CONFIDENCE SCORE WEIGHTING ADAPTATION FOR SOURCE-FREE UNSUPERVISED DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised domain adaptation (UDA) aims to achieve high performance within the unlabeled target domain by leveraging the labeled source domain. Source-free UDA, which is a more challenging UDA task, can access the pre-trained model within the source domain. The pre-trained model, however, provides a noisy pseudo-label; thus, source-free UDA requires robust training. In this study, we propose a Confidence score Weighting Adaptation (CoWA), which is a simple yet effective source-free UDA method. CoWA utilizes the Joint Model-Data Structure (JMDS) confidence score designed for source-free UDA as a sample-wise weight. As components of CoWA, we introduce Suppressed Cross Entropy (SCE) loss and a weight mixup to robustly leverage the low-confidence samples. Experiment results show that CoWA achieves a superior performance compared to other source-free UDA methods on various UDA benchmarks including open-set and partial-set domain adaptation. Furthermore, on several benchmarks, CoWA surpasses state-of-the-art conventional UDA methods that use labeled source domain data.

1 INTRODUCTION

In recent years, Deep Neural Networks (DNNs) (LeCun et al., 2015) have successfully demonstrated a high performance in various applications across diverse fields. However, if the distribution of the training and test data are different, a significant performance degradation occurs, which is known as a domain shift (Pan & Yang, 2009). Unsupervised Domain Adaptation (UDA) aims at mitigating the domain shift problem. It makes use of both fully annotated source domain data and unlabeled target domain data assuming that the two domains have different data distributions. The objective of a UDA task is to obtain a high target performance using the two domains jointly without accessing the target label. UDA has been developed through many different methods, most of which can be classified into two main paradigms: minimizing the discrepancy between the source and target domains (Kang et al., 2019; Long et al., 2015; 2017) and using adversarial training to obtain domain-invariant features (Ganin & Lempitsky, 2015; Tzeng et al., 2017; Xu et al., 2020).

All conventional UDA methods assume the availability of both the source data and their corresponding labels. However, this may be impractical for the following reasons. First, increasing concerns regarding the privacy and security of their data can force companies to release only the model and not the data. Second, many resources are required to train a model when the number of source data are much greater than the number of target data. For these reasons, source-free UDA has recently been studied (Li et al., 2020; Liang et al., 2020a). Source-free UDA assumes that we can only access a model pre-trained using the labeled source domain data instead of accessing the source data itself. In other words, the aforementioned UDA paradigms cannot be applied to source-free UDA.

The pre-trained source model provides noisy pseudo-labels for the target data. Existing source-free UDA methods regard noisy pseudo-labels as a minor problem. For example, Liang et al. (2020a) proposed naive self-supervised pseudo-labeling which is a variant of weighted k-means clustering. This approach utilizes a weighted mean using the prediction of the model to obtain new centroids. Although this can help mitigate the noisy pseudo-label problem, it has little overall effect. To the best of our knowledge, this is the first work to regard the noisy pseudo-label problem as a major problem in source-free UDA and propose a method to solve it.

We propose a novel yet simple source-free UDA method, called Confidence score Weighting Adaptation (CoWA) which utilizes a confidence score as a sample weight. CoWA is motivated based on

the following evidence. First, by definition, samples with high confidence scores are more likely to have correct pseudo-labels than samples with low-confidence scores (Geifman et al., 2018). Second, each sample has a different confidence score for its pseudo-label. Third, a model is trained robustly with noisy labels when easy samples are of higher importance (Ren et al., 2018). In experiments, we confirm that sample reweighting with a confidence score shows better performance than training without sample reweighting. Moreover, the performance is increased when the reliability of a confidence score increases.

CoWA uses a Joint Model-Data Structure (JMDS) confidence score and Suppressed Cross Entropy (SCE) loss to jointly utilize the information of the model and data structure. The JMDS score is computed offline using two probabilities: model- and data structure-based probabilities. To compensate for the offline property of the JMDS, the SCE loss considers the online model knowledge. Moreover, to robustly participate in low-confidence samples during training, we introduce a weight mixup, which is a variant of a mixup (Zhang et al., 2017). It mixes the sample weight and considers the confidence score of the mixed images. Finally, we add a data augmentation to CoWA for jointly learning the information of the global- and local-views of the raw image.

We evaluated CoWA on various public UDA benchmarks namely, Office-31 (Saenko et al., 2010), Office-Home (Venkateswara et al., 2017), and VISDA-2017 (Peng et al., 2017). CoWA achieved the best performance on source-free UDA settings for all three benchmarks. Although CoWA is a source-free UDA method, it obtained the best performance among all UDA methods, including conventional UDA methods that directly use fully annotated source data, on the medium-sized Office-Home and large-sized VISDA-2017 datasets. Moreover, CoWA achieved a state-of-the-art performance not only under general UDA setting, but also in open-set and partial-set UDA with little modification during implementation. Ablation studies demonstrated the effectiveness of the each CoWA component.

We summarize the contributions of our study as follows:

- To the best of our knowledge, this study is the first work to overcome the noisy pseudo-label problem of the pre-trained model in source-free UDA.
- We propose a novel source-free UDA method, CoWA, using the JMDS score designed for source-free UDA as a sample weight to robustly learn with noisy pseudo-labels.
- We achieved a state-of-the-art performance on source-free UDA settings for three UDA benchmarks using CoWA, and even obtained the best performance in terms of the average accuracy among all UDA methods on the Office-Home, and VisDA-2017.

2 Related work

2.1 Unsupervised domain adaptation

Conventional UDA is accessible to labeled source data during training. Therefore, it can extract the full source domain information. Many methods have been proposed to obtain domain-invariant features. Some methods (Kang et al., 2019; Long et al., 2015; 2017; Saito et al., 2018; Sun & Saenko, 2016; Tzeng et al., 2014) minimize the discrepancies between domains. Long et al. (2015) used Maximum Mean Discrepancy (MMD), whereas Kang et al. (2019) applied Contrastive Domain Discrepancy (CDD) to minimize the intra-class discrepancy and maximize the inter-class margin. Another approach to obtaining domain-invariant features is to use adversarial loss (Ganin & Lempitsky, 2015; Long et al., 2018; Pei et al., 2018; Tzeng et al., 2017; Zhao et al., 2018). Ganin & Lempitsky (2015), motivated by Goodfellow et al. (2014), first introduced the adversarial approach that learns the feature extractor to fool the domain discriminator which domain the feature comes from. Pei et al. (2018) used multiple domain discriminators corresponding to each class to provide more class-aware information in domain-invariant features.

Source-free domain adaptation is a more challenging UDA setting than the conventional UDA setting, where only a source-trained model can be used. Thus, the existing methods are inapplicable to source-free UDA. Collaborative Class Conditional Generative Adversarial Networks (3C-GAN) (Li et al., 2020) are based on a time-consuming target-style image generation through a conditional GAN, which requires auxiliary networks. Source HypOthesis Transfer (SHOT) (Liang et al., 2020a) matches the target feature to a fixed pre-trained source classifier fine-tuning the feature extractor. SHOT

applies self-supervised pseudo-labeling (SSPL) and uses information-maximization loss to balance the pseudo-label and avoid a trivial solution. However, SHOT cannot fully extract the knowledge of the target feature structure because it uses SSPL which does not consider the covariance of each dimension on the feature space and cluster density. In this study, we use Gaussian Mixture Modeling (GMM) on the feature space to extract the knowledge of the target data structure.

2.2 LEARNING FROM NOISY LABELS

Learning from noisy labels has become an important task in modern deep learning approaches. As described by Mirzasoleiman et al. (2020), such learning has been improved by various approaches: noise transition matrix estimation (Goldberger & Ben-Reuven, 2017; Patrini et al., 2017), a robust loss function design (Ghosh et al., 2017; Van Rooyen et al., 2015; Wang et al., 2019), label correction (Ma et al., 2018; Reed et al., 2014; Tanaka et al., 2018), regularization techniques (Cao et al., 2020; Zhang et al., 2017; 2020), and sample selection or sample reweighting (Chen et al., 2019; Han et al., 2018; Ren et al., 2018). However, learning from noisy labels leads to supervised learning. In our source-free UDA setting, we cannot directly use these methods. To reflect the characteristics of each sample well, we adopted a reweighting training sample scheme for our proposed method.

Sample reweighting has long been actively studied for a long time in various fields (Freund & Schapire, 1997; Kahn & Marshall, 1953; Lin et al., 2017; Malisiewicz et al., 2011). In general, hard examples with high importance are used to efficiently train the model. However, this is beneficial when a clean label is guaranteed. If noise or outliers exist, the training of easy samples with high importance is more beneficial. Self-paced learning (Kumar et al., 2010) obtains sample weights by encouraging easier sample learning first. Gong et al. (2016) theoretically showed that self-paced learning is robust to noisy data. In our study, we leverage a confidence score to detect an easy sample and use it as a sample weight.

2.3 CONFIDENCE SCORE

We deal with a confidence score for a non-Bayesian multi-class classification problem. Geifman et al. (2018) divided confidence scores into two main tasks; ordinal ranking and probability calibration. Ordinal ranking is used for selective classification (Geifman & El-Yaniv, 2017; Lakshminarayanan et al., 2016; Mandelbaum & Weinshall, 2017; Nair et al., 2020), which is a task that classifies samples according to their confidence level for labels to avoid low-confidence samples during training. Probability calibration (Guo et al., 2017; Heo et al., 2018; Kumar et al., 2018; Seo et al., 2018) aims at providing a score that accurately estimates the true correctness likelihood. In this study, we focus on ordinal ranking because our goal is to obtain high accuracy on the target data and not to obtain a calibrated score. We present for the first time source-free UDA method using a confidence score.

Geifman et al. (2018) proposed a consensus on how to measure the performance of the confidence score of ordinal ranking. They introduced a risk-coverage curve and proposed a normalized metric, *i.e.*, excess Area Under the Risk-Coverage Curve (E-AURC), as a measurement tool for the confidence score function of a label. In addition, Ding et al. (2020) compared the Area Under Receiver Operating Characteristic curve (AUROC), Area Under Precision-Recall curve (AUPR), and AURC instead of E-AURC. They claimed that AURC is the only reliable measurement when the underlying models are the same. Based on AURC with 0/1 loss, we compared our proposed confidence score with the previously used confidence scores.

3 CONFIDENCE SCORE WEIGHTING ADAPTATION (COWA)

In this section, we describe our method, Confidence score Weighting Adaptation (CoWA), for source-free domain adaptation. CoWA consists of Joint Model-Data Structure (JMDS) score, Suppressed Cross Entropy (SCE) loss, weight mixup, and data augmentation. An overview of CoWA is shown in Figure 1. We performed a K-way image classification task. In source-free UDA, we can only use the target data $X_t = \{x_i^t\}_{i=1}^{n_t}$ whose corresponding ground truth label, $Y_t = \{y_i^t\}_{i=1}^{n_t}$, which is inaccessible during the learning stage. A pre-trained model M is trained using labeled source data $X_s = \{x_i^s.y_i^s\}_{i=1}^{n_s}$. Here, M is composed of a feature extractor $f : X_t \to \mathbb{R}^d$ and classifier $g : \mathbb{R}^d \to \mathbb{R}^K$ where d is the dimension of the feature space. We aim to obtain a good target model M_t by fine-tuning M using a pseudo-label $\hat{Y}_t = \{\hat{y}_i^t\}_{i=1}^{n_t}$.

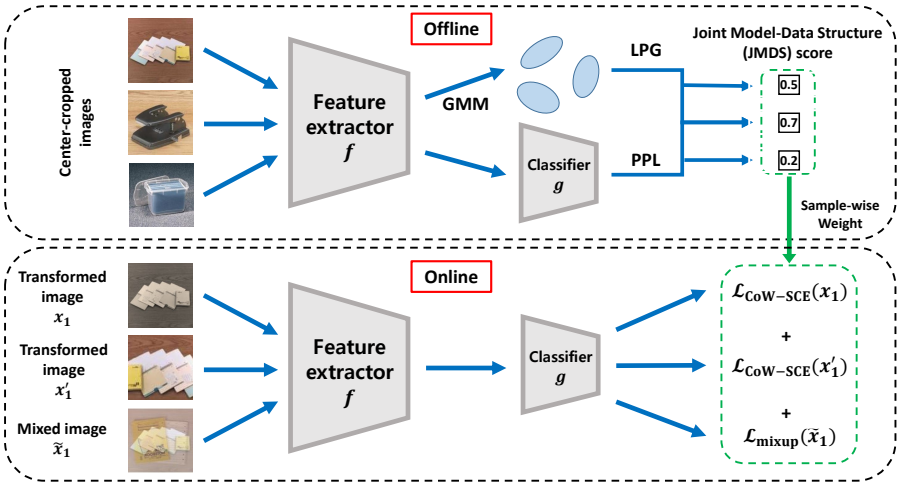


Figure 1: An overview of CoWA. During the evaluation, we compute JMDS scores for all features. Then, during training, we leverage the JMDS scores with SCE loss, weight mixup, and data augmentation to update the model M.

3.1 Joint Model-Data Structure (JMDS) score

Following Geifman et al. (2018), we define the confidence score function $\kappa(x_i, \hat{y}_i)$ for ordinal ranking. The confidence score function $\kappa(x_i, \hat{y}_i)$ should give a high score for more likely correct samples.

$$\kappa(x_i, \hat{y}_i) \leq \kappa(x_j, \hat{y}_j) \Rightarrow \Pr[\hat{y}_i = y_i] \leq \Pr[\hat{y}_j = y_j] \text{ with a high probability of } 1 - \delta, \tag{1}$$

where $0 \leq \delta \leq 1$. If $\kappa_1(\cdot)$ is a better confidence score function than $\kappa_2(\cdot)$, then $\delta_1 < \delta_2$.

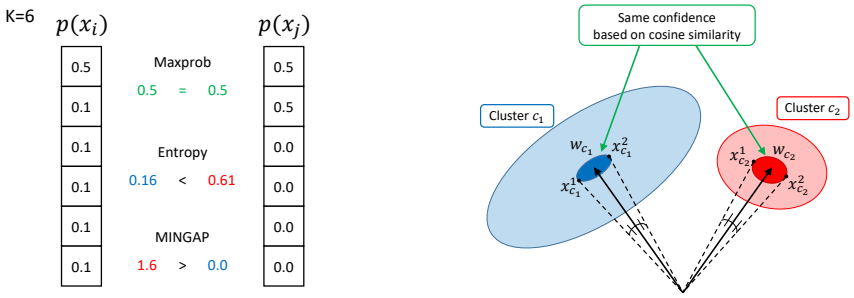
Many previous studies have used confidence scores to filter out low-confidence samples. However, as shown in Figure 2, commonly used confidence scores namely, the maximum probability (Maxprob), (negative) entropy, and cosine similarity (Kang et al., 2019; Mandelbaum & Weinshall, 2017) have limitations. We define the scores to be within [0, 1]:

$$p_M(x_i^t) = \text{softmax}\big(g(f(x_i^t))\big) \text{ where } \text{softmax}(z)_c = \frac{e^{z_c}}{\Sigma_{c'=1}^{K} e^{z_{c'}}}, \ \text{Maxprob}(x_i^t) = \max_c p_M(x_i^t)_c,$$
$$\text{Entropy}(x_i^t) = 1 + \frac{\Sigma_{c=1}^{K} p_M(x_i^t)_c \log p_M(x_i^t)_c}{\log K}, \ \text{Cosine}(x_i^t) = \frac{1}{2}\Big(1 + \frac{\langle x_i^t, C_{\hat{y}_i^t} \rangle}{\|x_i^t\| \|C_{\hat{y}_i^t}\|}\Big), \tag{2}$$

where p_M is the model prediction and $C_{\hat{y}_i^t}$ is the feature cluster center of the class $\hat{y}_i^t$. Figure 2a shows the limitations of the Maxprob and entropy scores. Because we use noisy pseudo-labels, it is desirable for easy samples to obtain high confidence scores (Ren et al., 2018). Based on the given probability, the Maxprob and entropy scores do not reflect the confidence properly, as shown in Figure 2a. Figure 2b shows the weakness of the cosine score. The dark regions in Figure 2b have the same confidence levels based on the cosine score. However, when the data structure is considered, the dark blue region should have a higher confidence level than the dark red region.

We need to obtain more reliable confidence score that overcomes the aforementioned limitations. To overcome the problem shown in Figure 2b, we apply Gaussian Mixture Modeling (GMM) on the feature space $f(X_t)$ to consider the covariance of the feature space and data structure. After conducting the GMM, we obtain the log-likelihood $\log p_{\text{data}}(x_i^t)_c$. Details of this are provided in the Appendix. To address the weakness shown in Figure 2a, we propose a Log Probability Gap (LPG). Given the probability $p(x) = \big(p(x)_1, p(x)_2, \cdots, p(x)_K\big)$ where $p(x)_c = p(\hat{y} = c|x)$, LPG is the scaled minimum gap from the pseudo-labeled element to those of the log probability $\log p_{\text{data}}(x)$ obtained by the GMM on the target feature space. First, we define MINGAP for each sample and the minimum gap from the pseudo-labeled element of the corresponding log probability.

$$\text{MINGAP}(x_i^t, \hat{y}_i^t) = \log p_{\text{data}}(x_i^t)_{\hat{y}_i^t} - \max_a \log p_{\text{data}}(x_i^t)_a, \text{ where } a \in \{1, 2, \cdots, K\}, \ a \neq \hat{y}_i^t. \tag{3}$$



(a) (b)

Figure 2: (a) An example of the limitations of Maxprob and Entropy scores. Red indicates a high confidence, blue means low-confidence, and green means the same confidence. MINGAP shows better reliability in this case. (b) An example of the weakness of the Cosine score. Based on the cosine distance, $(x^1_{c_1}, x^2_{c_1})$ has the same confidence-level with $(x^1_{c_2}, x^2_{c_2})$. However, $(x^1_{c_1}, x^2_{c_1})$ is more confident when the data structure is considered.

LPG is the normalized MINGAP, the value of which is between 0 and 1,

$$\text{LPG}(x_i^t, \hat{y}_i^t) = \frac{\text{MINGAP}(x_i^t)}{\max_j \text{MINGAP}(x_j^t, \hat{y}_j^t)}, \quad i, j \in \{1, 2, \cdots, n_t\}. \tag{4}$$

LPG provides a high score for a sample far from the decision boundary based on GMM: thus, LPG can complement the weakness of the other scores.

The confidence score Probability of Pseudo-label (PPL) is also used to include the knowledge of model into the confidence score. PPL is the model probability of the corresponding pseudo-label.

$$\text{PPL}(x_i^t, \hat{y}_i^t) = p_M(x_i^t)_{\hat{y}_i^t}. \tag{5}$$

The Joint Model-Data Structure (JMDS) score is the product of the LPG and PPL:

$$\text{JMDS}(x_i^t, \hat{y}_i^t) = \text{LPG}(x_i^t, \hat{y}_i^t) \cdot \text{PPL}(x_i^t, \hat{y}_i^t). \tag{6}$$

The JMDS score contains data-structure-wise knowledge from the LPG and model-wise knowledge from the PPL. In general, only one type of knowledge is considered for the existing scores. To the best of our knowledge, this is the first study proposing a confidence score jointly using the knowledge of the data structure and model. The experiment results regarding the superiority of the JMDS score compared to the other scores are presented in Section 4.3.

Why is it better to consider PPL and LPG together in source-free UDA? In general, the model fits the training data, and thus the model- and data-structure-wise knowledge are equivalent. However, in the source-free UDA, the pre-trained source model fits the source which is different from the training data. Therefore, the model- and data-structure-wise knowledge are different, and the learning process should consider both types of knowledge together. With CoWA, the PPL scores represent the model and source-wise knowledge, whereas the LPG scores represent the data-structure- and target-wise knowledge. Consequently, to extract both the source and target domain knowledge, we should consider the PPL and LPG scores together.

3.2 Suppressed Cross Entropy (SCE) loss

We conducted GMM on the feature space $f(X_t)$ and computed the JMDS score of the samples. Although the JMDS score can be obtained for every iteration, it requires heavy computation. Therefore, we compute the JMDS score offline, which ignores the online model information. Instead, to utilize online model information, we newly designed a Suppressed Cross Entropy (SCE) loss which is used instead of Cross Entropy (CE) loss as follows:

$$\begin{aligned} \mathcal{L}_{\text{SCE}}(x_i^t, \hat{y}_i^t) &= -p_M(x_i^t)_{\hat{y}_i^t} \log p_M(x_i^t)_{\hat{y}_i^t}, \\ \nabla \mathcal{L}_{\text{SCE}}(x_i^t)_{\hat{y}_i^t} = -p_M(x_i^t)_{\hat{y}_i^t} \nabla \mathcal{L}_{\text{CE}}(x_i^t)_{\hat{y}_i^t} &= -p_M(x_i^t)_{\hat{y}_i^t}(1 - p_M(x_i^t)_{\hat{y}_i^t}). \end{aligned} \tag{7}$$

Table 1: Accuracy (%) on Office-31 dataset for UDA and source-free UDA methods (ResNet-50).

Task	Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg.
SF-UDA	SFIT (Hou & Zheng, 2021)	89.9	91.8	73.9	98.7	72.0	99.9	87.7
	SHOT (Liang et al., 2020a)	94.0	90.1	74.7	98.4	74.3	**99.9**	88.6
	3C-GAN (Li et al., 2020)	92.7	93.7	75.3	98.5	77.8	99.8	89.6
	CoWA (worst)	**94.6**	**96.0**	75.4	**98.6**	76.1	99.4	90.0
	CoWA (average)	94.2	95.7	**77.3**	98.4	**78.0**	99.8	**90.6**
UDA	ResNet (He et al., 2016)	68.9	68.4	62.5	96.7	60.7	99.3	76.1
	CAN (Kang et al., 2019)	95.0	94.5	78.0	99.1	77.0	99.8	90.6
	RSDA-MSTN (Gu et al., 2020)	95.8	96.1	77.4	99.3	78.9	100	91.1
	FixBi (Na et al., 2020)	95.0	96.1	78.7	99.3	79.4	100	91.4

Note that $p_M(x_i^t)_{\hat{y}_i^t}$ in front of the log term is not trainable because we stop the gradient for it. The SCE loss acts as another confidence score weighting method based on the online model probability, whereas the PPL considers offline model probability. The gradient of SCE is a quadratic function in which the maximum of $|\nabla \mathcal{L}_{\text{SCE}}(x_i^t)_{\hat{y}_i^t}|$ is reached at $p_M(x_i^t)_{\hat{y}_i^t} = 0.5$. By contrast, $|\nabla \mathcal{L}_{\text{CE}}(x_i^t)_{\hat{y}_i^t}|$ is maximized at $p_M(x_i^t)_{\hat{y}_i^t} = 0.0$. In other words, SCE suppresses the effect of online low-confidence samples compared to CE.

CoWA adopts a sample reweighting scheme and thus, uses the SCE loss multiplied by the JMDS score,

$$\mathcal{L}_{\text{JMDS-SCE}}(x_i^t, \hat{y}_i^t) = -\text{JMDS}(x_i^t, \hat{y}_i^t) \cdot p_M(x_i^t)_{\hat{y}_i^t} \log p_M(x_i^t)_{\hat{y}_i^t}. \tag{8}$$

3.3 Weight mixup

The direct use of low-confidence samples with standard training leads to a loss of robustness against noisy pseudo-labels. To robustly learn using low-confidence samples, the sample reweighting scheme suppresses the low-confidence samples whose confidence scores are close to zero and who rarely participate in the training. Therefore, in this case, information provided by the target data distribution cannot be fully used. In this section, we propose a technique called a weight mixup, which is a variant of a mixup (Zhang et al., 2017), to enable robust learning even when involving more low-confidence samples during learning to utilize most of the information provided by the target data distribution.

A mixup mixes the images and corresponding labels. All mixed samples have the same weight for training. However, in the sample reweighting scheme, the sample has its own confidence score as a weight for training. This means that mixed images that use low-confidence samples for mixing should have a lower sample weight for mixup training. Therefore, the proposed weight mixup mixes the corresponding sample weight.

$$\begin{aligned} \tilde{x}_i^t &= \gamma x_i^t + (1-\gamma) x_j^t, \\ \tilde{y}_i^t &= \gamma \hat{y}_i^t + (1-\gamma) \hat{y}_j^t, \\ w(\tilde{x}_i^t) &= \gamma \cdot \text{JMDS}(x_i^t, \hat{y}_i^t) + (1-\gamma) \cdot \text{JMDS}(x_j^t, \hat{y}_j^t), \end{aligned} \tag{9}$$

where $\gamma \sim \text{Beta}(\alpha, \alpha)$, for $\alpha \in (0, \infty)$. The loss function of the weight mixup is modified as follows:

$$\mathcal{L}_{\text{mixup}}(\tilde{x}_i^t, \tilde{y}_i^t) = -w(\tilde{x}_i^t) \cdot (\gamma \cdot \log p_M(\tilde{x}_i^t)_{\hat{y}_i^t} + (1-\gamma) \cdot \log p_M(\tilde{x}_i^t)_{\hat{y}_j^t}). \tag{10}$$

Weight mixup makes the training robust in the following manner: A mixture of low- and high-confidence samples will produce a sample with mid-level confidence, which will robustly and effectively participate in the learning. By contrast, when low-confidence samples are mixed together, the resulting sample will also be suppressed.

3.4 Data augmentation

Data augmentation is a popular choice for various tasks such as supervised and unsupervised learning (Caron et al., 2020; Chen et al., 2020; Grill et al., 2020; Henaff, 2020; Krizhevsky et al., 2012). In our study, we first generate x_i^t, which is a transformed version of a raw image. We then create x'^t_i, which is the transformed image of x_i^t. Because the transformation we used has a RandomCrop function, x'^t_i emphasizes more local information on the raw image than x_i^t. In addition, x'^t_i shares the same JMDS score as x_i^t.

Table 2: Accuracy (%) on Office-Home for UDA and source-free UDA methods (ResNet-50).

Task	Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
SF-UDA	SHOT (Liang et al., 2020a)	57.1	**78.1**	81.5	68.0	78.2	78.1	67.4	54.9	**82.2**	73.3	58.8	84.3	71.8
	BAIT (Yang et al., 2020)	57.4	77.5	**82.4**	68.0	77.2	75.1	67.1	55.5	81.9	**73.9**	59.5	84.2	71.6
	CoWA (worst)	58.3	77.0	81.0	68.4	**81.0**	**80.3**	**68.1**	57.1	81.4	72.8	59.9	**84.7**	72.5
	CoWA (average)	**58.9**	**78.1**	81.1	69.3	**81.0**	**80.3**	67.8	**57.7**	81.7	72.5	**61.7**	84.5	**72.9**
UDA	ResNet-50 (He et al., 2016)	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
	RSDA-MSTN (Gu et al., 2020)	53.2	77.7	81.3	66.4	74.0	76.5	67.9	53.0	82.0	75.8	57.8	85.4	70.9
	FixBi (Na et al., 2020)	58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7

Algorithm 1 presents the pseudo-code for the entire CoWA procedure. The total loss of the CoWA is as follows:

$$\mathcal{L}_{\text{CoWA}}(x_i^t, \hat{y}_i^t) = \mathcal{L}_{\text{JMDS-SCE}}(x_i^t, \hat{y}_i^t) + \lambda_{\text{aug}} \cdot \mathcal{L}_{\text{JMDS-SCE}}(x'^t_i, \hat{y}_i^t) + \lambda_{\text{mixup}} \cdot \mathcal{L}_{\text{mixup}}(\tilde{x}^t, \tilde{y}^t). \quad (11)$$

4 EXPERIMENTS

4.1 SETUP

We evaluated our proposed method, CoWA, on three public UDA benchmarks. The first, Office-31 (Saenko et al., 2010), is a commonly used small-sized standard UDA benchmark that has three domains from different sources, *i.e.,* images collected from the Amazon website (A), a Web camera (W), and a DSLR camera (D). All three domains have 31 object classes of office supplies, and there are a total of 4,110 images. The second, Office-Home (Venkateswara et al., 2017), is a challenging medium-sized UDA benchmark containing Artistic images (Ar), Clip Art (Cl), Product images (Pr), and Real-World images (Rw) domains. All four domains have 15,500 images of 65 object classes. The third, VISDA-2017 (Peng et al., 2017), is a challenging large-sized UDA benchmark which contains a training dataset with 152,397 synthetic data and a test dataset with 55,388 real images with 12 categories. Implementation details are provided in the Appendix.

Algorithm 1: Training procedure for CoWA

Require : Unlabeled target data $X_t = \{x_i^t\}_{i=1}^{n_t}$, the model $M = g \circ f$ pre-trained by source data $X_s = \{x_i^s, y_i^s\}_{i=1}^{n_s}$, transformation $T(\cdot)$.

epoch $\leftarrow 0$;

while *epoch* < *max_epoch* **do**

- Conduct Gaussian Mixture Modeling on $f(X_t)$ and obtain log probability $\log p_{\text{data}}(X_t)$;
- Obtain pseudo-label $\hat{Y}_t = \{\hat{y}_i^t\}_{i=1}^{n_t}$ based on $\log p_{\text{data}}(X_t)$;
- Compute JMDS($X_t, \hat{Y}_t$) score using Equation (4), (5), (6);
- **for** $i \leftarrow 1$ **to** *iterations_per_epoch* **do**
  - Sample mini-batches of the target data $(x^t, \hat{y}^t)$;
  - Get transformed target data $x'^t = T(x^t)$;
  - Obtain $\tilde{x}, \tilde{y}, \tilde{w}$ using Equation (9);
  - Compute $\mathcal{L}_{\text{CoWA}}$ using Equation (11);
  - Update the model M with SGD;
- **end**
- *epoch* $\leftarrow$ *epoch* + 1;

end

4.2 COWA EVALUATION

We evaluated the target model M_t trained by CoWA using source-free UDA (Hou & Zheng, 2021; Li et al., 2020; Liang et al., 2020a; Yang et al., 2020) and various UDA baseline methods (Gu et al., 2020; Kang et al., 2019; Na et al., 2020) for comparison. Note that our task is source-free UDA, which is a more challenging task than conventional UDA which uses the source data during training. The performance of the CoWA depends on its pre-trained source model. Thus, we trained five different source models and reported the worst, and average performances to observe how the CoWA performance actually varies depending on the pre-trained model. Table 1, 2, and 3 show the classification accuracies for all tasks in each dataset. Here, SF-UDA indicates source-free UDA methods and UDA indicates conventional UDA methods. The best accuracy is indicated in bold, and the second-best accuracy is underlined.

In common, the worst case among the five CoWA trials achieved the best performance among the source-free UDA methods on all three UDA benchmarks. In source-free UDA, CoWA improved 1.0% point on the Office-31 dataset, 1.1% point on the Office-Home dataset, and 5.3% point on

Table 3: Accuracy (%) on VisDA-2017 for UDA and source-free UDA methods (ResNet-101).

Task	Method	airplane	bicycle	bus	car	horse	knife	motorcycle	person	plant	skateboard	train	truck	Average
SF-UDA	SFIT (Hou & Zheng, 2021)	94.3	79.0	84.9	63.6	92.6	92.0	88.4	79.1	92.2	79.8	87.6	43.0	81.4
	SHOT (Liang et al., 2020a)	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
	3C-GAN (Li et al., 2020)	94.8	73.4	68.8	74.8	93.1	95.4	88.6	84.7	89.1	84.7	83.5	48.1	81.6
	CoWA (worst)	96.3	**90.7**	**88.2**	65.2	**97.4**	**98.0**	89.5	86.4	95.4	**95.8**	**91.8**	62.4	87.7
	CoWA (average)	**96.8**	90.3	87.0	**67.4**	97.2	96.6	**90.4**	**87.3**	**95.6**	95.5	**91.8**	**62.5**	**88.2**
UDA	ResNet-101 (He et al., 2016)	72.3	6.1	63.4	91.7	52.7	7.9	80.1	5.6	90.1	18.5	78.1	25.9	49.4
	CAN (Kang et al., 2019)	97.0	87.2	82.5	74.3	97.8	96.2	90.8	80.7	96.6	96.3	87.5	59.9	87.2
	FixBi (Na et al., 2020)	96.1	87.8	90.5	90.3	96.8	95.3	92.8	88.7	97.2	94.2	90.9	25.7	87.2

Table 4: Accuracy (%) on Office-Home for ODA and PDA (ResNet-50).

Task (ODA)	Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
SF-UDA	SHOT (Liang et al., 2020a)	**64.5**	**80.4**	**84.7**	63.1	**75.4**	81.2	65.3	59.3	83.3	69.6	**64.6**	82.3	72.8
	CoWA	63.6	80.1	83.9	**67.6**	**75.4**	**84.3**	**66.4**	**61.6**	**84.4**	**73.2**	63.6	**83.8**	**74.0**
UDA	ResNet-50 (He et al., 2016)	53.4	52.7	51.9	69.3	61.8	74.1	61.4	64.0	70.0	78.7	71.0	74.9	64.3
	STA (Liu et al., 2019)	58.1	53.1	54.4	71.6	69.3	81.9	63.4	65.2	74.9	85.0	75.8	80.8	69.5
	PGL (Luo et al., 2020)	61.6	77.1	85.9	68.8	72.0	82.8	72.2	58.4	82.6	78.6	65.0	83.0	74.0
Task (PDA)	Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
SF-UDA	SHOT (Liang et al., 2020a)	64.8	85.2	**92.7**	**76.3**	77.6	88.8	79.7	64.3	89.5	80.6	66.4	85.8	79.3
	CoWA	**71.2**	**89.4**	91.1	75.9	**79.7**	**91.6**	**80.4**	**71.3**	**93.0**	**83.6**	**72.8**	**89.3**	**82.4**
UDA	ResNet-50 (He et al., 2016)	46.3	67.5	75.9	59.1	59.9	62.7	58.2	41.8	74.9	67.4	48.2	74.2	61.3
	SAFN (Xu et al., 2019)	58.9	76.3	81.4	70.4	73.0	77.8	72.4	55.3	80.4	75.8	60.4	79.9	71.8
	BA^{3}US (Liang et al., 2020b)	60.6	83.1	88.4	71.8	72.8	83.4	75.5	61.6	86.5	79.3	62.8	86.1	76.0

the VISDA-2017 dataset in terms of the average accuracy. It indicates that the larger the dataset size, the greater the effectiveness of CoWA. Moreover, the average accuracies of CoWA trials on the Office-Home and VISDA-2017 dataset achieved the best performance among all UDA methods including conventional UDA methods. CoWA improved 0.2% point on the Office-Home dataset, and 1.0% point on the VISDA-2017 dataset compared to state-of-the-art UDA methods.

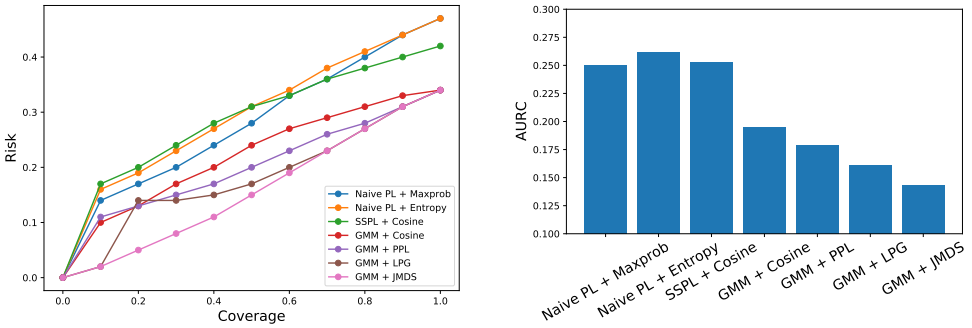
CoWA can also be easily extended to other UDA tasks: open-set DA (ODA), and partial-set DA (PDA). ODA assumes that the target domain contains all classes of the source and additional private unknown classes. PDA assumes that the source domain contains all classes of the target. The experiments of ODA and PDA follow the protocols of the Office-Home dataset in Liang et al. (2020a). Details are provided in Appendix. Table 4 provides the result of additional experiments. CoWA achieved the best performance on both ODA and PDA tasks.

4.3 Ablation Study and Discussion

We conducted ablation studies for the components of CoWA. The results are reported in Table 5. The upper part treats a sample reweighting scheme with JMDS confidence score, and the lower part treats SCE, weight mixup, and data augmentation.

Discussion of sample reweighting scheme and JMDS score We introduced the JMDS score as a reliable confidence score. To prove its efficacy, we compared the JMDS score with other scores (Kang et al., 2019; Mandelbaum & Weinshall, 2017). The pseudo-label of Maxprob and Entropy scores is given by the index of maximum model probability $\text{argmax}_c \, p_M(x_i^t)_c$ following naive PL (Lee et al., 2013). Pseudo-label of Cosine is given by SSPL, which is proposed by Liang et al. (2020a), and GMM. SSPL uses the modified K-means clustering to obtain pseudo-labels.

To evaluate the confidence score, we measured the AURC (Ding et al., 2020) using a 0/1 loss function which returns a value of 1 if the pseudo-label and ground truth label of the sample are different, and returns the value of 0 if they are the same. The risk-coverage curve is first proposed by Geifman et al. (2018). After obtaining the high-confidence set, $X_t^h = \{x_i^t | \kappa(x_i^t, \hat{y}_i^t) > \tau\}$, where τ is a threshold, risk is the average empirical loss of X_t^h, and the coverage means $|X_t^h|/|X_t|$. A lower AURC value indicates the better reliability since it implies a lower risk for the same coverage. When 0/1 loss is applied, a high AURC indicates a low correctness of the pseudo-label.



(a) (b)

Figure 3: (a) The Risk-Coverage Curve and (b) the AURC value of various strategies.

The experimental results are shown in Figure 3. We used the same pre-trained source model of the VISDA-2017 dataset for all strategies. We first compared GMM pseudo-labeling to naive PL and SSPL at a coverage of 1.0 in Figure 3a, which means empirical risk for all samples. GMM showed the lowest risk compared to other pseudo-labeling methods. This reveals that pseudo-labeling with GMM is better than naive PL and SSPL. Next, in Figure 3b, we can quantitatively compare the various strategies based on the AURC value. Using PPL or LPG score alone is worse than JMDS score, which is a product of two scores. This reveals that jointly using the knowledge of the model and data structure is better than considering only one aspect.

Then does sample reweighting scheme with a reliable confidence score really perform better? As shown in the upper part of Table 5, the Maxprob sample reweighting with naive PL achieves 3% point higher accuracy than the standard training with naive PL. Moreover, sample reweighting with the JMDS score, which is the best score based on AURC, shows the best performance among various strategies and even surpasses the state-of-the-art source-free UDA methods.

Table 5: Ablation results (%) of Office-31 dataset from the same source model

Ablation study	Avg
Source pre-trained model (not train)	78.6
Source pre-trained model + GMM PL (not train)	83.4
Naive PL + Standard training + CE	80.6
Naive PL + Maxprob weighting + CE	83.6
GMM PL + PPL weighting + CE	87.7
GMM PL + LPG weighting + CE	89.5
GMM PL + JMDS weighting + CE	90.1
GMM PL + JMDS + CE + Weight Mixup	90.0
GMM PL + JMDS + CE + Aug	89.7
GMM PL + JMDS + CE + Weight Mixup + Aug	<u>90.5</u>
GMM PL + JMDS + SCE + Weight Mixup + Aug (CoWA)	**90.9**

Discussion of SCE loss, weight mixup, and data augmentation As shown in the lower part of Table 5, we obtained the best performance when the SCE loss, weight mixup, and data augmentation are all applied. This clearly shows that each component is essential for CoWA training. First, as discussed in Section 3.2, SCE loss suppresses online low-confidence samples. Second, weight mixup produces samples near the decision boundary based on GMM because γ is near 0.5 and α in our experiments is high. Third, data augmentation creates a transformed image that contains more local-view information of the original image. The three components of CoWA play orthogonal roles and thus the interplay among these three components leads to the superior results of CoWA.

5 Conclusion and Future Work

In this study, we propose CoWA, a novel source-free UDA method that applies sample reweighting scheme to robustly learn with noisy pseudo-labels. CoWA uses the JMDS score designed for source-free UDA to extract both model- and data-structure-wise knowledge. The SCE loss and weight mixup are added to robustly leverage the low-confidence samples, and data augmentation enhances the feature discriminative power. Experiments show that the proposed CoWA achieves a state-of-the-art performance on various UDA benchmarks. However, the performance of the CoWA varies by approximately 1% point depending on the pre-trained source model, which must be resolved. In a future study, we will consider more stable training that is less affected by the pre-trained model and apply a strategy for choosing which model is more transferable.

REFERENCES

Kaidi Cao, Yining Chen, Junwei Lu, Nikos Arechiga, Adrien Gaidon, and Tengyu Ma. Heteroskedastic and imbalanced deep learning with adaptive regularization. *arXiv preprint arXiv:2006.15766*, 2020.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pp. 1062–1070. PMLR, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Yukun Ding, Jinglan Liu, Jinjun Xiong, and Yiyu Shi. Revisiting the evaluation of uncertainty estimation and its application to explore model complexity-uncertainty trade-off. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 4–5, 2020.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1180–1189, Lille, France, 07–09 Jul 2015. PMLR.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. *arXiv preprint arXiv:1705.08500*, 2017.

Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. Bias-reduced uncertainty estimation for deep neural classifiers. *arXiv preprint arXiv:1805.08206*, 2018.

Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. *International Conference on Learning Representations*, 2017.

Tieliang Gong, Qian Zhao, Deyu Meng, and Zongben Xu. Why curriculum learning & self-paced learning work in big/noisy data: A theoretical perspective. *Big Data & Information Analytics*, 1 (1):111, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Xiang Gu, Jian Sun, and Zongben Xu. Spherical space domain adaptation with robust pseudo-label loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9101–9110, 2020.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *arXiv preprint arXiv:1804.06872*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pp. 4182–4192. PMLR, 2020.

Jay Heo, Hae Beom Lee, Saehoon Kim, Juho Lee, Kwang Joon Kim, Eunho Yang, and Sung Ju Hwang. Uncertainty-aware attention for reliable interpretation and prediction. *arXiv preprint arXiv:1805.09653*, 2018.

Yunzhong Hou and Liang Zheng. Visualizing adapted knowledge in domain transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13824–13833, 2021.

Herman Kahn and Andy W Marshall. Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953.

Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4893–4902, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pp. 2805–2814. PMLR, 2018.

M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *NIPS*, volume 1, pp. 2, 2010.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

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

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

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

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

Jian Liang, Yunbo Wang, Dapeng Hu, Ran He, and Jiashi Feng. A balanced and uncertainty-aware approach for partial domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 123–140. Springer, 2020b.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Hong Liu, Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Qiang Yang. Separate to adapt: Open set domain adaptation via progressive separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2927–2936, 2019.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 97–105, Lille, France, 07–09 Jul 2015. PMLR.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2208–2217. JMLR. org, 2017.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 1640–1650, 2018.

Yadan Luo, Zijian Wang, Zi Huang, and Mahsa Baktashmotlagh. Progressive graph learning for open-set domain adaptation. In *International Conference on Machine Learning*, pp. 6468–6478. PMLR, 2020.

Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pp. 3355–3364. PMLR, 2018.

Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Ensemble of exemplar-svms for object detection and beyond. In *2011 International conference on computer vision*, pp. 89–96. IEEE, 2011.

Amit Mandelbaum and Daphna Weinshall. Distance-based confidence score for neural network classifiers. *arXiv preprint arXiv:1709.09844*, 2017.

Baharan Mirzasoleiman, Kaidi Cao, and Jure Leskovec. Coresets for robust training of deep neural networks against noisy labels. *Advances in Neural Information Processing Systems*, 33, 2020.

Jaemin Na, Heechul Jung, HyungJin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. *arXiv preprint arXiv:2011.09230*, 2020.

Tanya Nair, Doina Precup, Douglas L Arnold, and Tal Arbel. Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. *Medical image analysis*, 59: 101557, 2020.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1944–1952, 2017.

Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

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

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pp. 4334–4343. PMLR, 2018.

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

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*, pp. 3723–3732, 2018.

Seonguk Seo, Paul Hongsuck Seo, and Bohyung Han. Confidence calibration in deep neural networks through stochastic inferences. *arXiv preprint arXiv:1809.10877*, 2018.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450. Springer, 2016.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5552–5560, 2018.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

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*, pp. 7167–7176, 2017.

Brendan Van Rooyen, Aditya Krishna Menon, and Robert C Williamson. Learning with symmetric label noise: The importance of being unhinged. *arXiv preprint arXiv:1505.07634*, 2015.

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*, pp. 5018–5027, 2017.

Xinshao Wang, Yang Hua, Elyor Kodirov, and Neil M Robertson. Imae for noise-robust learning: Mean absolute error does not treat examples equally and gradient magnitude's variance matters. *arXiv preprint arXiv:1903.12141*, 2019.

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

Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1426–1435, 2019.

Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Unsupervised domain adaptation without source data by casting a bait. *arXiv preprint arXiv:2010.12427*, 2020.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zizhao Zhang, Han Zhang, Sercan O Arik, Honglak Lee, and Tomas Pfister. Distilling effective supervision from severe label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9294–9303, 2020.

Han Zhao, Shanghang Zhang, Guanhang Wu, José M. F. Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 8559–8570. Curran Associates, Inc., 2018.

A GMM

We conduct Gaussian Mixture Modeling (GMM) on the feature space $f(X_t)$ to consider the covariance of the feature space and data structure. Using GMM, we can obtain the parameters μ_c, Σ_c, andπ_c and log-likelihood

$$\log p(x_i^t|\mu_c, \Sigma_c) = -\frac{1}{2}(d \log 2\pi + \log|\Sigma_c| + (f(x_i^t) - \mu_c)^T \Sigma_c^{-1}(f(x_i^t) - \mu_c)), \tag{12}$$

where π_c, μ_c and Σ_c are the mixing coefficient, mean vector and covariance matrix of the class $c \in \{1, 2, \cdots, K\}$ respectively. Now, we can obtain the log probability of x_i for c:

$$\log p_{\text{data}}(x_i^t)_c = \frac{\log \pi_c(x_i^t) + \log p(x_i^t|\mu_c, \Sigma_c)}{\Sigma_{c'}\{\log \pi_{c'}(x_i^t) + \log p(x_i^t|\mu_{c'}, \Sigma_{c'})\}} \quad \text{where} \quad c, c' \in \{1, 2, \cdots, K\}. \tag{13}$$

B More implementation details

B.1 Datasets

First, Office-31 dataset was proposed by Saenko et al. (2010). They collected images from website (`www.amazon.com`) or themselves. Second, Office-Home dataset was proposed by Venkateswara et al. (2017). The Art (Ar) and Real-World (Rw) domains were created by public domain images from websites like `www.deviantart.com` and `www.flickr.com`. The Product (Pr) domain images were collected from website `www.amazon.com` and the Clipart (Cl) images were collected from multiple clipart websites. Lastly, VISDA-2017 dataset was proposed by Peng et al. (2017). The training domain images were collected from CAD-synthetic Images, the validation domain was created by MS COCO dataset, and the testing domain images were gathered from YouTube Bounding Boxes (YT-BB) dataset. All three datasets were collected from public data, so they do not contain personally identifiable information or offensive content.

Table 6: The license of datasets from `https://paperswithcode.com/datasets`.

Dataset	License
Office-31	Unknown
Office-Home	Custom (non-commercial research and educational purposes)
VISDA-2017	Custom

B.2 Resources for experiments

We used a single GeForce RTX-2080TI for all experiments. Table 7 shows the spent time for each task. The time to create the pre-trained source model is excluded. A training time for CoWA depends on target dataset and network architecture. Therefore, the source dataset is not indicated in the table.

Table 7: The times spent on each task.

Dataset	Task	Times
Office-31	$\cdot \to$ A	39m
Office-31	$\cdot \to$ D	12m
Office-31	$\cdot \to$ W	14m
Office-Home	$\cdot \to$ Ar	22m
Office-Home	$\cdot \to$ Cl	37m
Office-Home	$\cdot \to$ Pr	37m
Office-Home	$\cdot \to$ Rw	52m
VISDA-2017	T $\to$ V	6h 30m

B.3 Transformations

We apply the transform used in Liang et al. (2020a). During CoWA training, $\mathsf{Resize}(256, 256)$, $\mathsf{RandomCrop}(224)$, and $\mathsf{RandomHorizontalFlip}()$ functions in Pytorch are used sequentially.

B.4 ODA AND PDA IMPLEMENTATION

We follow the protocols in Liang et al. (2020a) on the Office-Home dataset which has 65 classes. ODA assumes first 25 source classes and 65 target classes. PDA assumes 65 source classes and first 25 target classes. For ODA, we follow the technique in SHOT that filters out a set of samples which has a high entropy value. For PDA, we should filter out absent classes. Therefore, we propose a technique to filtering classes. First, run EM iteration once. Second, filter out the founded classes which have a smaller number of samples than a threshold. Finally, iteratively run aforementioned steps until converges.

C MORE EXPERIMENTAL RESULTS

C.1 THE MEAN AND STANDARD DEVIATION

Table 8 shows the mean and standard deviation of the performance of CoWA. We conducted five experiments with different random seeds. The mean value is the same as Table 1, 2, 3 in Section 5.

Table 8: The mean accuracy (%) and standard deviation of it on Office-31 dataset.

Dataset	mean±std
Office-31	90.6 ± 0.6
Office-Home	72.9 ± 0.3
VISDA-2017	88.2 ± 0.4

C.2 EXPERIMENTS ABOUT WEIGHT MIXUP

The effectiveness of a weight mixup is shown in Table 9. Note that the other components except for the weight mixup are fixed for this experiment. The weight mixup boosts the average accuracy on Office-31 dataset more than the mixup Zhang et al. (2017). Moreover, the table shows that the performance of CoWA boosts when a mixup coefficient, α, is getting larger. It reveals that creating samples near decision boundary is beneficial for CoWA training.

Table 9: The experimental results (%) about weight mixup on Office-31 dataset.

	α	Avg.
Mixup	0.5	87.3
	1.0	87.5
	4.0	88.7
	16.0	89.0
Weight mixup	0.5	90.1
	1.0	90.3
	4.0	90.4
	16.0 (CoWA)	**90.6**

C.3 ADDITIONAL JMDS SCORE EVALUATIONS

Table 10 shows the results of the JMDS confidence score evaluation. The experiments demonstrate that the JMDS confidence score obtains the best performance based on AURC in most cases.

C.4 ABLATION STUDIES ON OTHER DATASETS

We conduct more ablation studies for other datasets, Office-Home and VISDA-2017. Table 11 shows the results of the average accuracy with five trials. Both of the datasets show the best performance when using all components of CoWA. Note that the performance of GMM PL + JMDS + CE strategy is also the best among source-free UDA methods.

Table 10: Evaluations of the JMDS score based on AURC.

Dataset	Task	Naive PL+Maxprob	Naive PL+Entropy	SSPL+Cosine	GMM+Cosine	GMM+PPL	GMM+LPG	GMM+JMDS
Office-31	A→D	0.047	0.051	**0.018**	0.031	0.039	0.033	0.033
	A→W	0.074	0.081	**0.034**	0.045	0.059	0.042	0.044
	D→A	0.158	0.165	0.140	0.130	0.131	0.127	**0.115**
	D→W	0.007	0.008	0.009	0.009	0.005	0.004	**0.004**
	W→A	0.157	0.167	**0.107**	0.108	0.132	0.120	0.113
	W→D	0.002	0.002	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
Office-Home	Ar→Cl	0.308	0.316	0.296	0.274	0.278	0.265	**0.256**
	Ar→Pr	0.140	0.145	**0.100**	0.105	0.116	0.125	0.104
	Ar→Rw	0.088	0.095	0.086	0.086	0.076	0.086	**0.068**
	Cl→Ar	0.238	0.249	0.200	0.194	0.212	0.216	**0.197**
	Cl→Pr	0.159	0.168	**0.105**	0.113	0.131	0.125	0.115
	Cl→Rw	0.151	0.159	0.113	0.113	0.125	0.115	**0.106**
	Pr→Ar	0.237	0.246	0.185	**0.184**	0.210	0.214	0.190
	Pr→Cl	0.365	0.375	0.339	0.315	0.327	0.293	**0.293**
	Pr→Rw	0.095	0.099	0.080	0.082	0.084	0.091	**0.073**
	Rw→Ar	0.138	0.147	0.129	0.125	0.126	0.15 4	**0.118**
	Rw→Cl	0.314	0.325	0.298	0.284	0.275	0.248	**0.238**
	Rw→Pr	0.073	0.078	0.062	0.063	0.065	0.078	**0.059**
VISDA-2017	T→V	0.274	0.284	0.261	0.202	0.204	0.172	**0.162**

Table 11: Additional ablation results (%) for the other datasets.

	Office-Home	VISDA-2017
GMM PL + JMDS + CE	71.9	83.1
GMM PL + JMDS + CE + Weight Mixup	72.1	86.5
GMM PL + JMDS + CE + Aug	71.9	83.4
GMM PL + JMDS + CE + Weight Mixup + Aug	72.3	86.7
GMM PL + JMDS + SCE + Weight Mixup + Aug (CoWA)	**72.9**	**88.3**