class c can be predicted with the non-parametric SUD:

$$c = \begin{cases} \text{unknown} & \text{SUD}(\mathbf{v}(\mathbf{x})) = 1 \\ \arg\max(\hat{\mathbf{y}}) & \text{SUD}(\mathbf{v}(\mathbf{x})) = 0 \end{cases} \tag{2}$$

which either detects the target sample $\mathbf{x}$ as an "unknown" class or classifies it as one of the common classes. If a leaf node predicts $\mathbf{x}$ being an open class, then the domain adaptation model outputs an open class, and $\hat{y}$ otherwise.

In practice, we adopt the classification and regression tree (CART) (Breiman et al., 1984) to establish the semi-separated uncertainty decision-maker because CART can provide uncertainty selection, tree generation, and pruning simultaneously. The generation process of the semi-separated uncertainty decision-maker is recursively constructing a binary classification and regression tree. We utilize the Gini index minimization criterion to perform uncertainty selection. The smaller the Gini index, the more optimal the uncertainty metric. We use the Gini index rather than information gain because information gain contradicts the entropy, one leader of the uncertainty metrics. Gini Index, also known as Gini impurity, calculates the amount of probability of a specific attribute that is classified incorrectly when selected randomly. The period of semi-separated uncertainty decision-maker generation is after the training of the domain adaptation network (predictor) because the predictor outputs are more stable at this time.

To train the semi-separated uncertainty decision-maker, we construct a new training dataset. We collect multiple uncertainty metrics: energy z_{ener} (Liu et al., 2020), entropy z_{entr}, confidence z_{conf}, consistency z_{cons}, distance z_{dist}, and similarity z_{simi} of each target instance. We normalize them by minmax normalization to unify them in $[0, 1]$. Then we compute $\mathbf{z}$ by aggregating the six uncertainties: $\mathbf{z} = (z_{\text{ener}} + z_{\text{entr}} + (1 - z_{\text{conf}}) + z_{\text{cons}} + z_{\text{dist}} + (1 - z_{\text{simi}}))/6$. The higher the $\mathbf{z}(\mathbf{x}_t)$, the more likely $\mathbf{x}_t$ is in open classes. We further sort the aggregated uncertainties of all target samples ($\mathbf{z}^t$) in ascending order. To ensure label cleanness and class balance, we select top $n\%$ samples as open-class data and select bottom $n\%$ samples as common class data. Although $n\%$ is a parameter that needs to be set, it is not sensitive and different from z_0 in Eq. (1) because $n\%$ does not directly affect the open-class detection. It only affect the training data amount of the semi-separated uncertainty decision-maker. Indeed, we have empirically proved that different values $n\%$ have same performance on multiple datasets. We finally create the training dataset termed $D = \{(\mathbf{v}_1, y_1), (\mathbf{v}_2, y_2), \cdots, (\mathbf{v}_N, y_N)\}$. $\mathbf{v}_i = (z^i_{\text{ener}}, z^i_{\text{entr}}, z^i_{\text{conf}}, z^i_{\text{cons}}, z^i_{\text{dist}}, z^i_{\text{simi}})$ is the attribute vector. $y_i \in \{-1, +1\}$ is a class label where the positive class represents open-class while the negative class represents common class. The learning objective is to establish the semi-separated uncertainty decision-maker according to the constructed dataset.

Multiple uncertainty metrics can promote each other in a complimentary from the perspectives of prediction results and feature representation. The details of the six uncertainty

metrics are in the appendix. Experimental results demonstrate the necessity of each uncertainty metric. For example, energy can estimate the physical properties and the outlier degrees of target samples. Still, energy is always dominated by the maximum logit and ignores the values of non-maximum logits. To mitigate this issue, we introduce entropy to consider output probability distributions carefully. Further, the relative difference in entropy values between unknown and common classes is minimal when a high number of target classes. To remedy this problem, we adopt confidence to measure the non-smooth distributions. Finally, the confidence of some samples is misplaced in poor-calibrated neural networks. We introduce consistency between multiple classifiers to calibrate the over-confidence values. Thus, the aggregated uncertainty metric would satisfy the following inequalities:

$$\begin{aligned} \mathbb{E}_{\mathbf{x}\sim p_y}\mathbf{z}_s(\mathbf{x}) &< \mathbb{E}_{\mathbf{x}\sim p_{\bar{y}s}}\mathbf{z}_s(\mathbf{x}) \\ \mathbb{E}_{\mathbf{x}\sim q_y}\mathbf{z}_t(\mathbf{x}) &< \mathbb{E}_{\mathbf{x}\sim q_{\bar{y}t}}\mathbf{z}_t(\mathbf{x}) \end{aligned} \tag{3}$$

The advantages of the semi-separated uncertainty decision-maker in disentangling open and common class data are three-fold. *Firstly*, it directly predicts whether a target instance belongs to an open class by feeding multiple uncertainty quantities without the manual thresholds and further without cross-validation. *Secondly*, it can adaptatively determine the importance of each uncertainty metric with less sensitivity. It also can cut them into multi-level decision rules and is more interpretable than other classifiers, such as neural networks, support vector machines, naive Bayes, and k-nearest neighbor. *Thirdly*, thanks to the feature selection function, it plays a vital role in automated uncertainty metric selection by Gini index minimization that is more robust to invalid uncertainty metrics for the current task because the indiscriminative and conflicting uncertainty metrics can be pruned.

3.4. Uncertainty Separation Loss

Semi-separated uncertainty decision-makers attempt to seek an optimal decision boundary between common and open classes, but this is hard to find. Since open-class samples intuitively have large uncertainties, we should better make the boundary more explicit by decreasing the uncertainty metrics of common class samples while keeping the open-class samples far from the common class samples. Thus we need an explicit optimization objective to encourage the uncertainty metrics to become more well-separated. Inspired by the pioneering work (Saito et al., 2020), we propose to draw a boundary between open and common samples with aggregated uncertainty $\mathbf{z}^{US}$ of above-mentioned uncertainty metrics. Note that the chief difference of our work with Saito et al. (2020) is that we consider the aggregated uncertainty. We denote by ρ the boundary and $|\mathbf{z}^{US}(\mathbf{x}) - \rho|$ the distance between the aggregated uncertainty and boundary. Through maximizing the distance, $\mathbf{z}^{US}(\mathbf{x})$ is far from ρ. We expect that the aggregated uncertainty of target open-class samples will be larger than ρ whereas, for the target common class samples, it will be reached zero. However, the boundary is ambiguous in many cases and can change due to domain shifts. Therefore, we introduce a tightening parameter $\gamma \geq 0$ to set a boundary margin that only penalties confident samples. The final form of the uncertainty separation loss is

$$\begin{aligned} &\mathcal{L}_{us} = \mathbb{E}_{\mathbf{x}\sim q}\mathcal{L}_{sus}(\mathbf{z}^{US}(\mathbf{x}))\,, \text{ where} \\ &\mathcal{L}_{sus}(\mathbf{z}^{US}(\mathbf{x})) = \begin{cases} -|\mathbf{z}^{US}(\mathbf{x}) - \rho| & |\mathbf{z}^{US}(\mathbf{x}) - \rho| > \gamma \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

in which ρ is set to the mean of $\mathbf{z}^{US}$ to avoid cross-validation and does empirically work well.

The principle behind US loss is to push the target open-class data towards their semantically close neighbors by encouraging high uncertainty predictions. When $|\mathbf{z}^{US}(\mathbf{x}) - \rho|$ is sufficiently large, the aggregated uncertainty metric likely satisfies the following inequalities:

$$\begin{aligned} \mathbb{E}_{\mathbf{x}\sim p_{\mathcal{Y}}} \mathbf{z}_s(\mathbf{x}) &\ll \mathbb{E}_{\mathbf{x}\sim p_{\bar{\mathcal{Y}}^s}} \mathbf{z}_s(\mathbf{x}) \\ \mathbb{E}_{\mathbf{x}\sim q_{\mathcal{Y}}} \mathbf{z}_t(\mathbf{x}) &\ll \mathbb{E}_{\mathbf{x}\sim q_{\bar{\mathcal{Y}}^t}} \mathbf{z}_t(\mathbf{x}) \end{aligned} \tag{5}$$

such that this loss promotes the semi-separated uncertainty decision-maker to separate the open-class and common class data easily.

3.5. Conditional-Weighted Adversarial Loss

Recall that the domain gap appeared as a critical challenge in UniDA, and it also influenced the reliability of uncertainty metrics. The vanilla adversarial loss (Ganin and Lempitsky, 2015) only supports matching the feature distributions for the whole label set, resulting in the class misalignment problem. We would better align the common label set solely between two domains such that we need a validated mechanism to exclude the open-class samples during adaptation. Following the uncertainty separation loss, we leverage the distance $|\mathbf{z}^{US}(\mathbf{x}) - \rho|$ between the aggregated uncertainty and boundary as the validated mechanism. Denote by D the adversarial domain discriminator and F the feature extractor. The conditional-weighted adversarial loss $\mathcal{L}_{ca}$ is formally defined as

$$\begin{aligned} \mathcal{L}_{ca} = &- \mathbb{E}_{\mathbf{x}\sim p} \mathbb{1}(|\mathbf{z}^{US}(\mathbf{x}) - \rho| < \gamma) \log D(F(\mathbf{x})) \\ &- \mathbb{E}_{\mathbf{x}\sim q} \mathbb{1}(|\mathbf{z}^{US}(\mathbf{x}) - \rho| < \gamma) \log(1 - D(F(\mathbf{x}))) \end{aligned} \tag{6}$$

where $\mathbb{1}$ is the indicator function. If a sample has remarkably low uncertainty, it can be viewed as a common class sample, participates in optimization, and excludes otherwise.

With well-established distance, $|\mathbf{z}^{US}(\mathbf{x}) - \rho|$, the adversarial domain discriminator D is confined to distinguish the source and target data in the common label set $\mathcal{Y}$. Unlike previous works (You et al., 2019; Fu et al., 2020; Lifshitz and Wolf, 2021) that consider the soft weight of each sample, we consider the hard weight that strictly excludes the open-class samples with the validated boundary ρ and the remarkable margin γ. Saito et al. (2020) used a deep clustering to construct the neighborhood clustering loss to make the target samples well-clustered and well-aligned to source samples. However, we cannot guarantee that the target data points can form clusters based on the proximity to Euclidean space.

Finally, we unify the uncertainty separation loss $\mathcal{L}_{us}$ and the conditional-weighted adversarial loss $\mathcal{L}_{ca}$ with the cross-entropy loss $\mathcal{L}_{cs}$ in a joint optimization problem:

$$\max_{D} \min_{F,G,G_i|_{i=1}} \mathcal{L}_{cs} + \lambda(\mathcal{L}_{us} - \mathcal{L}_{ca}) \tag{7}$$

where λ is a hyper-parameter to trade-off between discriminability and uncertainty. We also utilize the efficient gradient reversal layer (GRL) (Ganin and Lempitsky, 2015) to reverse the gradient between F and D to optimize the model in an end-to-end solution.

4. Experiments

4.1. Setup

We conduct experiments on **Office-31** (Saenko et al., 2010), **Office-Home** (Venkateswara et al., 2017), **VisDA-2017** (Peng et al., 2017), and **DomainNet** (Peng et al., 2019). For the first three datasets, we follow the setup of You et al. (2019). For **DomainNet**, we follow the setup of Fu et al. (2020). The more details of datasets are in the appendix.

We compare our designed **SNAIL** algorithm with state-of-the-art methods: (1) Source only: **ResNet-50** (He et al., 2016)); (2) Closed-set domain adaptation: Domain Adversarial Network (**DANN**) (Ganin and Lempitsky, 2015)), Residual Transfer Networks (**RTN**) (Long et al., 2016); (3) Partial domain adaptation: Importance Weighted Adversarial Nets (**IWAN**) (Zhang et al., 2018), Partial Adversarial Domain Adaptation (**PADA**) (Cao et al., 2018b); (4) Open-set domain adaptation: Assign-and-Transform-Iteratively (**ATI**) (Busto and Gall, 2017), Open Set Back-Propagation (**OSBP**) (Saito et al., 2018); (5) Universal domain adaptation: Universal Adaptation Network (**UAN**) (You et al., 2019), Calibrated Multiple Uncertainties (**CMU**) (Fu et al., 2020), Domain Adaptative Neighborhood Clustering via Entropy optimization (**DANCE**) (Saito et al., 2020), Domain Consensus Clustering (**DCC**) (Li et al., 2021), Sample Selection Approach (**SSA**) (Lifshitz and Wolf, 2021). We report the accuracy and H-score. H-score is the harmonic mean of the accuracy of the common and open classes (Fu et al., 2020), serving more credible evaluation.

Our method enjoys a minimal number of hyperparameters. We implement our algorithm in Pytorch with ResNet-50 (He et al., 2016) backbone pretrained on ImageNet (Deng et al., 2009). We set λ in Eq. (7) as 0.05, γ in Eq. (4) and Eq. (6) as 0.15. We use the same batch size, learning rate, and epoch for all the tasks. We set ensemble classifier numbers m to 2 to empirically calibrate the multiple uncertainty metrics. Since it is insensitive to set the ratio $n\%$, we set the ratio $n\%$ to 30% for all the tasks. In such a way, the constructed dataset has a certain amount of data and accurate labels, such that we can use this reliable dataset to generate an effective semi-separated uncertainty decision-maker.

Table 1: Accuracy (%) on **Office-31** for universal domain adaptation (ResNet-50).

Method	A→W		D→W		W→D		A→D		D→A		W→A		Avg	
	Acc	H-score	Acc	H-score	Acc	H-score	Acc	H-score	Acc	H-score	Acc	H-score	Acc	H-score
ResNet (He et al., 2016)	75.9	47.9	89.6	54.9	90.9	55.6	80.5	49.8	78.8	48.5	81.4	49.0	82.9	50.9
DANN (Ganin and Lempitsky, 2015)	80.7	48.8	80.9	52.7	88.1	54.9	82.7	50.2	74.8	47.7	83.5	49.3	81.8	50.6
RTN (Long et al., 2016)	85.7	50.2	87.8	54.7	88.9	55.2	82.7	50.2	74.6	47.7	83.3	49.3	83.8	51.2
IWAN (Zhang et al., 2018)	85.3	50.1	90.1	54.1	90.0	55.4	84.3	50.6	84.2	49.7	86.3	49.8	86.7	51.6
PADA (Cao et al., 2018b)	85.4	49.7	79.3	52.6	90.9	55.6	81.7	50.0	55.3	42.9	82.6	49.2	79.2	50.0
ATI (Busto and Gall, 2017)	79.4	48.6	92.6	55.0	90.1	55.5	84.4	50.5	78.9	48.5	81.6	49.0	84.5	51.2
OSBP (Saito et al., 2018)	66.1	50.2	73.6	55.5	85.6	57.2	72.9	51.1	47.4	49.8	60.5	50.2	67.7	52.3
UAN (You et al., 2019)	85.6	58.6	94.8	70.6	98.0	71.4	86.5	59.7	85.5	60.1	85.1	60.3	89.2	63.5
CMU (Fu et al., 2020)	86.9	67.3	95.7	79.3	98.0	80.4	89.1	68.1	88.4	71.4	88.6	72.2	91.1	73.1
DCC (Li et al., 2021)	91.7	78.5	94.5	79.3	96.2	88.6	**93.7**	**88.5**	90.4	70.2	92.0	75.9	93.1	80.2
DANCE (Saito et al., 2020)	92.8	67.4	97.8	89.9	97.7	90.7	91.6	70.8	**92.2**	79.1	91.4	71.9	93.9	78.3
SSA (Lifshitz and Wolf, 2021)	90.1	-	95.3	-	98.2	-	90.6	-	90.0	-	90.5	-	92.4	-
SNAIL (Ours)	**93.2**	**80.6**	**98.7**	**96.5**	**99.1**	**94.2**	93.5	82.4	**92.2**	**86.4**	**92.1**	**84.2**	**94.8**	**87.4**

4.2. Results

Table 1 reports the results on Office-31. Our algorithm significantly outperforms all compared methods regarding classification accuracy and open-class detection. It is worth noting that our

Table 2: Accuracy (%) on **Office-Home** for universal domain adaptation (ResNet-50).

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
ResNet (He et al., 2016)	59.4	76.6	87.5	69.9	71.1	81.7	73.7	56.3	86.1	78.7	59.2	78.6	73.2
DANN (Ganin and Lempitsky, 2015)	56.2	81.7	86.9	68.7	73.4	83.8	69.9	56.8	85.8	79.4	57.3	78.3	73.2
RTN (Long et al., 2016)	50.5	77.8	86.9	65.1	73.4	85.1	67.9	45.2	85.5	79.2	55.6	78.8	70.9
IWAN (Zhang et al., 2018)	52.6	81.4	86.5	70.6	71.0	85.3	74.9	57.3	85.1	77.5	59.7	78.9	73.4
PADA (Cao et al., 2018b)	39.6	69.4	76.3	62.6	67.4	77.5	48.4	35.8	79.6	75.9	44.5	78.1	62.9
ATI (Busto and Gall, 2017)	52.9	80.4	85.9	71.1	72.4	84.4	74.3	57.8	85.6	76.1	60.2	78.4	73.3
OSBP (Saito et al., 2018)	47.8	60.9	76.8	59.2	61.6	74.3	61.7	44.5	79.3	70.6	55.0	75.2	63.9
UAN (You et al., 2019)	63.0	82.8	87.9	76.9	78.7	85.4	78.2	58.6	86.8	83.4	63.2	79.4	77.0
CMU (Fu et al., 2020)	63.5	83.8	88.9	77.7	79.4	86.9	78.6	59.3	88.3	84.1	64.6	81.4	78.0
DCC (Li et al., 2021)	54.2	47.5	57.5	83.8	71.6	86.2	63.7	65.0	75.2	**85.5**	**78.2**	82.6	70.9
DANCE (Saito et al., 2020)	64.1	84.3	91.2	**84.3**	78.3	89.4	83.4	63.6	91.4	83.3	63.9	86.9	80.4
SSA (Lifshitz and Wolf, 2021)	65.1	**86.4**	91.4	79.5	**84.9**	89.6	82.0	56.8	89.8	79.5	61.5	**89.0**	79.6
SNAIL (Ours)	**68.3**	83.6	**93.3**	82.9	77.2	**91.1**	**83.7**	**68.0**	**92.5**	82.8	67.3	87.6	**81.5**

Table 3: H-score (%) on **Office-Home** for universal domain adaptation (ResNet-50).

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
ResNet (He et al., 2016)	44.7	48.0	50.1	46.6	46.9	49.0	47.5	43.2	50.2	48.5	44.8	48.4	47.3
DANN (Ganin and Lempitsky, 2015)	42.4	48.0	48.9	45.5	46.5	48.4	45.8	42.6	48.7	47.6	42.7	47.4	46.2
RTN (Long et al., 2016)	38.4	44.7	45.7	42.6	44.1	45.5	42.6	36.8	45.5	44.6	39.8	44.5	42.9
IWAN (Zhang et al., 2018)	40.5	47.0	47.8	45.0	45.1	47.6	45.8	41.4	47.6	46.3	42.5	46.5	45.3
PADA (Cao et al., 2018b)	34.1	41.9	44.1	40.6	41.5	44.0	37.0	32.6	44.2	43.1	35.8	43.4	40.2
ATI (Busto and Gall, 2017)	39.9	45.8	46.6	44.1	44.4	46.6	44.7	41.2	46.6	45.1	41.8	45.5	44.4
OSBP (Saito et al., 2018)	39.6	45.1	46.2	45.7	45.2	46.8	45.3	40.5	45.8	45.1	41.6	46.9	44.5
UAN (You et al., 2019)	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
DANCE (Saito et al., 2020)	35.9	29.3	35.2	42.6	18.0	29.3	50.2	45.4	41.1	19.8	38.8	52.6	36.5
SNAIL (Ours)	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

algorithm has a significant improvement in open-class detection. For example, it outperforms DANCE by almost 20% H-score and ResNet by almost 40% H-score. Table 2 and Table 3 reports the accuracy and H-score on the tough Office-Home, respectively. Our algorithm also achieves state-of-the-art accuracy on complex tasks, making a remarkable performance boost. Table 4 summarizes the results on the two most challenging and large-scale datasets: DomainNet, and VisDA-2017. Our algorithm consistently shows large margins on almost all tasks compared to all baselines. The above results validate that our algorithm performs well and has great application potential in large-scale datasets with large distribution shifts.

CMU (Fu et al., 2020) performs inadequately on open-class detection and per-class accuracy because of the following two points. Firstly, the manual thresholds make CMU inadequate to detect open-class samples because the thresholds are sensitive to set. Secondly, forcibly aligning (using soft weights) the distributions causes the mistaken alignment between the open and common classes. Our algorithm outperforms CMU on H-score by a large margin (25% in VisDA). The higher H-scores demonstrate that the semi-separated uncertainty decision-maker and the uncertainty separation loss enable higher-quality transferability measures to detect open classes more accurately. The higher accuracy verifies that the conditional-weighted adversarial loss encourages the weak alignment (hard weights) of distributions to promote common-class adaptation.

4.3. Analysis

Ablation Study. We dissect the efficacy of the proposed method by evaluating variants of SNAIL on Office-31 as shown in Table 5. **(1)** SNAIL w/o semi-separated uncertainty

Table 4: Performance on **DomainNet** and **VisDA**.

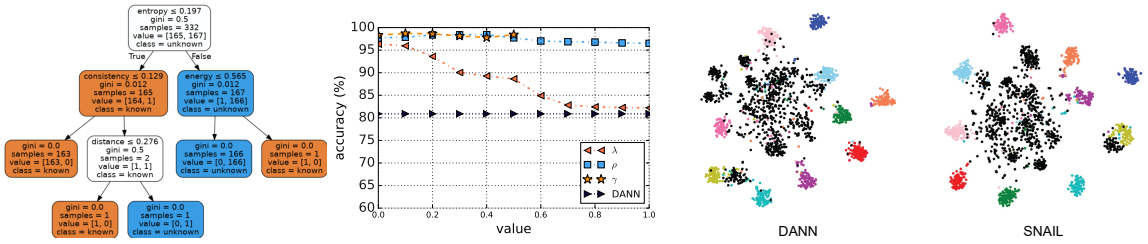
Method	DomainNet (H-score)							VisDA	
	P2R	R2P	P2S	S2P	R2S	S2R	Avg	Acc	H-score
ResNet (He et al., 2016)	30.1	28.3	27.0	27.0	26.9	29.7	28.2	52.8	25.4
DANN (Ganin and Lempitsky, 2015)	31.2	29.3	27.8	27.8	27.8	30.8	29.1	52.9	25.7
RTN (Long et al., 2016)	32.3	30.3	28.7	28.7	28.6	31.9	30.1	53.9	26.0
IWAN (Zhang et al., 2018)	35.4	33.0	31.2	31.2	31.1	34.9	32.8	58.7	27.6
PADA (Cao et al., 2018b)	28.9	27.3	26.0	26.0	26.0	28.6	27.2	45.0	23.1
ATI (Busto and Gall, 2017)	32.6	30.6	29.0	29.0	28.9	32.2	30.4	54.8	26.3
OSBP (Saito et al., 2018)	33.6	33.0	30.6	30.5	30.6	33.7	32.0	30.3	27.3
UAN (You et al., 2019)	41.9	43.6	39.1	39.0	38.7	43.7	41.0	60.8	30.5
CMU (Fu et al., 2020)	50.8	52.2	45.1	44.8	**45.6**	51.0	48.3	61.4	34.6
SNAIL (Ours)	**60.6**	**52.8**	**47.0**	**45.2**	42.4	**57.0**	**50.8**	**64.9**	**59.8**

Table 5: Ablation study on **Office-31** (ResNet-50).

Method	D→W		A→D		W→A		Avg (six tasks)	
	Acc	H-score	Acc	H-score	Acc	H-score	Acc	H-score
SNAIL	**98.7**	**96.5**	**93.5**	**82.4**	**92.1**	**84.2**	**94.8**	**87.4**
w/o Uncertainty Decision-maker	97.4	84.7	88.7	69.8	90.4	70.5	92.4↓	76.6↓
w/o Uncertainty Separation Loss	95.3	92.1	83.5	74.2	82.4	80.9	86.4↓	82.4↓
w/o Adversarial Loss	97.3	86.2	88.9	76.9	91.0	79.5	91.9↓	80.2↓
w/o Ensemble Classifiers	96.7	79.3	91.3	81.4	91.6	76.0	92.9↓	78.4↓
w/o Entropy	98.0	93.5	87.2	75.1	90.0	80.0	92.2↓	84.0↓
w/o Consistency	98.4	96.3	88.0	75.7	89.9	81.5	92.3↓	83.4↓
w/o Confidence	98.2	95.9	88.2	74.1	89.7	82.6	91.7↓	84.6↓
w/o Distance	98.5	95.1	88.0	77.6	90.0	84.0	92.3↓	84.8↓
w/o Similarity	97.9	95.0	88.7	77.8	89.0	83.8	91.0↓	83.6↓
w/o Energy	98.5	95.3	87.6	75.3	89.2	83.7	91.9↓	83.3↓

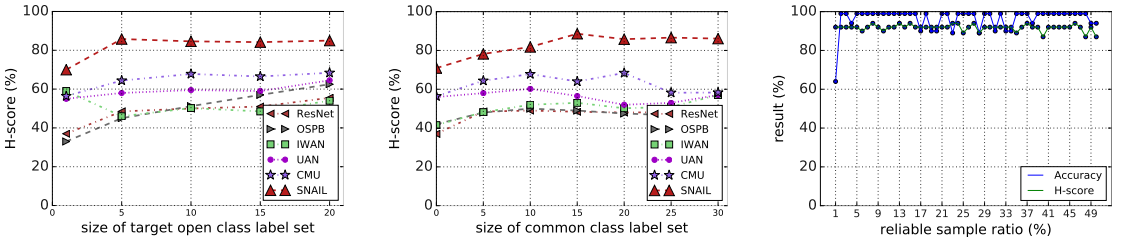
decision-maker is the variant without using the semi-separated uncertainty decision-maker but using threshold parameters to split the overall uncertainty by Eq (1). SNAIL remarkably outperforms it, indicating the contribution of the domain-level transferability criterion with a global view. Figure 4(a) visualizes the learned semi-separated uncertainty decision-maker on the A→D task, in which it adaptatively assigns the weights of four uncertainties and selects out two redundant uncertainty metrics. **(2)** SNAIL w/o uncertainty separation loss and w/o conditional-weighted adversarial loss are the variants without the corresponding objective, respectively. **(3)** SNAIL w/o ensemble classifiers is the variant without using multiple ensemble classifiers but only using the target classifier to calculate multiple uncertainties. **(4)** SNAIL w/o each uncertainty metric is the variant without using the corresponding uncertainty metric when building the semi-separated uncertainty decision-makers. As shown in Table 5, SNAIL remarkably exceeds these variants, proving the necessity of the corresponding modules. Figure 4(c) further verifies that the conditional-weighted adversarial loss promotes learning more discriminative domain-invariant features than the vanilla adversarial loss.

Varying Size of $\bar{\mathcal{Y}}^t$ and $\bar{\mathcal{Y}}^s$. Following UAN and DANCE, with fixed $|\mathcal{Y}^s \cap \mathcal{Y}^t|$ and $|\mathcal{Y}^s \cup \mathcal{Y}^t|$, we examine the H-score with numerous sizes of open classes $\bar{\mathcal{Y}}^t$ ($\bar{\mathcal{Y}}^s$ also changes correspondingly) on task A→D in Office-31. As shown in Figure 5(a), SNAIL consistently achieves the better performance than all the compared methods by large margins with various $|\bar{\mathcal{Y}}^t|$, which demonstrate that SNAIL has robustness and generalizability to diverse $\bar{\mathcal{Y}}^t$ and $\bar{\mathcal{Y}}^s$. In particular, when $|\bar{\mathcal{Y}}^t|$ is large (over 15), SNAIL also obtain the best performance.



(a) Semi-separated uncertainty decision-maker (b) Varying ρ, γ, λ in Eq. (4) and (7) (c) t-SNE plots of target examples

Figure 4: (a) The visualization of the semi-separated uncertainty decision-maker on A→D. (b) Varying the value of three hyperparameters. (c) t-SNE (Donahue et al., 2014) plots of target examples (Best viewed in color). Black dots represent open-class examples, while other colors represent common class examples.



(a) H-score w.r.t $\bar{\mathcal{Y}}^t$ (b) H-score w.r.t $\mathcal{Y}$ (c) Results w.r.t $n\%$ samples

Figure 5: (a)(b) H-score with respect to $\bar{\mathcal{Y}}^t$ and $\mathcal{Y}$. (c) Accuracy and H-score of various reliable instance ratios.

Varying Size of Common Label Set $\mathcal{Y}$. We investigate another dimension of SNAIL by ranging the size of $|\mathcal{Y}|$ and fix $|\mathcal{Y}^s \cup \mathcal{Y}^t|$ on task A→D in Office-31. Following UAN and DANCE, we let $|\bar{\mathcal{Y}}^t| = |\bar{\mathcal{Y}}^s| + 1$ and range $|\mathcal{Y}|$ from 0 to 31. As shown in Figure 5(b), SNAIL consistently gets better performance than all the compared methods by large margins on all the sizes of $\mathcal{Y}$. In particular, when $|\mathcal{Y}|$ is near zero, the performance of SNAIL is more stable than previous methods, indicating its robustness in the open and dynamic task environment. When $|\mathcal{Y}|$ tends to 31, the task is turning into closed set domain adaptation, and the performance of SNAIL is comparable with DANN's performance.

The Sensitivity to Hyperparameters. Figure 4(b) reveals the sensitivity to hyperparameters (γ, ρ, and λ) on the D→W task. We can see that when we increase the value of ρ, more samples will be decided as common classes, then the accuracy of common class decreases. Figure 5(c) shows that it is insensitive to the ratio $n\%$ of reliable target samples in training the uncertainty decision-maker. One may claim that the semi-separated uncertainty decision-maker is invalid when the target label set does not contain open class. Our algorithm still achieves the best accuracy when the open-class size is near zero as shown in Figure 5(a).

5. Conclusion

This paper presented a novel framework: semi-separated uncertainty adversarial learning. We designed a novel semi-separated uncertainty decision-maker to free the sensitive thresholds of multiple uncertainty metrics estimated and calibrated by the domain adaptation model. We proposed the uncertainty separation loss and the conditional-weighted adversarial loss. Both can simultaneously alleviate the category gap and the distribution shift. Comprehensive experiments have demonstrated that our framework achieves a balanced capacity to detect open classes and classify common class examples accurately. Our framework is simple and orthogonal to other algorithms. One can integrate our semi-separated uncertainty decision-maker into their UniDA to abolish the manual threshold parameters.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62176139) and the Major Basic Research Project of the Natural Science Foundation of Shandong Province (ZR2021ZD15).

References

Mahsa Baktashmotlagh, Masoud Faraki, Tom Drummond, and Mathieu Salzmann. Learning factorized representations for open-set domain adaptation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984. ISBN 0-534-98053-8.

Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 754–763. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.88.

Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Partial transfer learning with selective adversarial networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2724–2732. IEEE Computer Society, 2018a. doi: 10.1109/CVPR.2018.00288.

Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*, volume 11212 of *Lecture Notes in Computer Science*, pages 139–155. Springer, 2018b. doi: 10.1007/978-3-030-01237-3_9.

Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation. In *IEEE Conference on Computer Vision*

and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pages 2985–2994. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00310.

Martin Cihak, Asli Demirgüç-Kunt, Erik Feyen, and Ross Levine. Benchmarking financial systems around the world, 2012.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206848.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31th International Conference on Machine Learning*, pages 647–655, 2014.

Bo Fu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Learning to detect open classes for universal domain adaptation. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XV*, volume 12360 of *Lecture Notes in Computer Science*, pages 567–583. Springer, 2020. doi: 10.1007/978-3-030-58555-6_34.

Yaroslav Ganin and Victor S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1180–1189, 2015.

Zhongyi Han, Xian-Jin Gui, Chaoran Cui, and Yilong Yin. Towards accurate and robust domain adaptation under noisy environments. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 2269–2276. ijcai.org, 2020. doi: 10.24963/ijcai.2020/314.

Zhongyi Han, Haoliang Sun, and Yilong Yin. Learning transferable parameters for unsupervised domain adaptation. *CoRR*, abs/2108.06129, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Kilian Hendrickx, Lorenzo Perini, Dries Van der Plas, Wannes Meert, and Jesse Davis. Machine learning with a reject option: A survey. *CoRR*, abs/2107.11277, 2021.

Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Scholkopf. Correcting sample selection bias by unlabeled data. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pages 601–608, Cambridge, MA, USA, December 2006. MIT Press.

Mengmeng Jing, Jingjing Li, Lei Zhu, Zhengming Ding, Ke Lu, and Yang Yang. Balanced open set domain adaptation via centroid alignment. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Event, February 2-9, 2021*, pages 8013–8020. AAAI Press, 2021.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.

Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 9757–9766. Computer Vision Foundation / IEEE, 2021.

Omri Lifshitz and Lior Wolf. Sample selection for universal domain adaptation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Event, February 2-9, 2021*, pages 8592–8600. AAAI Press, 2021.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 97–105, Lille, France, July 2015. JMLR.org.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 1647–1657, 2018.

Yingwei Pan, Ting Yao, Yehao Li, Chong-Wah Ngo, and Tao Mei. Exploring category-agnostic clusters for open-set domain adaptation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 13864–13872. IEEE, 2020. doi: 10.1109/CVPR42600.2020.01388.

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

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 1406–1415. IEEE, 2019. doi: 10.1109/ICCV.2019.00149.

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

Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2988–2997, 2017.

Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part V*, volume 11209 of *Lecture Notes in Computer Science*, pages 156–171. Springer, 2018. doi: 10.1007/978-3-030-01228-1_10.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 2110–2118, 2016.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014. ISBN 978-1-10-705713-5.

Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, December 2008. ISSN 0020-3157, 1572-9052. doi: 10/d9t5g6.

Naveen Venkat, Jogendra Nath Kundu, Durgesh Kumar Singh, Ambareesh Revanur, and Venkatesh Babu R. Your classifier can secretly suffice multi-source domain adaptation. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.

Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Universal domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2720–2729. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00283.

Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8156–8164. IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00851.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I. Jordan. Bridging theory and algorithm for domain adaptation. In *Proceedings of the 36th International Conference on Machine Learning*, pages 7404–7413, 2019.