Gradient-Based Sample Selection for Black-Box Universal Domain Adaptation

Qiuyan He[1], **Minghua Deng**[1,2,3*],

[1]School of Mathematical Sciences, Peking University
[2]Center for Statistical Science, Peking University
[3]Center for Quantitative Biology, Peking University
heqy@pku.edu.cn, dengmh@math.pku.edu.cn

Abstract

Universal domain adaptation (UniDA) transfers knowledge from a labelled source domain to an unlabelled target domain under domain-shift and category-shift for annotation. In reality, due to privacy protection or other limits, not only source data but also pre-trained models on it may be unavailable when training on target data. In this paper, we go a step further to explore the black-box universal domain adaptation (B^2-UniDA) problem. It requires tackling the labelling task under shifts by only accessing the interface of pre-trained source models. To this end, we introduce GSS which proposes a novel sample selection criterion based on gradient descent and Bayes' Theorem to identify samples of potential unknown classes. This criterion doesn't require manually-set thresholds depending on data used and is suitable for various datasets. GSS builds an open-set classifier and enables it to estimate probabilities of belonging to each class including the unknown category and adjust estimates adaptively. To overcome class imbalance, especially imbalance between the unknown and known classes, we propose a balancing mechanism by measuring training status and estimating DA type. In addition to distilling knowledge from source model outputs, we focus on mining the categorical structure of target domain by self-training. Experiments on benchmarks show the state-of-the-art performance of GSS compared to typical methods, including source models or source data dependent methods.

1 Introduction

Unsupervised domain adaptation (UDA) (Ben-David et al. 2010) transfers knowledge from a labelled source domain ($\mathcal{D}_s$) to an unlabelled target domain ($\mathcal{D}_t$) for annotation under domain-shift (Wilson and Cook 2020). Let L_s and L_t be label sets of $\mathcal{D}_s$ and $\mathcal{D}_t$. UDA generally assumes $L_s = L_t$ (Ganin and Lempitsky 2015), which is also named closed-set DA (CDA). This doesn't consider category-shift between domains. Therefore, partial DA (PDA) (Cao et al. 2018a,b; Zhang et al. 2018b) assumes $L_t \subset L_s$, while the widely-used open-set DA (ODA) (Saito et al. 2018) assumes $L_s \subset L_t$. Early version of ODA (Busto and Gall 2017) assumes $L_t \cap L_s \neq \emptyset$ and both domains have private classes, which is called open-partial DA (OPDA) now. In fact, the relationship of L_s and L_t is usually unknown. Thus universal DA (UniDA) (You et al. 2019; Saito et al. 2020) only assumes $L_t \cap L_s \neq \emptyset$ to tackle all kinds of category-shift.

Traditional UniDA methods train with both domain data. Due to security concerns and other restrictions, source data can be inaccessible when training. Moreover, training with both domain data is computational heavy when datasets are large-scale. It's also inefficient to retrain all data once new target data is obtained. Therefore, source-free universal domain adaptation (SF-UniDA) (Kundu et al. 2020; Liang et al. 2021) tackles the UniDA problem when only pre-trained models on $\mathcal{D}_s$ are available. However, only utilizing source models can still suffer from attacks such as model-inversion attack (Zhang et al. 2020) and membership inference attack (Hu et al. 2022) leading to information leakage. A more practical setting is the black-box universal domain adaptation (B^2-UniDA) (Deng et al. 2021), where only the interface of pre-trained models is available. Querying target data through the interface returns test results of pre-trained models. B^2-UniDA uses these results to label target data when unknown domain-shift and category-shift exist.

Compared with traditional UniDA and SF-UniDA, B^2-UniDA is more practical and safer, but also more challenging. Firstly, test results of pre-trained models provide less knowledge of $\mathcal{D}_s$. Initialization of target models can't use source information. Secondly, limited source information increases the difficulty of tackling shifts. Different from traditional UniDA methods (You et al. 2019; Fu et al. 2020; Chen et al. 2022a,b,c), B^2-UniDA methods can't use cross-domain samples to eliminate domain-shift and find common classes.

B^2-UniDA also inherits challenges of UniDA. First, class imbalance (Johnson and Khoshgoftaar 2019) may occur in both domains. Due to domain-shift, conditions of class imbalance are different between domains even for common classes. Moreover, since all target-private classes are regarded as unknown when labelling, imbalance may exist between the unknown category and each known class or the known category overall. Such imbalance may lead to model collapse. Besides, negative transfer (Wang et al. 2019) may occur especially when there're many source-private classes.

Nowadays, UniDA has attracted much attention but B^2-UniDA is seldom studied (Deng et al. 2021). Current UniDA works mainly detect unknown classes by comparing self-defined statistics with given thresholds (You et al. 2019; Saito et al. 2020; Fu et al. 2020; Yin et al. 2021). Perfor-

*Corresponding author

mance of them highly depends on thresholds used, best values of which are unknown and rely on data. Applying them to different datasets is inconvenient. Besides, performance of adversarial methods (You et al. 2019; Fu et al. 2020; Yin et al. 2021) greatly relies on hyperparameters and are easy to collapse. Training them can be computational heavy.

To tackle challenges of B^2-UniDA and avoid drawbacks of current methods, we propose a ***G**radient-based **S**ample **S**election* (GSS) criterion for B^2-UniDA. This criterion is derived from gradient descent and Bayes' Theroem (Joyce 2021). It doesn't need to select thresholds manually for different datasets to detect unknown classes. Such criterion enables our open-set classifier to estimate the probability of belonging to unknown classes. To tackle class imbalance especially imbalance between the unknown and known categories and learn each category fully, we propose a balancing mechanism where we estimate possible DA type and measure the training status. Due to shifts, we use source model outputs as reference and focus on learning target class knowledge. On one hand, we mine the categorical structure of $\mathcal{D}_t$ by self-training. On the other hand, we conduct self-supervised learning based on sample selection. Extensive experiments show the superiority of GSS.

Our contributions can be summarized as follows:

- We propose a novel sample selection criterion to detect unknown classes which is theoretically meaningful. It doesn't need to change thresholds with the training data and can be applied directly.
- We devise a balancing mechanism to avoid imbalanced categorical learning. It adjusts the relative attention the model pays to the 'unknown' and 'known' categories according to the training status and DA type estimated.
- We define an open-set classifier that can estimate probabilities of belonging to each category including the unknown category and adjust estimates during training. It provides predictions directly when inference.
- Experiments on benchmarks under all DA settings show the superiority of our GSS compared with current B^2-UniDA method UB2DA (Deng et al. 2021). GSS also exhibits comparable, if not the best, performance to source data or source models dependent UniDA methods.

2 Related Work

Universal Domain Adaptation. Early UniDA methods adopt adversarial learning to learn domain-invariant features of common classes (You et al. 2019; Fu et al. 2020; Yin et al. 2021). Performance of them relies on hyperparameters and is unstable. When detecting unknown classes, UniDA methods mainly compare self-defined metrics with given thresholds. Metrics including margin (Yin et al. 2021), entropy (Saito et al. 2020), consistency (Yu, Hashimoto, and Ushiku 2021), uncertainty (Chen et al. 2022b), similarity score (You et al. 2019) and combinations of multiple metrics (You et al. 2019; Fu et al. 2020; Cai et al. 2021). Best values of thresholds are data-dependent and unknown. Some methods turn to other ways. DCC (Li et al. 2021) learns target prototypes for labelling. But determining the class number can be computationally heavy. OVANet (Saito and Saenko 2021) uses multiple binary classifiers to identify unknown samples in a one-vs-all manner.

Source-free UniDA. Few works have studied SF-UniDA. USFDA (Kundu et al. 2020) generates negative samples to detect non-source classes in pre-training. It requires much storage space and is computationally heavy. Simulated categories through image-composition are meaningless in reality. UMAD (Liang et al. 2021) designs an informative consistency score to identify unknown classes. But the bandwidth to select highly-confident samples is set subjective.

Black-box UniDA. Deng et al. (Deng et al. 2021) propose UB2DA which relies on entropy to identify unknown classes. It optimizes entropy for high-confident samples. The threshold and band-width used are set empirically.

3 Methodology

3.1 Preliminary

Problem Formulation. Assume there is a labelled source domain $\mathcal{D}_s$ with data $D_s = \{x_i^s, y_i^s\}_{i=1}^{N_s}$ and distribution $p_s(x, y)$, and an unlabelled target domain $\mathcal{D}_t$ with data $D_t = \{x_i^t\}_{i=1}^{N_t}$ and distribution $p_t(x, y)$. Let $\mathcal{X}$ be the feature space, $\mathcal{Y}_s$ and $\mathcal{Y}_t$ be the label space of $\mathcal{D}_s$ and $\mathcal{D}_t$. $f_s : \mathcal{X} \to \mathcal{Y}_s$ is the pre-trained model on D_s. Denote the label set of $\mathcal{D}_s$ and $\mathcal{D}_t$ as L_s and L_t. Assuming $p_s(x, y) \neq p_t(x, y)$ and $L_s \cap L_t \neq \emptyset$, B^2-UniDA aims at learning $f : \mathcal{X} \to \mathcal{Y}_t$ to minimize the expected loss $\mathbb{E}_{p_t(x,y)} [\ell (f(x), y)]$ only with the access to D_t and the interface of f_s. Here $\ell(\cdot, \cdot)$ is a loss function.

Framework of GSS. Assume there is a low-dimensional space $\mathcal{Z}$, where samples of the same class are closer than those of different classes. Define a feature extractor $G : \mathcal{X} \to \mathcal{Z}$. Let $L_s = \{1, \ldots, K\}$ be source classes or known classes. The label $K + 1$ denotes target-private classes. We use an open-set classifier $C_1 : \mathcal{Z} \to \mathbb{R}^{K+1}$ to estimate probabilities of belonging to each category including the unknown one. Besides, a closed-set classifier $C_2 : \mathcal{Z} \to \mathbb{R}^K$ is used to estimate probabilities assuming samples are in known classes.

Given a sample with feature x, $z = G(x)$ is its latent representation. For $i = 1, 2$, denote the dimension and direct outputs of C_i as d_i and $g^{(i)}(x) = C_i(z)$. Let $h^{(i)}(x) = \exp(g^{(i)}(x))$. We have $p^{(i)}(x) \in \mathbb{R}^{d_i}$, $p_j^{(i)}(x) = h_j^{(i)}(x) / \sum_{l=1}^{d_i} h_l^{(i)}(x)$. Note that $d_1 = K + 1$ and $d_2 = K$.

3.2 Pre-training

We use source labels to guide models to learn representations of source classes and category boundaries. The total loss of pre-training is $\mathcal{L}_s = \mathcal{L}_{\text{crs}}^{(1)}(D_s) + \mathcal{L}_{\text{crs}}^{(2)}(D_s)$, where

$$\mathcal{L}_{\text{crs}}^{(l)}(D_s) = -\frac{1}{N_s} \sum_{i=1}^{N_s} \log p_{y_i^s}^{(l)}(x_i^s),\ l = 1, 2. \tag{1}$$

3.3 Distillation of Knowledge on Source Domain

To obtain categorical knowledge of $\mathcal{D}_s$, we can only use test results of f_s on D_t. Denote probabilities output by f_s as $\{p^{\text{pre}(l)}(x_i^t)\}_{i=1}^{N_t}$, $l = 1, 2$. Since f_s only learns known classes,

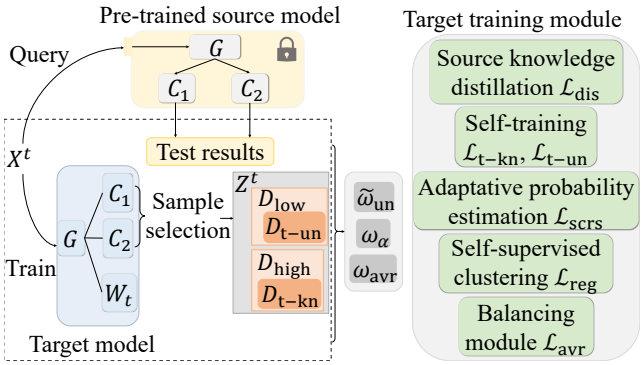


Figure 1: The structure of GSS training on the target domain

$p^{\text{pre}(1)}_{K+1}(x)$ is inaccurate. We use $\tilde{p}^{\text{pre}(1)}(x)$ to replace $p^{\text{pre}(1)}(x)$, where $\tilde{p}^{\text{pre}(1)}_j(x) = p^{\text{pre}(1)}_j(x)/(1-p^{\text{pre}(1)}_{K+1}(x))$, $j = 1, \ldots, K$.

To make test results more discriminative for better clustering on $\mathcal{D}_t$, for $j = 1, \ldots, K$ and $l = 1, 2$, we follow DEPICT (Dizaji et al. 2017) to redefine them as

$$q^{\text{pre}(l)}_j(x^t_i) = \frac{\tilde{p}^{\text{pre}(l)}_j(x^t_i)/(\sum_i \tilde{p}^{\text{pre}(l)}_j(x^t_i))^{\frac{1}{2}}}{\sum_{m=1}^{K}[\tilde{p}^{\text{pre}(l)}_m(x^t_i)/(\sum_i \tilde{p}^{\text{pre}(l)}_m(x^t_i))^{\frac{1}{2}}]}. \quad (2)$$

For simplicity, we denote $\tilde{p}^{\text{pre}(2)}(x) = p^{\text{pre}(2)}(x)$ here.

Then we let current probabilities distill knowledge from refined results. The loss is $\mathcal{L}_{\text{dis}} = \mathcal{L}^{(1)}_{\text{dis}}(D_t) + \mathcal{L}^{(2)}_{\text{dis}}(D_t)$, where

$$\mathcal{L}^{(l)}_{\text{dis}}(D_t) = -\frac{1}{N_t}\sum_{i=1}^{N_t}\sum_{j=1}^{K} q^{\text{pre}(l)}_j(x^t_i)\log p^{(l)}_j(x^t_i),\ l = 1, 2. \quad (3)$$

3.4 Gradient-Based Sample Selection Criterion

Gradient descent is a first-order iterative optimization algorithm (Lemaréchal 2012). Assume the function L is differentiable with respect to h. To solve $\min_h L(h)$ by gradient descent, let h_j and $r_j > 0$ be the value of h and learning rate at j-th step, the j-th update of h is $h_{j+1} = h_j - r_j \frac{\partial L}{\partial h}(h_j)$. If $\frac{\partial L}{\partial h}(h_j) < 0$, h will be increased to let L decrease. Conversely, h will be decreased. Since gradient descent focuses on the neighborhood of current point, we can take a simple function $L(h) = -h\log(h) - (1-h)\log(1-h)$ for example. When $L'(h) > 0$, i.e. $h < 0.5$, the h shall be decreased, otherwise be increased. Motivated by this, to enable C_1 to adjust estimates of probabilities belonging to unknown classes, we should find L and h both related to $p^{(1)}_{K+1}(x)$.

Given sample x, since $\sum_{j=1}^{K+1} p^{(1)}_j(x) = 1$, $p^{(1)}_{K+1}(x)$ is not a free variable when solving $\min_{p^{(1)}_{K+1}} L(p^{(1)})$. Recall that $p^{(1)}_j(x) = h^{(1)}_j(x)/(\sum_{l=1}^{K+1} h^{(1)}_l(x))$. It monotonically increases with $h^{(1)}_j(x)$. Besides, $h^{(1)}_j(x) = \exp(g^{(1)}_j(x))$, $j = 1, \ldots, K+1$ are positive and independent with each other. Therefore, we use $h^{(1)}_{K+1}(x)$ as the variable h.

To find the objective L, recall that C_1 estimates probabilities of belonging to each category and C_2 estimates probabilities when assuming samples are in known classes. Then

$$p^{(1)}_j(x) = P(Y = j|X = x),\ j = 1, \ldots, K+1. \quad (4)$$

$$p^{(2)}_j(x) = P(Y = j|X = x, Y \le K),\ j = 1, \ldots, K. \quad (5)$$

By Bayes' Theorem (Joyce 2021), we can have

$$\begin{aligned} &P(Y \neq j|X = x) = 1 - P(Y = j|X = x) \\ &= P(Y = K+1|X = x) + P(Y \le K, Y \neq j|X = x) \\ &\Longrightarrow p^{(1)}_j(x) = (1 - p^{(1)}_{K+1}(x))p^{(2)}_j(x),\ j = 1, \ldots, K. \end{aligned} \quad (6)$$

Let $p^{(1)}_{:K}(x) = (p^{(1)}_1(x), \ldots, p^{(1)}_K(x))$ and $p^{\text{mul}}(x) = (1 - p^{(1)}_{K+1}(x))p^{(2)}(x)$. To ensure Equation(6) holds, we should measure the discrepancy between $p^{(1)}_{:K}(x)$ and $p^{\text{mul}}(x)$ and minimize it. Such discrepancy is the L we are looking for.

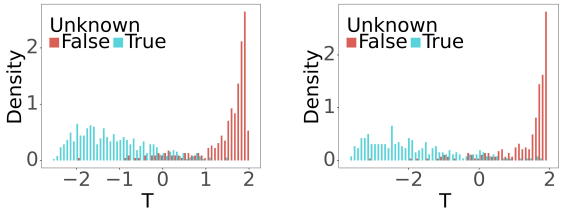
Common measures of differences in probability distributions are cross-entropy ($H(\cdot,\cdot)$) and KL divergence (Csiszar 1975) ($D_{KL}(\cdot,\cdot)$). By definition, $D_{KL}(p^{(1)}_{:K}(x), p^{\text{mul}}(x)) = -H(p^{(1)}_{:K}(x)) + H(p^{(1)}_{:K}(x), p^{\text{mul}}(x))$. $H(p^{(1)}_{:K}(x))$ may be maximized to minimize $D_{KL}(p^{(1)}_{:K}(x), p^{\text{mul}}(x))$. However, $H(p^{(1)}_{:K}(x))$ should be small if x is correctly predicted. Therefore, minimizing $H(p^{(1)}_{:K}(x), p^{\text{mul}}(x))$ is more suitable than D_{KL}. Due to the asymmetry of cross-entropy, we use the symmetric cross-entropy. When $p^{(1)}_{K+1}(x) < 1$, it is

$$\begin{aligned} &H_{\text{sym}}(p^{(1)}_{:K}(x), p^{\text{mul}}(x)) \\ &= -\sum_{j=1}^{K}\left[p^{(1)}_j(x)\log p^{\text{mul}}_j(x) + p^{\text{mul}}_j(x)\log p^{(1)}_j(x)\right]. \end{aligned} \quad (7)$$

Since $H_{\text{sym}} \ge 0$, $H_{\text{sym}} \to 0$ as $p^{(1)}_{:K}(x)$ and $p^{\text{mul}}(x)$ tend to the same one-hot vector, and $H_{\text{sym}} = 0$ when $p^{(1)}_{K+1}(x) = 1$, using H_{sym} as the objective L is suitable whether x is unknown or not. For each x, let $L = H_{\text{sym}}(p^{(1)}_{:K}(x), p^{\text{mul}}(x))$ and $h = h^{(1)}_{K+1}(x)$. When solving $\min_h L(h)$ with gradient descent, h will be increased if $\frac{\partial L}{\partial h} < 0$ and be decreased if $\frac{\partial L}{\partial h} > 0$. Select the term determining the sign of $\frac{\partial L}{\partial h}$, we get the sample selection metric T for each sample x as

$$\begin{aligned} T(x) = &-H(p^{(1)}_{:K}(x), p^{(2)}(x)) + 2(1 - p^{(1)}_{K+1}(x)) \\ &- (1 - p^{(1)}_{K+1}(x))H(p^{(2)}(x), p^{(1)}_{:K}(x)) \\ &+ (1 - p^{(1)}_{K+1}(x))\log(1 - p^{(1)}_{K+1}(x)). \end{aligned} \quad (8)$$

If $T(x) > 0$, x is highly likely to be in known classes since $h^{(1)}_{K+1}(x)$ is guided to decrease. Conversely, x most likely belongs to unknown classes. Using dlsr and webcam in Office (Saenko et al. 2010) as $\mathcal{D}_s$ and $\mathcal{D}_t$, the distribution of $T(x)$ is bimodal after pre-training with regard to whether x is unknown in Figure 2 when target-private classes exist.



(a) D2W in ODA (b) D2W in OPDA

Figure 2: Distribution of $T(x)$ for dslr2webcam

Self-training Based on Sample Selection. Divide D_t into $D_{\text{high}} = \{x \in D_t | T(x) > 0\}$ and $D_{\text{low}} = \{x \in D_t | T(x) < 0\}$. In the early stage of training, the knowledge distillation of $\mathcal{D}_s$ is inefficient. The model tends to learn pseudo labels from $\tilde{p}^{\text{pre}(1)}$ and $p^{\text{pre}(2)}$ first rather than specific probabilities. At this time, discrepancy between $p^{(1)}$ and $p^{(2)}$ mainly arises from different initialization of classifiers rather than difference between $\tilde{p}^{\text{pre}(1)}$ and $p^{\text{pre}(2)}$. When K is large, d_1 and d_2 are large. After summation along the dimension, large K can lead to $T(x) < 0$. This enlarges the inaccurate discrepancy between $p^{(1)}(x)$ and $p^{(2)}(x)$ during the initial training. Recall that the relative relationship between elements of a probability vector is easier to learn compared to specific values. To reduce the impact of large K, we further use the consistency between $\hat{y}^{(i)} = \arg\max p^{(i)}(x)$, $i = 1, 2$. Let

$$\begin{aligned} D_{\text{t-kn}} &= \{x \in D_t | T(x) > 0, \hat{y}^{(1)} = \hat{y}^{(2)}\}, \\ D_{\text{t-un}} &= \{x \in D_t | T(x) < 0, \hat{y}^{(1)} \neq \hat{y}^{(2)}\}. \end{aligned} \tag{9}$$

In fact, $D_{\text{t-kn}}$ is almost the same as D_{high}. Although $D_{\text{t-un}} \subseteq D_{\text{low}}$, they gradually become same during training.

To learn categorical knowledge of common classes in $\mathcal{D}_t$, we believe that samples in $D_{\text{t-kn}}$ are most likely to belong to known classes and utilize them. Let $N_{\text{t-kn}}$ be the size of $D_{\text{t-kn}}$. We define the loss $\mathcal{L}_{\text{t-kn}} = \mathcal{L}^{(1)}_{\text{pse}}(D_{\text{t-kn}}) + \mathcal{L}^{(2)}_{\text{pse}}(D_{\text{t-kn}})$, where

$$\mathcal{L}^{(l)}_{\text{pse}}(D_{\text{t-kn}}) = -\frac{1}{N_{\text{t-kn}}} \sum_{i=1}^{N_{\text{t-kn}}} \log p^{(l)}_{\hat{y}^{(l)}_i}(x^t_i),\ l = 1, 2. \tag{10}$$

Samples in $D_{\text{t-un}}$ are considered highly likely to be in unknown classes. Using them to detect unknown classes by

$$\mathcal{L}^{(1)}_{\text{p-un}}(D_{\text{t-un}}) = -\frac{1}{N_{\text{t-un}}} \sum_{i=1}^{N_{\text{t-un}}} [1 - (p^{(1)}_{K+1}(x^t_i))^3] \log p^{(1)}_{K+1}(x^t_i). \tag{11}$$

Here $1 - (p^{(1)}_{K+1}(x^t_i))^3$ is to let the model pay more attention to samples difficult to identify. $N_{\text{t-un}}$ is the size of $D_{\text{t-un}}$.

Since C_2 doesn't consider the unknown category, it can just output uncertain probabilities for $x \in D_{\text{t-un}}$ rather than direct predictions. Entropy is the common measure of uncertainty for probabilities output by the closed-set classifier (Saito et al. 2020). Thus we maximize $H(p^{(2)}(x))$ and the degree is determined by $p^{(1)}_{K+1}(x)$. Let

$$\mathcal{L}^{(2)}_{\text{pse}}(D_{\text{t-un}}) = \frac{1}{N_{\text{t-un}}} \sum_{i=1}^{N_{\text{t-un}}} \sum_{j=1}^{K} p^{\text{de}(1)}_{K+1}(x^t_i) p^{(2)}_j(x^t_i) \log p^{(2)}_j(x^t_i). \tag{12}$$

$p^{\text{de}(1)}_{K+1}$ means the gradient of $p^{(1)}_{K+1}$ isn't considered Here.

In the early stage of training, categorical learning is inadequate. $D_{\text{t-un}}$ may contain samples of known classes. These need to be corrected. Besides, $\mathcal{L}^{(1)}_{\text{p-un}}(D_{\text{t-un}})$ has no constraints on probabilities of being in known classes. Recalling Equation(6), we correct samples and estimate known class probabilities for $D_{\text{t-un}}$ by

$$\begin{aligned} \mathcal{L}^{(1)}_{\text{p-kn}}(D_{\text{t-un}}) = -\frac{1}{N_{\text{t-un}}} \sum_{i=1}^{N_{\text{t-un}}} \sum_{j=1}^{K} \left(1 - p^{\text{de}(1)}_{K+1}(x^t_i)\right) \\ \cdot p^{(2)}_j(x^t_i) \log p^{(1)}_j(x^t_i), \end{aligned} \tag{13}$$

Then the total self-training loss on $D_{\text{t-un}}$ is

$$\mathcal{L}_{\text{t-un}} = \mathcal{L}^{(1)}_{\text{p-kn}}(D_{\text{t-un}}) + \mathcal{L}^{(1)}_{\text{p-un}}(D_{\text{t-un}}) + \mathcal{L}^{(2)}_{\text{pse}}(D_{\text{t-un}}). \tag{14}$$

3.5 Adaptative Probability Estimation

Previous section focuses on learning known and unknown classes using samples most likely in them. However, $D_{\text{t-kn}}$ and $D_{\text{t-un}}$ may have few samples in the early stage. We need to utilize all samples and estimate probabilities for them. Moreover, the ability of T to detect unknown samples needs to be transferred to C_1. Since T is derived from using gradient descent to solve $\min_{h^{(1)}_{K+1}(x)} H_{\text{sym}}(p^{(1)}_{:K}(x), p^{\text{mul}}(x))$, we include this in training to estimate probabilities adaptively and ensure Equation(6) holds. We define the loss as

$$\begin{aligned} \mathcal{L}_{\text{scrs}} = -\frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{j=1}^{K} p^{(1)}_j(x^t_i) \log p^{\text{mul}}_j(x^t_i) \\ -\frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{j=1}^{K} p^{\text{mul}}_j(x^t_i) \log p^{(1)}_j(x^t_i). \end{aligned} \tag{15}$$

3.6 Self-Supervised Clustering

To improve accuracy, samples should be well-clustered in $\mathcal{Z}$, i.e. distances between samples of the same class are smaller than those of different classes. $\mathcal{L}_{\text{dis}}$ distills source knowledge to cluster target samples. Due to domain-shift and category-shift, it's vital to mine the categorical structure of $\mathcal{D}_t$.

Referring to UB2DA (Deng et al. 2021), we define K_{clu} learnable prototypes $W_t = \{w^t_1, \ldots, w^t_{K_{\text{clu}}}\}$ as clustering centers of D_t which are initialized by k-Means. K_{clu} is set manually. We let each sample find its most similar center. Calculate the cosine similarity between sample $z = f(x)$ and w^t_j as $s_j(z) = \frac{<z, w^t_j>}{||z||_2 \cdot ||w^t_j||_2}$. Then $p^t_j(x) = \frac{\exp(s_j(z))}{\sum_{j=1}^{K_{\text{clu}}} \exp(s_j(z))}$, $j = 1, \ldots, K_{\text{clu}}$ are clustering probabilities of x.

To make the clustering boundaries clear and avoid the domination of large clusters in feature learning, we follow similar strategies of DEC (Xie, Girshick, and Farhadi 2016) and DEPICT (Dizaji et al. 2017) to construct self-supervised information. For each $p^t_j(x)$, define $q^t_j(x) = \frac{p^t_j(x)/(\sum_l p^t_j(x^t_l))^{0.5}}{\sum_{m=1}^{K_{\text{clu}}} [p^t_m(x)/(\sum_l p^t_m(x^t_l))^{0.5}]}$. The clustering loss is

$$\mathcal{L}_{\text{reg}} = -\frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{j=1}^{K_{\text{clu}}} q^t_j(x^t_i) \log p^t_j(x^t_i). \tag{16}$$

3.7 Balancing Mechanism

Multi-category Equilibrium Based on Sample Division. To avoid only predicting few categories, previous works mainly let the mean of probabilities approach the uniform vector (Liang et al. 2021; Ma, Gao, and Xu 2021). However, since C_1 includes the unknown category, the mean should be calculated separately for unknown and known sample sets.

We first divide D_t into an unknown candidate set $D_{\text{t-un}}$, and a non-candidate one $D^c_{\text{t-un}} = D_t - D_{\text{t-un}}$. Then we let the average probability of $D^c_{\text{t-un}}$ output by C_1 to be close to the uniform vector. The unknown category is considered since there may be misclassified unknown samples. To better describe the predicted label structure of $D^c_{\text{t-un}}$, we define $q^{\text{kn}(1)}_j(x) = \frac{p^{(1)}_j(x)/(\sum_{x \in D^c_{\text{t-un}}} p^{(1)}_j(x))^{0.5}}{\sum_{l=1}^{K+1} [p^{(1)}_l(x)/(\sum_{x \in D^c_{\text{t-un}}} p^{(1)}_l(x))^{0.5}]}$, $\bar{q}^{\text{kn}(1)}_j =$

$\frac{1}{N^c_{\text{t-un}}} \sum_{x \in D^c_{\text{t-un}}} q_j^{\text{kn}(1)}(x)$. KL divergence is used as

$$\mathcal{L}^{(1)}_{\text{avr}}(D^c_{\text{t-un}}) = \sum_{j=1}^{K+1} \bar{q}_j^{\text{kn}(1)} \log \bar{q}_j^{\text{kn}(1)} + \log(K+1). \tag{17}$$

Since $p^{(1)}_{K+1}(x) \geq \max_{j \neq K} p^{(1)}_j(x)$ always holds for $x \in D_{\text{t-un}}$, applying Equation(17) to $D_{\text{t-un}}$ directly is improper. However, when not considering unknown classes, probabilities predicted are uncertain and the average probability is close to the uniform vector. Consider $\tilde{p}^{(1)}_j(x^t_i) = p^{(1)}_j(x^t_i)/(1 - p^{(1)}_{K+1}(x^t_i))$. Let $q_j^{\text{un}(1)}(x) = \frac{\tilde{p}_j^{(1)}(x)/(\sum_{x \in D_{\text{t-un}}} \tilde{p}_j^{(1)}(x))^{0.5}}{\sum_{l=1}^{K}[\tilde{p}_l^{(1)}(x)/(\sum_{x \in D_{\text{t-un}}} \tilde{p}_l^{(1)}(x))^{0.5}]}$ and $\bar{q}_j^{\text{un}(1)} = \frac{1}{N_{\text{t-un}}} \sum_{x \in D_{\text{t-un}}} q_j^{\text{un}(1)}(x)$. Use the KL divergence as

$$\mathcal{L}^{(1)}_{\text{avr}}(D_{\text{t-un}}) = \sum_{j=1}^{K} \bar{q}_j^{\text{un}(1)} \log \bar{q}_j^{\text{un}(1)} + \log(K). \tag{18}$$

When the class number is larger than the batch size, each category in the batch may contain inadequate samples for learning. It may also take many iterations to learn the same category again, leading to model forgetting. Let $\omega_{\text{dyn}} = \frac{\max(K, K_{\text{est}})}{b_t}$, where K_{est} is the estimated target class number and defined below. Such condition can be described as $\omega_{\text{dyn}} > 1$, in which the model needs to pay attention to all categories and the degree is determined by $\frac{K}{b_t}$. When $\omega_{\text{dyn}} < 1$, balancing from the point of $D^c_{\text{t-un}}$ and $D_{\text{t-un}}$ is enough. Let $q_j^{(1)}(x) = \frac{p_j^{(1)}(x)/(\sum_{x \in B_t} p_j^{(1)}(x))^{0.5}}{\sum_{l=1}^{K+1}[p_l^{(1)}(x)/(\sum_{x \in B_t} p_l^{(1)}(x))^{0.5}]}$, $\bar{q}_j^{(1)} = \frac{1}{b_t} \sum_{x \in B_t} q_j^{(1)}(x)$. and $1_{\{\cdot\}}$ be the indicator function. Using KL divergence, the overall balance loss is

$$\mathcal{L}^{(1)}_{\text{full}}(B_t) = 1_{\{\omega_{\text{dyn}} > 1\}} \frac{K}{b_t} \left[\sum_{j=1}^{K+1} \bar{q}_j^{(1)} \log \bar{q}_j^{(1)} + \log(K+1) \right]. \tag{19}$$

In summary, we balance the categorical learning from the view of C_1 with $\mathcal{L}^{(1)}_{\text{avr}} = \mathcal{L}^{(1)}_{\text{avr}}(D^c_{\text{t-un}}) + \mathcal{L}^{(1)}_{\text{avr}}(D_{\text{t-un}}) + \mathcal{L}^{(1)}_{\text{full}}(B_t)$.

Since C_2 doesn't include the unknown category, we let the average probability output by C_2 to approach the uniform vector. Define $q_j^{(2)}(x) = \frac{p_j^{(2)}(x)/(\sum_{x \in B_t} p_j^{(2)}(x))^{0.5}}{\sum_{l=1}^{K}[p_l^{(2)}(x)/(\sum_{x \in B_t} p_l^{(2)}(x))^{0.5}]}$ and $\bar{q}_j^{(2)} = \frac{1}{b_t} \sum_{x \in B_t} q_j^{(2)}(x)$, $j = 1, \ldots, K$. Still using KL divergence, the balance loss viewed from C_2 is

$$\mathcal{L}^{(2)}_{\text{avr}} = \sum_{j=1}^{K} \bar{q}_j^{(2)} \log \bar{q}_j^{(2)} + \log(K). \tag{20}$$

The loss of multi-category equilibrium is $\mathcal{L}_{\text{avr}} = \mathcal{L}^{(1)}_{\text{avr}} + \mathcal{L}^{(2)}_{\text{avr}}$.

Equilibrium Coefficient. $\mathcal{L}_{\text{avr}}$ pays equal attention to the unknown category and each known class. But the existence of domain-private classes affects its necessity. Therefore, we measure the relative size of target and source classes.

To estimate the target class number K_t, we identify target prototypes that correspond one-to-one with their pseudo labels by clustering entropy. For w^t_i, the clustering entropy is $H(p^t(w^t_i)) = -\sum_{j=1}^{K_{\text{clu}}} p^t_j(w^t_i) \log p^t_j(w^t_i)$. Compare it with threshold $e_0 > 0$. We estimate K_t as $K_{\text{est}} = |\{w^t_i | H\left(p^t(w^t_i)\right) \leq e_0\}|$. K_{est} may be small at the beginning and converges to a constant as the clustering stabilizes.

Let $\gamma_t = \frac{K_{\text{est}}}{K}$. If $\gamma_t > 1$, the larger $\gamma_t - 1$ is, the more likely there are unknown classes and known classes need more attention. If $\gamma_t < 1$, the smaller γ_t is, the more likely there are source-private classes which need less attention.

Since the model tends to learn large-scale or easily distinguished classes first, it should focus on each class equally in the early stage rather than considering γ_t. Note that if $\omega_{\text{dyn}} > 1$, samples to learn each category in the batch are inadequate and forgetting may occur. The larger ω_{dyn} is, the more attention should be paid to equal balance. Therefore, we use $e_{\text{thr}} = \omega_{\text{dyn}} \cdot epoch_{\text{total}}$ as the threshold of epoch. The weight of $\mathcal{L}_{\text{avr}}$ at the e-th epoch is defined as

$$\omega_{\text{avr}} = \begin{cases} 1 & \text{if } e \leq e_{\text{thr}}, \\ (\gamma_t - 1)1_{\{\gamma_t > 1\}} + \gamma_t 1_{\{\gamma_t \leq 1\}} & \text{if } e > e_{\text{thr}}. \end{cases} \tag{21}$$

Binary Equilibrium. In $\mathcal{L}_{\text{t-kn}}$ and $\mathcal{L}_{\text{t-un}}$, we learn categorical knowledge from $D_{\text{t-kn}}$ and $D_{\text{t-un}}$. Denote sample sizes of them as $N_{\text{t-kn}}$ and $N_{\text{t-un}}$. If $N_{\text{t-kn}} > N_{\text{t-un}}$, the ability of detecting unknown classes may be weakened. If $N_{\text{t-kn}} < N_{\text{t-un}}$, the model may not learn each known category well. Besides, D_{high} and D_{low} affect the unknown detection through $\mathcal{L}_{\text{scrs}}$. Denote N_{high} and N_{low} as their sample sizes. Let

$$\gamma_{\text{train}} = \begin{cases} \frac{N_{\text{t-kn}}}{N_{\text{t-un}}} & \text{if } N_{\text{t-un}} > 0, \\ 1 & \text{if } N_{\text{t-un}} = 0. \end{cases} \gamma_{\text{thr}} = \begin{cases} \frac{N_{\text{high}}}{N_{\text{low}}} & \text{if } N_{\text{low}} > 0, \\ 1 & \text{if } N_{\text{low}} = 0. \end{cases} \tag{22}$$

$\gamma_{\text{train}} \cdot \gamma_{\text{thr}}$ represents the ratio of the number of samples will be labelled known to that of those will be annotated unknown. It's a measure of current training status through sample selection. However, $\gamma_{\text{train}} \cdot \gamma_{\text{thr}} = 1$ isn't the ideal status for all datasets or DA types. Thus we estimate the initial binary relationship and use it as a reference for training status. Calculate $\{T^{\text{pre}}(x^t_i)\}_{i=1}^{N_t}$ to obtain $D^{\text{pre}}_{\text{high}} = \{x \in D_t | T^{\text{pre}}(x) > 0\}$ and $D^{\text{pre}}_{\text{low}} = \{x \in D_t | T^{\text{pre}}(x) < 0\}$ with sizes as $N^{\text{pre}}_{\text{high}}$ and $N^{\text{pre}}_{\text{low}}$. Let $\gamma^{\text{pre}}_{\text{thr}} = \begin{cases} \frac{N^{\text{pre}}_{\text{high}}}{N^{\text{pre}}_{\text{low}}} & \text{if } N^{\text{pre}}_{\text{low}} > 0, \\ 1 & \text{if } N^{\text{pre}}_{\text{low}} = 0. \end{cases}$ Due to shifts, $\gamma^{\text{pre}}_{\text{thr}}$ needs adjustments to be used as a reference. Let

$$\tilde{\gamma}^{\text{pre}}_{\text{thr}} = \begin{cases} (\gamma^{\text{pre}}_{\text{thr}})^{\gamma^{\text{pre}}_{\text{thr}}} & \text{if } \gamma^{\text{pre}}_{\text{thr}} \leq \log(K), \\ (\gamma^{\text{pre}}_{\text{thr}})^{\log(K)} & \text{if } \gamma^{\text{pre}}_{\text{thr}} > \log(K). \end{cases} \tag{23}$$

Such adjustment makes sense when $\gamma^{\text{pre}}_{\text{thr}}$ is large or small. When K is large, N_s tends to be large, leading to large $\gamma^{\text{pre}}_{\text{thr}}$. $(\gamma^{\text{pre}}_{\text{thr}})^{\gamma^{\text{pre}}_{\text{thr}}}$ can be too large to detect potential unknown classes. If $\gamma^{\text{pre}}_{\text{thr}}$ is small, exponentization amplifies its impact.

If $\frac{\gamma_{\text{train}} \cdot \gamma_{\text{thr}}}{\tilde{\gamma}^{\text{pre}}_{\text{thr}}} > 1$, more samples will be viewed as known and more emphasis should be put on unknown detection. Contrarily, the model should pay attention to known classes.

Combining the estimate of DA type through γ_t, we define $\omega_{\text{un}} = \gamma_t \cdot \frac{\gamma_{\text{train}} \cdot \gamma_{\text{thr}}}{\tilde{\gamma}^{\text{pre}}_{\text{thr}}}$. The relative attention required for unknown classes to known classes grows with ω_{un}. Thus we use $\tilde{\omega}_{\text{un}} = \frac{\omega_{\text{un}}}{\omega_{\text{un}}+1}$ and $1 - \tilde{\omega}_{\text{un}}$ as weights of $\mathcal{L}_{\text{t-un}}$ and $\mathcal{L}_{\text{t-kn}}$.

3.8 Total Loss of Training on the Target Domain

Source information is necessary for annotation but reliance on it should be gradually reduced due to shifts. Besides, binary balance ought to be kept during training. Thus we define $\alpha = \max(1 - \frac{e}{epoch_{\text{total}}}, 0)$, $\omega_\alpha = (\alpha)^{\omega_{\text{un}}}$ and $\omega_\alpha^{\text{kn}} = (1-\alpha)^{\omega_{\text{un}}}$ at the $e-$th epoch. The total loss on $\mathcal{D}_t$ is

$$\begin{aligned}\mathcal{L}_t = \mathcal{L}_{\text{reg}} + \omega_{\text{avr}}\mathcal{L}_{\text{avr}} + \omega_\alpha \mathcal{L}_{\text{dis}} + \omega_\alpha^{\text{kn}}\left(1-\tilde{\omega}_{\text{un}}\right)\mathcal{L}_{\text{t-kn}} \\ + \left(1-\omega_\alpha\right)\left(\tilde{\omega}_{\text{un}}\mathcal{L}_{\text{t-un}} + \mathcal{L}_{\text{scrs}}\right).\end{aligned} \tag{24}$$

3.9 Inference

The prediction of x is $\hat{y} = \arg\max_{j=1,\ldots,K+1} p_j^{(1)}(x)$. If $\hat{y} = K+1$, then x is predicted to be in unknown classes.

4 Experiments

4.1 Setup

Datasets. We use four benchmark datasets in DA: Office (Saenko et al. 2010), OfficeHome (OH) (Venkateswara et al. 2017), VisDA (Peng et al. 2017) and DomainNet (Peng et al. 2019). We follow existing methods (Saito and Saenko 2021; Deng et al. 2021) to split datasets and the split $|L_s - L_t|/|L_s \cap L_t|/|L_t - L_s|$ is showed in each table.

Evaluation Metric. For CDA and PDA, we evaluate the mean of known class accuracy. For ODA and OPDA, we use H-score (Fu et al. 2020) to evaluate the trade-off between the accuracy of known and unknown classes. Let acc_{com} be the mean of known class accuracy and acc_{un} be the unknown accuracy. H-score can be written as $\frac{2acc_{\text{com}} \cdot acc_{\text{un}}}{acc_{\text{com}} + acc_{\text{un}}}$.

Implementation. We use ResNet50 (He et al. 2016) pre-trained on ImageNet (Deng et al. 2009) as the backbone network. For Office, OfficeHome, VisDA and DomainNet, we set $K_{\text{clu}} = 35$, 70, 20 and 300 respectively under all DA settings. Besides, we let $e_0 = 0.1$ in all experiments.

Baselines. We reproduce results of experiments where codes and default parameters can be obtained. For B^2-UniDA method, we reproduce UB2DA (Deng et al. 2021). For SF-UniDA methods, we choose USFDA (Kundu et al. 2020) and UMAD (Liang et al. 2021). But USFDA is computationally heavy due to simulation. Codes of UMAD are inaccessable. We report raw results of them. We also reproduce typical UniDA methods including UAN (You et al. 2019), CMU (Fu et al. 2020), DANCE (Saito et al. 2020), DCC (Li et al. 2021) and OVANet (Saito and Saenko 2021). UAN and CMU highly rely on hyperparameters. Results using default parameters are not ideal. We use results after tuning parameters which are slightly better. Labelling samples in $D_{\text{low}}^{\text{pre}}$ as unknown and others with $\hat{y}^{(1)}$, we obtain predictions after pre-training denoted as source-only (SO).

4.2 Results

OPDA. Table 1 and Table 2 show that performance of GSS is significantly improved by training compared to SO. GSS outperforms UB2DA in all experiments. The average improvements of experiments on Office, OfficeHome, VisDA and DomainNet are 2.6%, 11.2%, 70.9% and 8.6% respectively. GSS even surpasses SF-UniDA and UniDA methods in all experiments on OfficeHome. Moreover, GSS performs comparatively with best results of SF-UniDA and UniDA methods on Office and DomainNet.

Type	Method	Office (10/10/11)							VisDA
		A2W	D2W	W2D	A2D	D2A	W2A	Avg	(6/3/3)
Uni	UAN	38.8	67.3	61.5	33.1	70.4	56.4	54.6	13.7
	CMU	*83.6*	51.8	54.7	50.4	73.7	64.6	63.1	25.5
	DANCE	70.8	91.4	89.6	77.9	*79.0*	71.7	80.1	4.0
	DCC	76.3	86.0	84.7	*85.1*	73.5	82.4	81.3	*43.9*
	OVANet	79.9	*95.2*	*95.5*	84.5	77.5	*82.6*	*85.9*	30.0
SF-Uni	USFDA	85.5	79.8	83.2	90.6	81.2	88.7	84.9	48.8
	UMAD	77.4	90.7	97.2	79.1	87.4	90.4	87.0	58.3
B^2-Uni	UB2DA	77.7	91.4	85.8	78.3	91.3	85.3	85.0	25.5
	SO	66.6	91.0	86.6	67.0	75.1	79.7	77.7	17.0
	GSS	**80.9**	**92.0**	**89.4**	**81.1**	**91.4**	**88.1**	**87.1**	**43.5**

Table 1: H-score (%) on Office and VisDA in OPDA. Uni is short for UniDA. Italic, underlined and bold numbers represent maxima within Uni, SF-Uni and B^2-Uni methods.

ODA. In the appendix, GSS also surpasses UB2DA in all experiments. Average improvements on Office and OfficeHome are 9.0% and 9.2%. H-score of GSS on VisDA is almost 2.8 times that of UB2DA. Besides, the average H-score of GSS on Office and OfficeHome are 3.0% and 4.9% larger than the best results of UniDA methods and 1.8% and 3.3% than SF-UniDA ones.

PDA and CDA. In the appendix, measured by the average accuracy, GSS outperforms UB2DA on Office, OfficeHome and VisDA by 10.9%, 16.8% and 6.9% respectively in PDA. In CDA, GSS achieves average accuracy gains of 11.0% and 6.0% on Office and OfficeHome. The accuracy of GSS on VisDA is about 2.8 times that of UB2DA. Besides, GSS surpasses all UniDA methods except DANCE considering average accuracy on Office in CDA and PDA.

4.3 Experiment Analysis

Ablation Study. We conduct ablation studies by removing each component in GSS. Table 3 shows that $\mathcal{L}_{\text{full}}^{(1)}$ is vital for class-balance when K is large and ω_{avr} is important when K isn't so large. Without deteriorating the performance of GSS on any dataset, no component can be removed. Combined with the appendix, we find $\mathcal{L}_{\text{reg}}$ and $\mathcal{L}_{\text{pse}}^{(2)}$ help detect unknown samples, while $\mathcal{L}_{\text{t-kn}}$ and $\mathcal{L}_{\text{p-kn}}^{(1)}$ are vital for known class categorization. $\mathcal{L}_{\text{scrs}}$, $\mathcal{L}_{\text{avr}}$, $\mathcal{L}_{\text{full}}^{(1)}$ and ω_{un} balance both.

Validity of the Sample Selection Criterion. We select different thresholds for metrics in CMU and DANCE for comparision with SO after pre-training. Conduct experiments on OfficeHome in OPDA. Figure 3 and the appendix show that our criterion performs comparatively with the best results of CMU and entropy used by DANCE and UB2DA.

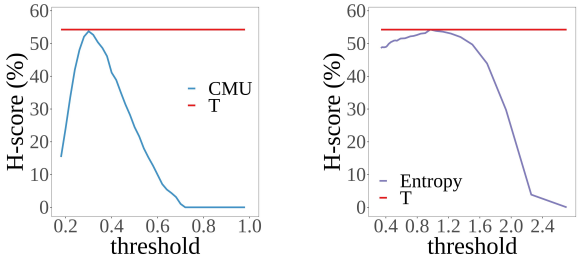
Sensitivity Analysis of e_0. We study the parameter sensitivity of e_0 on OfficeHome (OH) under all DA settings. From Table 4, we find that using different e_0 has little influence.

Type	Method	OfficeHome (10/5/50)													DomainNet(150/50/145)						
		A2C	A2P	A2R	C2A	C2P	C2R	P2A	P2C	P2R	R2A	R2C	R2P	Avg	P2R	R2P	P2S	S2P	R2S	S2R	Avg
Uni	UAN	14.7	13.3	15.0	18.8	11.5	15.0	12.1	9.5	16.9	15.0	7.4	8.5	13.1	41.9	43.6	39.1	39.0	38.7	43.7	41.0
	CMU	42.2	55.7	60.0	39.5	45.9	54.1	61.2	42.1	67.7	64.0	51.3	62.0	53.8	48.1	47.6	41.0	36.6	43.9	48.0	44.2
	DANCE	40.6	41.3	34.5	43.5	27.5	36.2	54.4	49.7	41.5	26.7	42.5	52.1	40.9	21.0	47.3	37.0	27.7	*46.7*	21.0	33.5
	DCC	*62.9*	75.3	*81.3*	28.7	*73.2*	*82.4*	68.7	58.5	*83.3*	74.9	60.0	*81.7*	69.2	53.4	47.8	34.6	32.1	17.4	48.9	39.0
	OVANet	62.1	*76.3*	80.0	*71.0*	69.7	75.8	*73.0*	*60.1*	80.0	*75.7*	*63.6*	79.2	*72.2*	*55.8*	*51.8*	*47.1*	*47.9*	44.6	*56.3*	*50.6*
SF-Uni	UMAD	61.1	76.3	82.7	70.7	67.7	75.7	64.4	55.7	76.3	73.2	60.4	77.2	70.1	59.0	50.1	44.3	32.0	42.1	55.3	47.1
B^2-Uni	UB2DA	61.6	68.3	77.3	72.9	68.8	76.7	70.7	60.8	75.5	72.6	63.0	71.0	69.9	56.7	49.8	45.9	33.9	43.1	51.4	46.8
	SO	58.0	66.2	71.5	66.7	62.3	70.4	69.1	54.0	70.2	68.8	58.4	67.1	65.2	50.8	46.0	44.4	36.1	42.2	45.8	44.2
	GSS	**67.9**	**83.2**	**86.4**	**76.4**	**76.6**	**85.5**	**77.6**	**65.2**	**87.0**	**78.7**	**67.6**	**82.1**	**77.9**	**59.1**	**52.1**	**46.7**	**43.9**	**45.0**	**55.2**	**50.3**

Table 2: H-score (%) on OfficeHome and DomainNet in OPDA. Uni is short for UniDA. Italic and bold numbers represent maximum values within UniDA and B^2-UniDA methods respectively.

Method	Office	OfficeHome	VisDA	DomainNet
GSS w/o $\mathcal{L}_{\text{reg}}$	0.0	0.0	0.0	48.8
GSS w/o $\mathcal{L}_{\text{t-kn}}$	84.9	76.4	23.1	40.1
GSS w/o $\mathcal{L}^{(1)}_{\text{p-kn}}$	84.9	76.8	35.7	49.3
GSS w/o $\mathcal{L}^{(1)}_{\text{p-un}}$	0.0	1.8	0.0	13.3
GSS w/o $\mathcal{L}^{(2)}_{\text{pse}}$	83.2	73.7	33.5	49.2
GSS w/o $\mathcal{L}_{\text{scrs}}$	84.4	75.9	30.0	35.8
GSS w/o $\mathcal{L}_{\text{avr}}$	84.7	72.9	27.8	12.7
GSS w/o $\mathcal{L}^{(1)}_{\text{full}}$	87.1	77.9	43.5	33.6
GSS w/o ω_{un}	83.3	72.7	31.7	0.0
GSS w/o ω_{avr}	82.2	76.5	43.0	50.3
GSS (full)	**87.1**	**77.9**	**43.5**	**50.3**

Table 3: Ablation Study. The mean of H-score (%) on each dataset in OPDA with different variants of GSS.



(a) CMU and T (b) Entropy and T

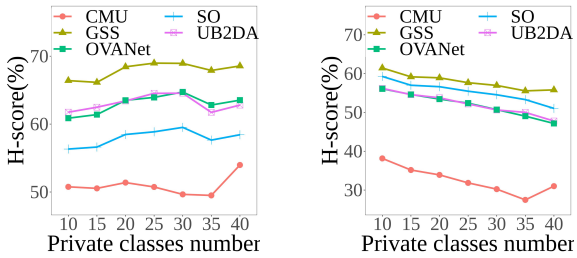
Figure 3: H-score of Product2Clipart in OPDA

e_0	Acc.(%)		H-score(%)	
	CDA	PDA	ODA	OPDA
0.01	51.5	55.2	68.8	77.7
0.05	51.5	55.6	68.4	77.8
0.1	51.5	55.8	68.6	77.9
0.2	51.4	55.8	68.4	77.8
0.4	51.4	55.8	68.5	77.7
0.6	51.4	55.9	68.3	78.0

Table 4: Vary e_0 on OH.

K_{clu}	Acc.(%)		H-score(%)	
	CDA	PDA	ODA	OPDA
50	51.4	54.7	68.2	77.6
60	51.4	54.6	68.2	77.8
70	51.5	55.8	68.6	77.9
80	51.4	55.1	69.3	78.0
90	51.4	54.6	69.2	78.0
100	51.3	54.4	68.5	78.0

Table 5: Vary K_{clu} on OH.



(a) Avr. H-score in ODA (b) Avr. Acc. in PDA

Figure 4: Vary domain-private class number on OfficeHome.

The main reason is when clustering becomes stable, i.e. $\mathcal{L}_{\text{reg}}$ converges, entropies of non-redundant prototypes are small.

Sensitivity Analysis of K_{clu}. We vary K_{clu} on OfficeHome (OH) under all settings. Since prototypes are used for clustering rather than alignment, varying K_{clu} has minor influence in Table 5. Large K_{clu} may conducive to mining unknown classes as they can be distinguished. But larger K_{clu} takes more time to cluster. When K and N_t are large, we advise choosing K_{clu} close to K, otherwise larger than K.

Varying Domain-private Classes. Negative transfer may be severe as source-private classes increase. The imbalance between unknown and known classes will be exacerbated as unknown classes increase. Therefore, we compare GSS with UB2DA, SO, CMU and OVANet in ODA and PDA when varying domain-private classes on OfficeHome. Figure 4 shows GSS consistently outperforms others.

5 Conclusion

In this paper, we propose a **G**radient-based **S**ample **S**election method (GSS) for B^2-UniDA. Our sample selection criterion is derived from gradient descent and Bayes' theorem. It doesn't need to select thresholds when data changes. We also propose a balancing mechanism to avoid imbalanced learning especially imbalance between the known and unknown classes. Overall target training modules include knowledge distillation, self-training, adaptative estimation, clustering and balancing. Superiority of GSS is verified by experiments.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (32270689) and the National Key Research and Development Program of China (2021YFF1200902).

References

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A theory of learning from different domains. *Mach. Learn.*, 79(1–2): 151–175.

Busto, P. P.; and Gall, J. 2017. Open Set Domain Adaptation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 754–763. Piscataway, NJ: IEEE.

Cai, Z.; Song, J.; Zhang, T.; Jing, X.-Y.; and Shao, L. 2021. Dual Contrastive Universal Adaptation Network. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. Piscataway, NJ: IEEE.

Cao, Z.; Long, M.; Wang, J.; and Jordan, M. I. 2018a. Partial Transfer Learning with Selective Adversarial Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2724–2732. Piscataway, NJ: IEEE.

Cao, Z.; Ma, L.; Long, M.; and Wang, J. 2018b. Partial Adversarial Domain Adaptation. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*, 139–155. Berlin, Heidelberg: Springer-Verlag.

Chen, L.; Du, Q.; Lou, Y.; He, J.; Bai, T.; and Deng, M. 2022a. Mutual Nearest Neighbor Contrast and Hybrid Prototype Self-Training for Universal Domain Adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36: 6248–6257.

Chen, L.; Lou, Y.; He, J.; Bai, T.; and Deng, M. 2022b. Evidential Neighborhood Contrastive Learning for Universal Domain Adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36: 6258–6267.

Chen, L.; Lou, Y.; He, J.; Bai, T.; and Deng, M. 2022c. Geometric Anchor Correspondence Mining With Uncertainty Modeling for Universal Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16134–16143. Piscataway, NJ: IEEE.

Csiszar, I. 1975. I-Divergence Geometry of Probability Distributions and Minimization Problems. *The Annals of Probability*, 3(1): 146–158.

Deng, B.; Zhang, Y.; Tang, H.; Ding, C.; and Jia, K. 2021. On Universal Black-Box Domain Adaptation. arXiv:2104.04665.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248–255. Piscataway, NJ: IEEE.

Dizaji, K. G.; Herandi, A.; Deng, C.; Cai, W.; and Huang, H. 2017. Deep Clustering via Joint Convolutional Autoencoder Embedding and Relative Entropy Minimization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 5747–5756. Piscataway, NJ: IEEE.

Fu, B.; Cao, Z.; Long, M.; and Wang, J. 2020. Learning to Detect Open Classes for Universal Domain Adaptation. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 567–583. Cham: Springer International Publishing. ISBN 978-3-030-58555-6.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised Domain Adaptation by Backpropagation. In Bach, F.; and Blei, D., eds., *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 1180–1189. Lille, France: PMLR.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. Piscataway, NJ: IEEE.

Hu, H.; Salcic, Z.; Sun, L.; Dobbie, G.; Yu, P. S.; and Zhang, X. 2022. Membership Inference Attacks on Machine Learning: A Survey. *ACM Comput. Surv.*, 54(11s): 1–37.

Johnson, J.; and Khoshgoftaar, T. 2019. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1): 1–54.

Joyce, J. 2021. Bayes' Theorem. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2021 edition.

Kundu, J. N.; Venkat, N.; V, R. M.; and Babu, R. V. 2020. Universal Source-Free Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE.

Lemaréchal, C. 2012. Cauchy and the Gradient Method. *DOCUMENTA MATHEMATICA*, Extra Volume: 251–254.

Li, G.; Kang, G.; Zhu, Y.; Wei, Y.; and Yang, Y. 2021. Domain Consensus Clustering for Universal Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9757–9766. Piscataway, NJ: IEEE.

Liang, J.; Hu, D.; Feng, J.; and He, R. 2021. UMAD: Universal Model Adaptation under Domain and Category Shift. arXiv:2112.08553.

Ma, X.; Gao, J.; and Xu, C. 2021. Active Universal Domain Adaptation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 8948–8957. Piscataway, NJ: IEEE.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment Matching for Multi-Source Domain Adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Piscataway, NJ: IEEE.

Peng, X.; Usman, B.; Kaushik, N.; Hoffman, J.; Wang, D.; and Saenko, K. 2017. VisDA: The Visual Domain Adaptation Challenge. arXiv:1710.06924.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, 213–226. Berlin, Heidelberg: Springer-Verlag.

Saito, K.; Kim, D.; Sclaroff, S.; and Saenko, K. 2020. Universal Domain Adaptation through Self Supervision. In

Advances in Neural Information Processing Systems, volume 33, 16282–16292. New York: Curran Associates, Inc.

Saito, K.; and Saenko, K. 2021. OVANet: One-vs-All Network for Universal Domain Adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 9000–9009. Piscataway, NJ: IEEE.

Saito, K.; Yamamoto, S.; Ushiku, Y.; and Harada, T. 2018. Open Set Domain Adaptation by Backpropagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Berlin, Heidelberg: Springer-Verlag.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep Hashing Network for Unsupervised Domain Adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5385–5394. Piscataway, NJ: IEEE.

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; and Bengio, Y. 2019. Manifold Mixup: Better Representations by Interpolating Hidden States. In *Proceedings of the 36th International Conference on Machine Learning*, 6438–6447. New York, USA: PMLR.

Wang, Z.; Dai, Z.; Poczos, B.; and Carbonell, J. 2019. Characterizing and Avoiding Negative Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE.

Wilson, G.; and Cook, D. 2020. A Survey of Unsupervised Deep Domain Adaptation. *ACM Trans. Intell. Syst. Technol.*, 11(5): 1–46.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised Deep Embedding for Clustering Analysis. In Balcan, M. F.; and Weinberger, K. Q., eds., *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, 478–487. New York, New York, USA: PMLR.

Yin, Y.; Yang, Z.; Wu, X.; and Hu, H. 2021. Pseudo-margin-based universal domain adaptation. *Knowledge-Based Systems*, 229(1–2): 107315.

You, K.; Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2019. Universal Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2715–2724. Piscataway, NJ: IEEE.

Yu, Q.; Hashimoto, A.; and Ushiku, Y. 2021. Divergence Optimization for Noisy Universal Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2515–2524. Piscataway, NJ: IEEE.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018a. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*. Washington, DC: ICLR Press.

Zhang, J.; Ding, Z.; Li, W.; and Ogunbona, P. 2018b. Importance Weighted Adversarial Nets for Partial Domain Adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8156–8164. Piscataway, NJ: IEEE.

Zhang, Y.; Jia, R.; Pei, H.; Wang, W.; Li, B.; and Song, D. 2020. The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE.