Unleashing the Potential of Adaptation Models via Go-getting Domain Labels

Xin Jin[1(✉)], Tianyu He[2], Xu Shen[2], Songhua Wu[3], Tongliang Liu[3], Jingwen Ye[4], Xinchao Wang[4], Jianqiang Huang[2], Zhibo Chen[5], and Xian-Sheng Hua[2]

[1] Eastern Institute for Advanced Study, Ningbo, China
jinxin@eias.ac.cn
[2] Alibaba Group, Hangzhou, China
[3] The University of Sydney, Sydney, Australia
[4] National University of Singapore, Singapore, Singapore
[5] University of Science and Technology of China, Hefei, China

Abstract. In this paper, we propose an embarrassingly simple yet highly effective adversarial domain adaptation (ADA) method. We view ADA problem primarily from an optimization perspective and point out a fundamental dilemma, in that the real-world data often exhibits an imbalanced distribution where the large data clusters typically dominate and bias the adaptation process. Unlike prior works that either attempt loss re-weighting or data re-sampling for alleviating this defect, we introduce a new concept of go-getting domain labels (Go-labels) to replace the original immutable domain labels on the fly. The reason why call it as "Go-labels" is because "go-getting" means able to deal with new or difficult situations easily, like here Go-labels adaptively transfer the model attention from over-studied aligned data to those overlooked samples, which allows each sample to be well studied (*i.e.*, alleviating data imbalance influence) and fully unleashes the potential of adaption model. Albeit simple, this dynamic adversarial domain adaptation framework with Go-labels effectively addresses data imbalance issue and promotes adaptation. We demonstrate through theoretical insights, empirical results on real data as well as toy games that our method leads to efficient training without bells and whistles, while being robust to different backbones.

1 Introduction

Most deep models rely on huge amounts of labeled data and their learned features have proven brittle to data distribution shifts [58,68]. To mitigate the data discrepancy issue and reduce dataset bias, unsupervised domain adaptation (UDA) is extensively explored, which has access to labeled samples from a source domain and unlabeled data from a target domain. Its objective is to train a model that generalizes well to the target domain [8,10,14,15,19,24,25,28].

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-25085-9_18.

L. Karlinsky et al. (Eds.): ECCV 2022 Workshops, LNCS 13808, pp. 308–325, 2023.
https://doi.org/10.1007/978-3-031-25085-9_18

As a mainstream branch of UDA, adversarial domain adaptation (ADA) approaches leverage a domain discriminator paired with a feature generator to adversarially learn a domain-invariant feature [9,11,15,35,50]. For the domain discriminator training, all source data are equally taken as one domain (*e.g.*, positive '1') while target data as another one (*e.g.*, negative '0') [11,15,35]. However, this fixed positive-negative separation neglects a fact that most real-world data exhibit imbalanced distributions [12,13]: the clusters with abundant examples (*i.e.*, large clusters) may **swamp** the clusters with few examples (*i.e.*, small clusters). Such imbalanceness contains two aspects, intra-class long-tailed distribution [34,44] and inter-class long-tailed distribution [55,64], and is widely existed in many UDA benchmarks. For example, in DomainNet [42], the "dog" class in the "clipart" domain has 70 image samples while has 782 image samples in the "real" domain. The majority "bike" samples (90%) in "Amazon" domain in Office31 [46] have no background scene (empty) while minority "bike" samples have real-world background instead.

On the other hand, deep neural networks (DNNs) typically learn simple patterns first before memorizing. In other words, DNN optimization is content-aware, taking advantage of patterns shared by multiple training examples [2]. Therefore, in the process of domain adaptation, the large domain clusters would dominate the optimization of domain discriminator, so that bias its decision boundary and hinder the effective adaptation. As shown in Fig. 1(a), only the large clusters of two domains (*i.e.*, two large circles) have been pulled close as the adaptation goes on, but those minority clusters (four small circles) are still under-aligned. This bias the optimization of domain discriminator so that misleads the feature extractor to learn unexpected domain-specific knowledge from large clusters. As a result, the adapted model still can not correctly classify these under-explored samples (marked by "misclassify").

In this paper, we attempt to design an optimization strategy to progressively take full advantage of both large and small data clusters across different domains,

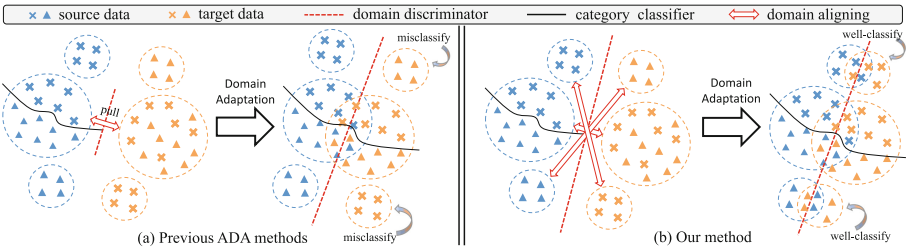


Fig. 1. Motivation illustration. (×, △) denote two different classes, and (blue, orange) color mean different domains. (a) Previous DA methods tend to be dominated by those large clusters and neglects small clusters, which will bias the domain discriminator optimization, leading to a sub-optimal adaptation accuracy. (b) Our method attempts to fully leverage both large and small data clusters for alignment, to enhance the domain-invariant representation learning, and thus achieving a better adaptation performance on the target set. (Color figure online)

like shown in Fig. 1(b). In this way, the domain-invariant representation learning could be gradually promoted, and the potential of adaptation model will be unleashed, leading a satisfied classification performance. Our study is different from existing methods that purely designed for long-tailed classification [22,69,71] in application scenarios and exhibits advantages in domain-agnostic representation learning. This problem is challenging, but valuable and meaningful for DA task.

There also exists few works have noticed the distribution imbalance issues in the domain adaptation task, and try to tackle it by re-weighting (IWAN [61,70]), data re-sampling (RADA [24]), or data augmentation (Domain Mixup [65]). Differently, our paper focuses on a more general imbalance setting, which contains two aspects of long-tailed intra-class and inter-class distribution. Besides, we try to achieve a high-powered optimization strategy to empower DA model study each sample well to promote distribution alignment without any cost increase.

To this end, we propose to replace the original immutable domain labels with an adjustable and importance-aware alternative, dubbed Go-getting Domain Labels (Go-labels). Its core idea is to adaptively reduce the importance of these dominated training data that have been aligned, and timely encourage the domain discriminator to pay more attention to those easy-to-miss minority clusters, which ensures each sample can be well studied. In the implementation, we assign a go-getting domain label (Go-label) to each sample according to its own optimization situation: If one sample has ambiguous domain predictions (*e.g.*, $\sim$0.5) when passing through domain discriminator, it means such sample has been well studied, or said, the learned feature w.r.t this sample has been domain-invariant. Then, we enforce a relaxation constraint on it through changing its groundtruth (*i.e.*, directly taking 0.5 as its new domain label), so as to reduce its optimization importance. Our contributions are summarized as follows,

- We revisit domain adaptation problem from an optimization perspective, and pinpoint the training defect caused by imbalanced data distributions issue.
- To alleviate this issue, we propose a novel concept of go-getting domain labels (Go-labels) to achieve a dynamic adaptation, which allows each sample to be well studied and reduces long-tailed influence, so as to promote domain alignment for DA without any increase in computational cost.
- As a byproduct, our work also provides a new perspective to understand the task of adaptation, and gives theoretical insights about the effectiveness of dynamic training strategy with Go-labels.

We thoroughly study the proposed Go-labels with several toy cases, and conduct experiments on multiple domain adaptation benchmarks, including Digit-Five, Office-31, Office-Home, VisDA-2017, and large-scale DomainNet, upon various baselines, to show it is effective and reasonable.

2 Related Work

Unsupervised Domain Adaptation. Recent UDA works focus on two mainstream branches, (1) moment matching and (2) adversarial training. The former

works typically align features across domains by minimizing some distribution similarity metrics, such as Maximum Mean Discrepancy (MMD) [7,36,62] and second-/higher-order statistics [28,42,54]. Adversarial domain adaptation (ADA) methods have achieved superior performance and this paper also focuses on it. The pioneering works of DANN [15] and ADDA [59] both employ a domain discriminator to compete with a feature extractor in a two-player mini-max game. CDAN [35] improves this idea by conditioning domain discriminator on the information conveyed by the category classifier. MADA [41] uses multiple domain discriminators to capture multi-modal structures for fine-grained domain alignment. Recent GVB [11] gradually reduces the domain-specific characteristics in domain-invariant representations via a bridge layer between the generator and discriminator. MCD [49], STAR [37] and Symnet [72] all build an adversarial adaptation framework by leveraging the collision of multiple object classifiers. Unfortunately, all these methods ignore the imbalanced distribution issue in DA.

Imbalanced Domain Adaptation. Several prior works have noticed the distribution imbalance issues in domain-adversarial field, and provided rigorous analysis and explanations [23,26,55,64,74]. In particular, IWAN [70] leverages the idea of re-weighting for adaptation, and RADA [24] enhances the ability of domain discriminator in DA via sample re-sampling and augmentation. Besides, the works of [30,55,64] focus on the subpopulation shift issue (partial DA), where the source and target domains have imbalanced **label** distribution. Differently, our paper focuses on the more general covariate shift setting in DA, which contains two aspects of long-tailed intra-class and inter-class distribution. Such imbalanced problems are widely existed in the existing UDA benchmarks.

Adversarial Training. Our work is also related to the researches which aim to leverage or modify the discriminator output to further augment the standard GAN training [1,3,17,18,39,52,63]. Their core idea is to distill useful information from the discriminator to further regularize generator to obtain a better generation performance. Although our work shares a similar idea of enhancing adversarial training, the main contributions and target task are different.

3 Adversarial Domain Adaptation with Go-labels

3.1 Prior Knowledge Recap and Problem Definition

To be self-contained, we first simply review the problem formulation of adversarial domain adaptation (ADA). Taking classification task as example, we denote the source domain as $\mathcal{D}_S = \{(x_i^s, y_i^s, d_i^s)\}_{i=1}^{N_s}$ with N_s labeled samples covering C classes, $y_i^s \in [0, C-1]$. d_i^s is the domain label of each source sample and it always equals to '1' during the training [15,35]. The target domain is similarly denoted as $\mathcal{D}_T = \{x_j^t, d_i^t\}_{j=1}^{N_t}$ with N_t unlabeled samples that belong to the same C classes, d_i^t denotes the domain label of each target sample and it always equals to '0' so as to construct a '0–1' pair with source samples for adversarial optimization. Most ADA algorithms tend to learn domain-invariant representations, by adversarially training the feature extractor and domain discriminator

in a minmax two-player game [11,15,21,35]. They typically use classification loss $\mathcal{L}_{cls}$ (*i.e.*, cross-entropy loss $\mathcal{L}_{ce}$) and domain adversarial loss $\mathcal{L}_{adv}$ (*i.e.*, binary cross-entropy loss $\mathcal{L}_{bce}$) for training,

$$\begin{aligned} \mathcal{L}_{cls} &= \frac{1}{N_s}\sum_{i=1}^{N_s}\mathcal{L}_{ce}(C(F(x_i^s)), y_i^s), \\ \mathcal{L}_{adv} &= \frac{1}{N_s}\sum_{i=1}^{N_s}\mathcal{L}_{bce}(D(F(x_i^s)), d_i^s = 1) + \frac{1}{N_t}\sum_{i=1}^{N_t}\mathcal{L}_{bce}(D(F(x_i^t)), d_i^t = 0), \end{aligned} \tag{1}$$

where F, C, D represents the feature extractor, the category classifier, and the domain discriminator, respectively. They are shared across domains. The total optimization objective is described as $(\min_{D}\mathcal{L}_{adv} + \min_{F,C}\mathcal{L}_{cls} - \mathcal{L}_{adv})$. Note that, a gradient reversal layer (GRL) [15] is often used to connect feature extractor F and domain discriminator D to achieve the adversarial function by multiplying the gradient from D by a certain negative constant during the back-propagation to the feature extractor F.

Problem Definition of Imbalanced Data Distributions in DA. This paper focuses on the general covariate shift setting following [51,53] in the DA field, and assumes each domain presents an "imbalanced" data distributions. Suppose a source/target domain $\{(x_i, y_i)\}_{i=1}^n$ drawn i.i.d. from an imbalanced distribution $P(x, y)$. Such imbalanceness comprises two aspects: 1). the marginal distribution $P(y)$ of classes are likely long-tailed, i.e., inter-class long-tailed. 2). the data distribution within each class is also long-tailed, i.e., intra-class long-tailed distribution. We expect to learn a well adapted model $F(\cdot;\theta)$ with adversarial DA technique equipped with a domain discriminator $D(\cdot;\omega)$, to learn domain-invariant representations.

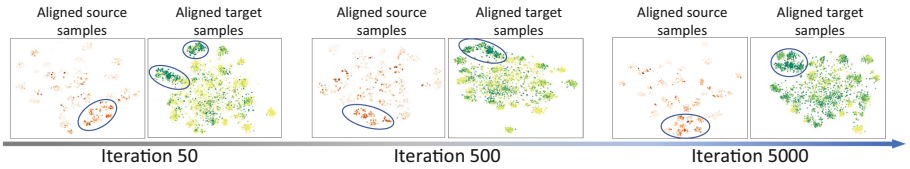


Fig. 2. Red and green points denote source and target domain data, respectively. The darker the color, the better the alignment, the more possible to be mis-classified by domain discriminator. (Color figure online)

Motivation Re-clarification. Here we look into whether the imbalanced data distribution issue actually hinders the effective ADA training, through a t-SNE [48] visualization results. This experiment is conducted on Office31 [46] (W→A setting) with the baseline of DANN [15]. We count the number of times each sample was **misclassified** by the domain discriminator during the DA training, and use this number as the color parameter. The darker the color, the

better the alignment, the more possible to be mis-classified by domain discriminator. From Fig. 2, we see that, there obviously exists an imbalance situation with training going on, where some samples (surrounded by a blue circle) have been well aligned/studied by the domain discriminator (the darker the color, the better the alignment), but some samples are still under-studied or not aligned well. Therefore, treating those aligned and not aligned training data in different ways to promise each sample being well explored to alleviate imbalance influence is urgently required.

3.2 Proposed Go-getting Domain Labels

To alleviate the optimization difficulty caused by imbalanced data distributions and thus enhance the domain-invariant representation learning, we introduce a dynamic adversarial domain adaptation framework with the proposed go-labels: when calculating the domain adversarial loss on a mini-batch that contains both source and target domain samples, we replace the original immutable domain labels of samples (source as '1', target as '0') with an adjustable domain labels (*i.e.*, Go-labels) on the fly. In formula, we modify the domain adversarial loss $\mathcal{L}_{adv}$ of Eq. 1 to

$$\mathcal{L}_{adv} = \frac{1}{N_s}\sum_{i=1}^{N_s}\mathcal{L}_{bce}(D(F(x_i^s)), g_i^s) + \frac{1}{N_t}\sum_{i=1}^{N_t}\mathcal{L}_{bce}(D(F(x_i^t)), g_i^t), \tag{2}$$

where g_i^s and g_i^t are the updated go-getting domain labels for i-th source sample and i-th target sample in the mini-batch, they are **no longer** a fixed '1' or '0', but become adjustable and adaptive. Intuitively, a reliable metric to distinguish the well-aligned large cluster data and not aligned small cluster data is needed for the new updated domain labels assignment/decision.

Measurement of Alignment. The critic, domain discriminator D, can be seen as an online scoring function for data: one sample will receive a higher score ($\sim$1) if its extracted feature is close to the source distribution, and a lower score ($\sim$0) if its extracted feature is close to the target distribution. Thus, we directly take the predicted domain results of domain discriminator, denoted as $\widetilde{d^s}/\widetilde{d^t}$, as the alignment measurement metric for each source/target sample. For example, if the domain discriminator prefers to classify a source sample ($d_i^s = 1$) as target data, *i.e.*, $\widetilde{d_i^s} \to 0$, we believe the learned feature w.r.t this sample has been well aligned and is fake enough to fool domain discriminator. In this way, we could online distinguish the well-aligned and not aligned data during training.

Go-getting Domain Labels Update. In the implementation, we merge the alignment measurement (*i.e.*, well-aligned samples selection) and domain label update into a single step. Formally, we leverage a non-parametric mathematical rounding $Round(\cdot)$ to modify the original domain labels $d_i^s = 1$, $d_i^t = 0$ of i-th source, target sample according to their predicted domain results $\widetilde{d_i^s}$, $\widetilde{d_i^t}$:

$$g_i^s = \frac{d_i^s + Round(\widetilde{d}_i^s)}{2}, \qquad g_i^t = \frac{d_i^t + Round(\widetilde{d}_i^t)}{2} \tag{3}$$

where go-getting domain labels of g_i^s, g_i^t are dynamic and adjustable, depending on the different domain prediction results $\widetilde{d}_i^s$, $\widetilde{d}_i^t$. The original domain label $d_i^s = 1$, $d_i^t = 0$ can be taken as groundtruth, and their intermediate decision boundary $(d_i^s + d_i^t)/2 = 0.5$ can be regarded as a threshold to automatically update go-getting domain labels through $Round(\cdot)$. That means that if the domain prediction result of a source sample is lower than the threshold of 0.5, *i.e.*, $\widetilde{d}_i^s < 0.5$, we believe the learned feature w.r.t this sample has been well aligned and is fake enough to fool domain discriminator.

It can be see that the $Round(\cdot)$ function could keep the raw domain labels unchanged for those correctly classified samples by D. They have not been well aligned (*i.e.*, $\widetilde{d}_i^s > 0.5$ and $\widetilde{d}_i^t < 0.5$). We only update the domain labels for these mis-classified well-aligned samples (*i.e.*, $\widetilde{d}_i^s \leq 0.5$ and $\widetilde{d}_i^t \geq 0.5$), which reduces the optimization importance of these aligned training data and encourage the domain discriminator to pay more attention to those not aligned data.

Implementation in PyTorch. A simple PyTorch-like [40] pseudo-code snippet is shown below. The dynamic adversarial DA with go-getting domain labels (Go-labels) modification amounts simply to the addition of lines 9, 10 of the example code, which indicates its ease of implementation and generality.

```
1  # Extract features from source (s) and target (t) domain
2  feat_s, feat_t = Extractor(sample_s, sample_t)
3
4  # Get true domain labels and domain predictions
5  d_s, d_t = 1, 0
6  p_s, p_t = Domain_Discriminator(feat_s, feat_t)
7
8  # Get updated go-getting domain labels
9  g_s = (d_s + torch.Round(p_s.detach()) / 2.0
10 g_t = (d_t + torch.Round(p_t.detach()) / 2.0
11
12 # Compute adversarial loss with new go-getting domain labels
13 loss_adv = torch.BCELoss(p_s, g_s) + torch.BCELoss(p_t, g_t)
```

Discussion: Why use Rounding? Rounding-based dynamic domain labels *only* reduce the importance for these well-aligned (*i.e.*, mis-classified by discriminator) majority samples progressively, while keep unchanged for those not aligned minority data. This design makes the "dynamically change" of go-getting domain labels more "targeted". If no rounding, the real-valued soft Go-labels will be *always* affected by the probability scores of domain discriminator, even the discriminator has not yet been well-trained at early stage. In short, the physical meanings behind Go-labels is to **softly reduce** the importance for these dominated majority samples on the fly while **progressively** transferring optimization focus to those minority data.

3.3 Theoretical Insights of Go-labels

Many classic domain adaptation approaches typically bound/model the adapted target error by the sum of the *(1) source error* and *(2) a notion of distance*

between the source and the target distributions. The classic generalization bound theory of the $\mathcal{H}$-divergence that based on the earlier work of [29] and used by [4,5,15] is obtained following theorem-1 in [6]:

$$\mathcal{R}_t(h) \leq \hat{\mathcal{R}}_s(h) + \frac{1}{2} d_{\mathcal{H}}(\hat{\mathcal{D}}_S^N, \hat{\mathcal{D}}_T^N) + C, \tag{4}$$

where C is a constant when such bound is achieved by hypothesis in $\mathcal{H}$. And $\hat{\mathcal{D}}_S^N$, $\hat{\mathcal{D}}_T^N$ denote the empirical distribution induced by sample of size N drawn from $\mathcal{D}_S$, $\mathcal{D}_T$ respectively. $\mathcal{R}_t$ denote the true risk on target domain, and $\hat{\mathcal{R}}_s$ denote the empirical risk on source domain.

Let $\{\mathbf{x}_i^s\}_{i=1}^N$, $\{\mathbf{x}_i^t\}_{i=1}^N$ be the samples in the empirical distributions $\hat{\mathcal{D}}_{\mathrm{S}}$ and $\hat{\mathcal{D}}_{\mathrm{T}}$ respectively. The empirical source risk can be written as $\hat{\mathcal{R}}_s(h) = \frac{1}{N} \sum_i^N \hat{\mathcal{R}}_{\mathbf{x}_i^s}(h)$. Now, considering a *dynamic updated* source-target domain distributions $\hat{\mathcal{D}}_{dS}$ and $\hat{\mathcal{D}}_{dT}$ achieved by the proposed **adjustable** go-getting domain labels, which corresponds to relabeling the well-aligned samples (assuming that the number of selected well-aligned target samples is M), the new generalization bound for this updated data distribution can be modified as

$$\mathcal{R}_t(h) \leq \big(\frac{1}{N} \sum_i^N \hat{\mathcal{R}}_{\mathbf{x}_i^s}(h) + \frac{1}{M} \sum_j^M \hat{\mathcal{R}}_{\mathbf{x}_j^t}(h)\big) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\hat{\mathcal{D}}_{dS}^{N+M}, \hat{\mathcal{D}}_{dT}^{N-M}), \tag{5}$$

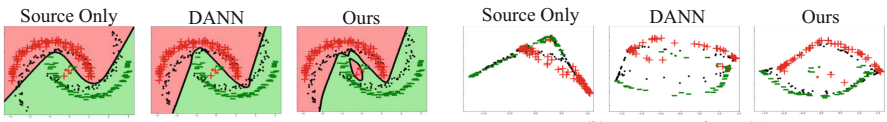
the first term on right becomes an **updated** source risk that could **re-energize** the object classifier optimization, and the second term becomes an **updated** domain discrepancy/divergency that could **re-energize** the domain discriminator optimization. They together unleash the potential of adaptation model. Besides, the risk of the target domain can be re-bounded by the risk of the updated source domain and the updated domain discrepancy, providing theoretical guarantees for the proposed approach. When $M = 0$, we get the original bound of Eq. (4). Hence, the original bound is in the feasible set of our optimization with Go-labels.

4 Experiments

4.1 Validation on Toy Problems

2D Random Point Classification. First, we observe the behavior of our method on toy problem of *2D random point classification*. We compared the class decision boundary of our method with *Baseline* obtained from the domain discriminator trained with immutable domain labels. To better evaluate adaptation performance of the trained model, we visualize source and target data separately. Experimental details are provided in **Supplementary**. We observe that the *Baseline* scheme is prone to miss the small tail cluster, especially when it is very closed to a large cluster belonged to the different class. In contrast, our method could better leverage both large/head and small/tail data clusters in the different domains to reduce discrepancy.

Inter-twinning Moons. Furthermore, we observe the behavior of Go-labels on toy problem of *inter-twinning moons* [15,49]. We compare our method with the model trained with source data only and DANN [15] in the Fig. 3. We observe that both baselines of *Source only* and *DANN* neglect the outlier samples. In contrast, our method not only gets a satisfactory classification boundary between two classes in the source domain, but also covers these minority tail data well and classifies them to the correct class. More details are presented in **Supplementary**.



(a) Label Classification (b) Representation PCA

Fig. 3. The second toy game of *inter-twinning moons*. Red "+", green "-", and black "·" markers indicate the source positive samples (label 1), source negative samples (label 0), and target samples, respectively.

4.2 Experiments on the General UDA Benchmarks

Table 1. Classification accuracy (mean ± std %) of different schemes. We evaluate the effectiveness of Go-labels with different baselines, including DANN [15], CDAN [35], GVB [11], on the Digit-Five/Office31 datasets with Cov_3FC_2 [42]/ResNet-50 [20] as backbone. We re-implement all the baselines, thus the results are sightly different from the reported ones in the original papers.

(a) Comparison results on Digit-Five.

Method	mn → sv	mn → sy	sv → mn	sv → sy	sy → mn	sy → sv	Avg.
DANN [15]	23.2 ± 0.5	40.0 ± 0.3	71.0 ± 0.3	84.6 ± 0.1	93.6 ± 0.4	84.7 ± 0.3	66.2
+Go-labels	26.3 ± 0.4	40.7 ± 0.3	79.0 ± 0.2	87.7 ± 0.7	95.3 ± 0.2	85.1 ± 0.2	**69.0**
CDAN [35]	29.8 ± 0.3	39.3 ± 0.5	69.3 ± 0.1	90.5 ± 0.0	92.5 ± 0.5	86.3 ± 0.1	67.9
+Go-labels	28.1 ± 0.5	41.3 ± 0.3	78.6 ± 0.0	90.6 ± 0.0	95.5 ± 0.5	86.4 ± 0.1	**70.1**
GVB [11]	30.0 ± 0.1	40.4 ± 0.2	72.5 ± 0.2	90.8 ± 0.5	91.9 ± 0.3	86.6 ± 0.3	68.7
+Go-labels	30.3 ± 0.1	42.1 ± 0.2	79.6 ± 0.1	90.9 ± 0.5	95.9 ± 0.3	87.2 ± 0.0	**71.0**

(b) Comparison results on Office31.

Method	A → D	A → W	D → W	W → D	D → A	W → A	Avg.
DANN [15]	82.9 ± 0.5	88.7 ± 0.3	98.5 ± 0.3	100 ± 0.0	64.9 ± 0.4	62.8 ± 0.3	82.9
+Go-labels	89.9 ± 0.4	92.4 ± 0.3	98.9 ± 0.2	100 ± 0.0	71.6 ± 0.0	68.3 ± 0.2	**86.9**
CDAN [35]	92.2 ± 0.3	93.1 ± 0.5	98.7 ± 0.1	100 ± 0.0	72.8 ± 0.5	70.1 ± 0.0	87.8
+Go-labels	93.2 ± 0.5	93.3 ± 0.3	98.6 ± 0.0	100 ± 0.0	73.8 ± 0.5	74.2 ± 0.3	**88.9**
GVB [11]	94.8 ± 0.1	92.2 ± 0.3	94.5 ± 0.3	100 ± 0.0	75.3 ± 0.2	73.2 ± 0.3	88.3
+Go-labels	95.0 ± 0.3	93.7 ± 0.2	98.5 ± 0.2	100 ± 0.0	74.9 ± 0.4	74.3 ± 0.5	**89.4**

Datasets. Except for toy tasks, we also conduct experiments on the commonly-used domain adaptation (DA) datasets, including Digit-Five [14], Office31 [46], Office-Home [60], VisDA-2017 [43], and DomainNet [42]. These datasets cover various kinds of domain gaps, such as handwritten digit style discrepancy, office supplies imaging discrepancy, and synthetic↔real-world environment discrepancy. The data distribution imbalanced issue is also widely existed, and especially serious for the large-scale set, like DomainNet. The detailed introductions for each dataset can be found in **Supplementary**.

Implementation Details. As a plug-and-play optimization strategy, we apply our Go-labels on top of four representative ADA baselines, DANN [15], CDAN [35], GVB [11], and ASAN [45] for validation. DANN has been described in Sect. 3, and CDAN additionally conditions the domain discriminator on the information conveyed by the category classifier predictions (class likelihood). Recently-proposed GVB equips the adversarial adaptation framework with a gradually vanishing bridge, which reduces the transfer difficulty by reducing the domain-specific characteristics in representations. ASAN [45] integrates relevance spectral alignment and spectral normalization into CDAN. All reported results are obtained from the average of multiple runs (**Supplementary**).

Effectiveness of Go-getting Domain Labels. Our proposed Go-labels is generic and can be applied into most existing ADA frameworks, to alleviate the optimization difficulty caused by imbalanced domain data distributions, and thus enhance the domain-invariant representation learning. To prove that, we adopt three baselines, DANN [15], CDAN [35], GVB [11], and evaluate adaptation performance on Digit-Five and Office31, respectively. Table 1(a)(b) shows the comparison results, we observe that, regardless of the difference in framework design, our Go-labels (all *+Go-labels* schemes) consistently improves the accuracy of all three baselines on two datasets, *i.e.*, 2.8%/4.0%, 2.2%/1.1%, 2.3%/1.1% gains on average for DANN, CDAN, GVB, respectively on Digit-Five/Office31. With the help of Go-labels, each sample can be well explored in a dynamic way, resulting in better adaptation performance.

What Happens to Domain Discriminator When Training with Go-labels? For this experiment, we made statistics on the mis-classified cases of the domain discriminator during the training, and then visualize the *changing trend* in Fig. 4. There are two symmetrical mis-classified cases that need to be counted: mis-classify the raw source sample into the target domain or mis-classify the raw target sample into the source domain. Experiments are conducted on the Office31 and VisDA-2017 datasets, the compared baseline scheme is DANN [15]. As shown in Fig. 4, we observe that, the number of mis-classified cases by domain discriminator in our method is more than that in the baseline. We know that, 'mis-classified by domain discriminator' can be approximately equivalent to 'well-aligned'. Therefore, more 'mis-classified' samples by domain

discriminator indicates that our method with Go-labels has a capability to align more samples, or said, could better cover those easy-to-miss minority clusters for alignment.

Loss Curve Comparison. Here we also show and compare the loss curves of domain discriminator for baseline DANN and our method. From Fig. 5, we can observe that the loss curve of baseline first drops quickly and gradually rises to near a constant as training progresses. In comparison, the domain discriminator loss curve of our method drops slowly, because more samples (including large and small cluster data) need to be studied/aligned during the training, which could in turn further drive better domain-invariant representations learning.

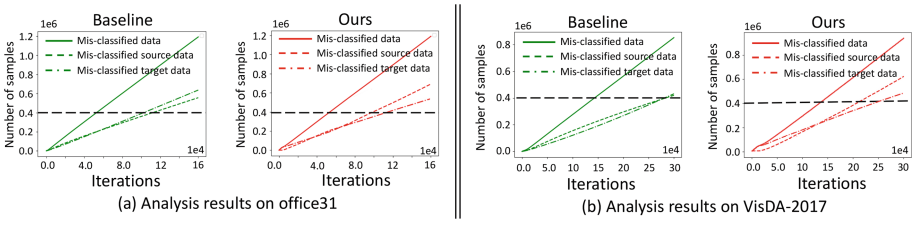


Fig. 4. Trend analysis of the mis-classified cases statistics for the domain discriminator in the training. Here, baseline is DANN [15] with ResNet-50 as backbone.

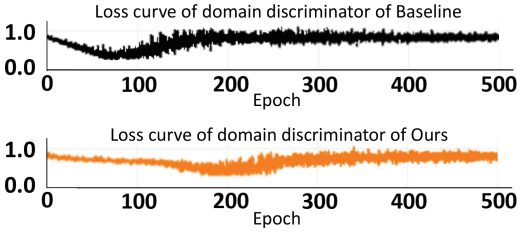


Fig. 5. Domain discriminator loss curves of baseline (DANN) and our method (DANN + Go-labels). Experiments are on the setting of W→A of Office31.

Why Not Directly Ignore Well-Aligned Data? The core idea of our dynamic adversarial domain adaptation with Go-labels is to transfer the model attention from over-studied aligned data to those overlooked samples progressively, so as to allow each sample to be well studied. Therefore, an intuitive alternative solution is to directly discard these over-aligned data, *e.g.*, simply zero out their gradients. We conduct this experiment on the Office31 based on DANN [15]. In Table 2, we see the scheme of *DANN + Zero Out* that directly discards these well-aligned samples is even inferior to *Baseline (DANN)* by 2.1% on average. This indicates that such **'hard and rude'** data filtering trick is

sub-optimal because it may lose some important knowledge by mistake. Differently, our Go-labels training strategy could **softly** and **progressively** transfer the focus of optimization from the over-aligned samples to the under-explored data.

Comparison with Re-weighting Based Methods. As pointed in previous researches [27,33], the re-weighting schemes have the risks of over-fitting the tail data (by over-sampling) and also have the risk of under-fitting the global data distribution (by under-sampling), when data imbalance is extreme [76]. Besides, most sample re-weighting techniques [67] start re-weighting operation from the beginning of the entire training process. However, the non-converged feature extractor may affect the re-weighting decision, and cause unstable training. To prove that, we further compare our Go-labels with some sample (re)weighting based methods, including entropy-based re-weighting (+ E) [35], IWAN [70]. Entropy-based re-weighting (+ E) aims to prioritize the easy-to-transfer samples according to predictions of the category classifier to ease the entire adaptation optimization. IWAN [70] re-weights the source samples to exclude the outlier classes in the source domain. Table 2 shows the comparison results. We can observe that even all the sample re-weighting strategies bring performance gains, 3.0% for *+ E* and 2.4% for *+ IWAN*, but our Go-labels strategy still outperforms all competitors. In addition, our Go-labels is also complementary to these re-weighting techniques, the scheme of *DANN + E + Go-labels* still could achieve 1.6% gain in comparison with *DANN + E*.

Table 2. Comparison with gradient penalization and re-weighting related methods on Office31. The adopted baseline is DANN.

Method	A→D	A→W	D→W	W→D	D→A	W→A	Avg.
DANN [15]	82.9 ± 0.5	88.7 ± 0.3	98.5 ± 0.3	100 ± 0.0	64.9 ± 0.4	62.8 ± 0.3	82.9
+ Zero Out	84.3 ± 0.2	82.9 ± 0.1	98.2 ± 0.3	100 ± 0.0	58.0 ± 0.4	61.1 ± 0.5	80.8
+ E [35]	86.3 ± 0.1	91.0 ± 0.2	98.8 ± 0.3	100 ± 0.0	69.6 ± 0.3	69.8 ± 0.5	85.9
+ IWAN [70]	85.9 ± 0.1	91.9 ± 0.1	98.3 ± 0.2	100 ± 0.0	68.3 ± 0.4	67.5 ± 0.5	85.3
+ Go-labels	89.9 ± 0.4	92.4 ± 0.3	98.9 ± 0.2	100 ± 0.0	71.6 ± 0.0	68.3 ± 0.2	86.9
+ E + Go-labels	91.2 ± 0.2	91.4 ± 0.4	99.1 ± 0.1	100 ± 0.0	71.4 ± 0.1	71.9 ± 0.3	87.5

Go-labels is Well-Suited to DA Settings with Intra-class and Inter-class Imbalance. The results on DomainNet [42] can be taken as experimental evidence to prove this point. Because DomainNet has multiple domains, when testing the model adaptation ability on the certain target domain, the rest domains are mixuped as a large source domain. Such large source domain is seriously imbalanced, with both of intra-class and inter-class situations [55]. From the Table 3, we can observe that our Go-labels consistently achieves gains on the different sub-settings, which demonstrates it is always effective to DA settings with the different imbalances to some extents. We analyze that the go-getting labeling encourages the domain discriminator to learn well each sample to get a

better source and target domain alignment. This in turn drives a better feature extractor to learn discriminative and domain-invariant features for all samples (they promote each other). Thus, a better feature extractor further improves the classifier and classification accuracy even the classes are still imbalanced.

Analysis About Rounding Operation in Go-labels. To validate the rounding design in Go-labels, we experimented with *real-valued soft* go-getting domain labels (based on the probability scores without rounding) for comparison. Actually, this is the initial version of our Go-labels. This scheme of using real-valued soft go-getting domain labels (built upon DANN) is inferior to our rounding version by 9.4% in average accuracy on Office31 (77.5% vs. 86.9%, baseline of DANN is 82.9%). We analyze such large drop is because that the real-valued soft go-getting domain labels of training samples are **always** affected by the probability scores of domain discriminator, even the discriminator has not yet been well-trained at early stage. On the contrary, our rounding-based Go-labels makes no influence for the entire optimization at the stage where the domain discriminator could clearly/correctly classify source-target sample. And, it **only** reduce the importance for these well-aligned (mis-classified by discriminator) majority samples progressively while keep unchanged for those not aligned minority data. In short, the rounding design makes Go-labels more robust.

Table 3. Classification accuracy on DomainNet. ResNet-101 as backbone.

Methods	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Average
MDAN [75]	60.3 ± 0.41	25.0 ± 0.43	50.3 ± 0.36	8.2 ± 1.92	61.5 ± 0.46	51.3 ± 0.58	42.8
M3SDA [42]	58.6 ± 0.53	26.0 ± 0.89	52.3 ± 0.55	6.3 ± 0.58	62.7 ± 0.51	49.5 ± 0.76	42.7
CMSS [67]	64.2 ± 0.18	**28.0 ± 0.20**	53.6 ± 0.39	16.0 ± 0.12	63.4 ± 0.21	**53.8 ± 0.35**	46.5
CDAN [35]	63.3 ± 0.21	23.2 ± 0.11	54.0 ± 0.34	16.8 ± 0.41	62.8 ± 0.14	50.9 ± 0.43	45.2
+ Go-labels	**65.7 ± 0.22**	25.7 ± 0.34	**55.6 ± 0.21**	**18.4 ± 0.31**	**63.6 ± 0.28**	53.6 ± 0.13	**47.1**

Table 4. Performance (%) comparisons with the state-of-the-art UDA approaches on Office31. All experiments are based on ResNet-50 pre-trained on ImageNet.

Method	Venue	A→D	A→W	D→W	W→D	D→A	W→A	Avg.
DANCE [47]	NeurIPS'20	89.4 ± 0.1	88.6 ± 0.2	97.5 ± 0.4	**100.0 ± .0**	69.5 ± 0.5	68.2 ± 0.2	85.5
Re-weight [61]	IJCAI'20	91.7 ± —-	95.2 ± —-	98.6 ± —-	**100.0** ± –	74.5 ± —-	73.7 ± —-	89.0
DADA [57]	AAAI'20	92.3 ± 0.1	93.9 ± 0.2	99.2 ± 0.1	**100.0 ± .0**	74.4 ± 0.1	74.2 ± 0.1	89.0
MetaAlign [62]	CVPR'21	93.0 ± 0.5	94.5 ± 0.3	98.6 ± 0.0	**100.0 ± .0**	75.0 ± 0.3	73.6 ± 0.0	89.2
FGDA [16]	ICCV'21	93.3 ± —-	93.2 ± —-	99.1 ± —-	**100.0 ± .0**	73.2 ± —-	72.7 ± —-	88.6
SCDA [32]	ICCV'21	94.2 ± —-	95.2 ± —-	98.7 ± —-	99.8 ± —-	75.7 ± —-	76.2 ± —-	90.0
RADA [24]	ICCV'21	96.1 ± 0.4	96.2 ± 0.4	**99.3 ± 0.1**	**100.0 ± .0**	**77.5 ± 0.1**	**77.4 ± 0.3**	91.1
CDAN + E (Baseline) [35]	NIPS'18	90.8 ± 0.3	94.0 ± 0.5	98.1 ± 0.3	**100.0 ± .0**	72.4 ± 0.4	72.1 ± 0.3	87.9
CDAN + E + Go-labels	This work	94.2 ± 0.3	93.3 ± 0.1	99.0 ± 0.1	**100.0 ± .0**	75.8 ± 0.1	75.2 ± 0.3	89.6
GVB (Baseline) [11]	CVPR'20	94.8 ± 0.1	92.2 ± 0.2	94.5 ± 0.2	**100.0 ± .0**	75.3 ± 0.3	73.2 ± 0.4	88.3
GVB + Go-labels	This work	95.0 ± 0.3	93.7 ± 0.1	98.5 ± 0.1	**100.0 ± .0**	74.9 ± 0.1	74.3 ± 0.3	89.4
ASAN (Baseline) [45]	ACCV'20	95.6 ± 0.4	**98.8 ± 0.2**	94.4 ± 0.9	**100.0 ± .0**	74.7 ± 0.3	74.0 ± 0.9	90.0
ASAN + Go-labels	This work	**96.9** ± 0.2	**98.8 ± 0.1**	99.1 ± 0.2	**100.0 ± .0**	77.0 ± 0.3	75.9 ± 0.6	**91.3**

4.3 Comparison with State-of-the-Arts

As a general technique, we insert our Go-labels into multiple DA algorithms to validate: CDAN with entropy regularization [35] (CDAN+E), GVB [11], ASAN [45], and RADA [24]. Table 4, Table 5 and Table 6 show the comparisons with the state-of-the-art approaches on Office31, Office-Home and VisDA-2017, respectively. For fair comparison, we report the results from their original papers if available, and we also report the results of the baseline schemes *GVB* and *CDAN+E* reproduced by our implementation. We find *GVB+Go-labels*, *CDAN+E+Go-labels*, and *ASAN+Go-labels* all outperform their corresponding baselines and also achieves the state-of-the-art performance on three datasets, and Go-labels is also more simple and efficient per without extra computation.

Table 5. Performance (%) comparisons with the state-of-the-art UDA approaches on Office-Home. All experiments are based on ResNet-50 pre-trained on ImageNet.

Method	Venue	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
BNM [10]	CVPR'20	52.3	73.9	80.0	63.3	72.9	**74.9**	61.7	49.5	79.7	70.5	53.6	82.2	67.9
DANCE [47]	NIPS'20	54.3	75.9	78.4	64.8	72.1	73.4	63.2	53.0	79.4	73.0	58.2	82.9	69.1
Reweight [61]	IJCAI'20	55.5	73.5	78.7	60.7	74.1	73.1	59.5	55.0	80.4	72.4	60.3	84.3	68.9
SRDC [56]	CVPR'20	52.3	76.3	81.0	**69.5**	76.2	78.0	**68.7**	53.8	81.7	**76.3**	57.1	85.0	71.3
CKB [38]	CVPR'21	54.2	74.1	77.5	64.6	72.2	71.0	64.5	53.4	78.7	72.6	58.4	82.8	68.7
TSA [31]	CVPR'21	57.6	75.8	80.7	64.3	76.3	75.1	66.7	55.7	81.2	75.7	61.9	83.8	71.2
MetaAg. [62]	CVPR'21	**59.3**	76.0	80.2	65.7	74.7	75.1	65.7	56.5	81.6	74.1	61.1	**85.2**	71.3
FGDA [16]	ICCV'21	52.3	**77.0**	78.2	64.6	75.5	73.7	64.0	49.5	80.7	70.1	52.3	81.6	68.3
SCDA [32]	ICCV'21	57.1	75.9	79.9	66.2	**76.7**	75.2	65.3	55.6	81.9	74.7	**62.6**	84.5	71.3
CDAN+E [35]	NIPS'18	55.6	72.5	77.9	62.1	71.2	73.4	61.2	52.6	80.6	73.1	55.5	81.4	68.1
+Go-labels	This work	56.0	74.4	78.2	63.9	72.7	72.0	63.7	54.1	81.7	73.3	59.6	83.0	69.4
ASAN [45]	ACCV'20	53.6	73.0	77.0	62.1	73.9	72.6	61.6	52.8	79.8	73.3	60.2	83.6	68.6
+Go-labels	This work	55.5	75.1	79.3	65.0	74.1	74.3	64.8	54.4	81.8	74.7	61.8	85.2	70.5
GVB [11]	CVPR'20	57.0	74.7	79.8	64.6	74.1	74.6	65.2	55.1	81.0	74.6	59.7	84.3	70.4
+Go-labels	This work	57.9	76.2	**81.1**	65.9	75.0	73.7	67.0	**56.6**	**82.9**	75.2	61.0	84.6	**71.4**

Table 6. Performance (%) comparisons with the state-of-the-art UDA approaches on VisDA-2017. All experiments are based on ResNet-50 pre-trained on ImageNet.

Method	Venue	Avg.
MDD [73]	ICML'19	74.61
SAFN [66]	ICCV'19	76.10
DANCE [47]	NeurIPS'20	70.20
CDAN + E (Baseline) [35]	NIPS'18	70.83
CDAN + E + Go-labels	This work	75.12
ASAN (Baseline) [45]	ACCV'20	72.34
ASAN + Go-labels	This work	75.21
GVB (Baseline) [11]	CVPR'20	75.34
GVB + Go-labels	This work	76.42
RADA (Baseline) [24]	ICCV'21	76.30
RADA + Go-labels	This work	**77.52**

5 Conclusion

We propose a simple plug-and-play technique dubbed go-getting domain labels (Go-labels) to achieve a dynamic adversarial domain adaptation framework, which effectively alleviates the imbalanced data distribution issue and significantly enhances the domain-invariant representation learning. Go-labels requires changing only two lines of code that yields non-trivial improvements across a wide variety of adversarial based UDA architectures. In fact, improvements of Go-labels come without bells and whistles on all domain adaptation benchmarks we evaluated, despite embarrassingly simple.

Acknowledgments. This work was supported in part by NSFC under Grant U1908209, 62021001 and the National Key Research and Development Program of China 2018AAA0101400. This work was also supported in part by the Advanced Research and Technology Innovation Centre (ARTIC), the National University of Singapore under Grant (project number: A-0005947-21-00).

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: ICML, pp. 214–223. PMLR (2017)
2. Arpit, D., et al.: A closer look at memorization in deep networks. In: ICML, pp. 233–242. PMLR (2017)
3. Azadi, S., Olsson, C., Darrell, T., Goodfellow, I., Odena, A.: Discriminator rejection sampling. arXiv preprint arXiv:1810.06758 (2018)
4. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Mach. Learn. **79**(1–2), 151–175 (2010)
5. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: NeurIPS, pp. 137–144 (2007)
6. Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Wortman, J.: Learning bounds for domain adaptation. In: NeurIPS, pp. 129–136 (2008)
7. Borgwardt, K.M., Gretton, A., Rasch, M.J., Kriegel, H.P., Schölkopf, B., Smola, A.J.: Integrating structured biological data by kernel maximum mean discrepancy. Bioinformatics **22**(14), e49–e57 (2006)
8. Chen, L., et al.: Reusing the task-specific classifier as a discriminator: discriminator-free adversarial domain adaptation. In: CVPR, pp. 7181–7190 (2022)
9. Chen, Q., Liu, Y., Wang, Z., Wassell, I., Chetty, K.: Re-weighted adversarial adaptation network for unsupervised domain adaptation. In: CVPR, pp. 7976–7985 (2018)
10. Cui, S., Wang, S., Zhuo, J., Li, L., Huang, Q., Tian, Q.: Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In: CVPR, pp. 3941–3950 (2020)
11. Cui, S., Wang, S., Zhuo, J., Su, C., Huang, Q., Tian, Q.: Gradually vanishing bridge for adversarial domain adaptation. In: CVPR, pp. 12455–12464 (2020)
12. Feldman, V.: Does learning require memorization? a short tale about a long tail. In: Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, pp. 954–959 (2020)
13. Feldman, V., Zhang, C.: What neural networks memorize and why: discovering the long tail via influence estimation. arXiv preprint arXiv:2008.03703 (2020)

14. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: ICML, pp. 1180–1189. PMLR (2015)
15. Ganin, Y., et al.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**(1), 2030–2096 (2016)
16. Gao, Z., Zhang, S., Huang, K., Wang, Q., Zhong, C.: Gradient distribution alignment certificates better adversarial domain adaptation. In: ICCV, pp. 8937–8946 (2021)
17. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. arXiv preprint arXiv:1704.00028 (2017)
18. Guo, T., et al.: On positive-unlabeled classification in gan. In: CVPR, pp. 8385–8393 (2020)
19. Haeusser, P., Frerix, T., Mordvintsev, A., Cremers, D.: Associative domain adaptation. In: ICCV, pp. 2765–2773 (2017)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
21. Hoffman, J., et al.: Cycada: cycle-consistent adversarial domain adaptation. In: ICML, pp. 1989–1998. PMLR (2018)
22. Huang, C., Li, Y., Loy, C.C., Tang, X.: Learning deep representation for imbalanced classification. In: CVPR, pp. 5375–5384 (2016)
23. Jiang, X., Lao, Q., Matwin, S., Havaei, M.: Implicit class-conditioned domain alignment for unsupervised domain adaptation. In: ICML, pp. 4816–4827. PMLR (2020)
24. Jin, X., Lan, C., Zeng, W., Chen, Z.: Re-energizing domain discriminator with sample relabeling for adversarial domain adaptation. In: ICCV (2021)
25. Jin, X., Lan, C., Zeng, W., Chen, Z.: Style normalization and restitution for domain generalization and adaptation. IEEE Trans. Multimedia **24**, 3636–3651 (2021)
26. Johansson, F.D., Sontag, D., Ranganath, R.: Support and invertibility in domain-invariant representations. In: AISTATS, pp. 527–536. PMLR (2019)
27. Kang, B., et al.: Decoupling representation and classifier for long-tailed recognition. In: ICLR (2019)
28. Kang, G., Jiang, L., Yang, Y., Hauptmann, A.G.: Contrastive adaptation network for unsupervised domain adaptation. In: CVPR, pp. 4893–4902 (2019)
29. Kifer, D., Ben-David, S., Gehrke, J.: Detecting change in data streams. In: VLDB, vol. 4, pp. 180–191. Toronto, Canada (2004)
30. Li, B., et al.: Rethinking distributional matching based domain adaptation. arXiv preprint arXiv:2006.13352 (2020)
31. Li, S., Xie, M., Gong, K., Liu, C.H., Wang, Y., Li, W.: Transferable semantic augmentation for domain adaptation. In: CVPR, pp. 11516–11525 (2021)
32. Li, S., et al.: Semantic concentration for domain adaptation. In: ICCV, pp. 9102–9111 (2021)
33. Li, Y., et al.: Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In: CVPR, pp. 10991–11000 (2020)
34. Liu, J., Sun, Y., Han, C., Dou, Z., Li, W.: Deep representation learning on long-tailed data: a learnable embedding augmentation perspective. In: CVPR, pp. 2970–2979 (2020)
35. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: NeurIPS, pp. 1640–1650 (2018)
36. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: ICML, pp. 2208–2217 (2017)
37. Lu, Z., Yang, Y., Zhu, X., Liu, C., Song, Y.Z., Xiang, T.: Stochastic classifiers for unsupervised domain adaptation. In: CVPR, pp. 9111–9120 (2020)

38. Luo, Y.W., Ren, C.X.: Conditional bures metric for domain adaptation. In: CVPR, pp. 13989–13998 (2021)
39. Nowozin, S., Cseke, B., Tomioka, R.: f-gan: training generative neural samplers using variational divergence minimization. In: NeurIPS (2016)
40. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703 (2019)
41. Pei, Z., Cao, Z., Long, M., Wang, J.: Multi-adversarial domain adaptation. In: AAAI, vol. 32 (2018)
42. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: ICCV, pp. 1406–1415 (2019)
43. Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., Saenko, K.: Visda: the visual domain adaptation challenge. arXiv preprint arXiv:1710.06924 (2017)
44. Peng, Z., Huang, W., Guo, Z., Zhang, X., Jiao, J., Ye, Q.: Long-tailed distribution adaptation. In: ACMMM, pp. 3275–3282 (2021)
45. Raab, C., Vath, P., Meier, P., Schleif, F.M.: Bridging adversarial and statistical domain transfer via spectral adaptation networks. In: ACCV (2020)
46. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_16
47. Saito, K., Kim, D., Sclaroff, S., Saenko, K.: Universal domain adaptation through self supervision. In: NeurIPS (2020)
48. Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Strong-weak distribution alignment for adaptive object detection. In: CVPR, pp. 6956–6965 (2019)
49. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: CVPR, pp. 3723–3732 (2018)
50. Sankaranarayanan, S., Balaji, Y., Castillo, C.D., Chellappa, R.: Generate to adapt: aligning domains using generative adversarial networks. In: CVPR, pp. 8503–8512 (2018)
51. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. J. Stat. Plan. Inference **90**(2), 227–244 (2000)
52. Sinha, S., Zhao, Z., Alias Parth Goyal, A.G., Raffel, C.A., Odena, A.: Top-k training of gans: improving gan performance by throwing away bad samples. In: NeurIPS, vol. 33, pp. 14638–14649 (2020)
53. Sugiyama, M., Krauledat, M., Müller, K.R.: Covariate shift adaptation by importance weighted cross validation. J. Mach. Learn. Res. **8**(5), 1–21 (2007)
54. Sun, B., Feng, J., Saenko, K.: Return of frustratingly easy domain adaptation. In: AAAI, vol. 30 (2016)
55. Tan, S., Peng, X., Saenko, K.: Class-imbalanced domain adaptation: an empirical odyssey. In: Bartoli, A., Fusiello, A. (eds.) ECCV 2020. LNCS, vol. 12535, pp. 585–602. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66415-2_38
56. Tang, H., Chen, K., Jia, K.: Unsupervised domain adaptation via structurally regularized deep clustering. In: CVPR, pp. 8725–8735 (2020)
57. Tang, H., Jia, K.: Discriminative adversarial domain adaptation. In: AAAI, vol. 34, pp. 5940–5947 (2020)
58. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR 2011. pp. 1521–1528
59. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR, pp. 7167–7176 (2017)
60. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: CVPR (2017)

61. Wang, S., Zhang, L.: Self-adaptive re-weighted adversarial domain adaptation. IJCAI (2020)
62. Wei, G., Lan, C., Zeng, W., Chen, Z.: Metaalign: coordinating domain alignment and classification for unsupervised domain adaptation. In: CVPR (2021)
63. Wu, Y., Donahue, J., Balduzzi, D., Simonyan, K., Lillicrap, T.: Logan: latent optimisation for generative adversarial networks. arXiv preprint arXiv:1912.00953 (2019)
64. Wu, Y., Winston, E., Kaushik, D., Lipton, Z.: Domain adaptation with asymmetrically-relaxed distribution alignment. In: ICML, pp. 6872–6881. PMLR (2019)
65. Xu, M., et al.: Adversarial domain adaptation with domain mixup. In: AAAI (2020)
66. Xu, R., Li, G., Yang, J., Lin, L.: Larger norm more transferable: an adaptive feature norm approach for unsupervised domain adaptation. In: ICCV, pp. 1426–1435 (2019)
67. Yang, L., Balaji, Y., Lim, S.-N., Shrivastava, A.: Curriculum manager for source selection in multi-source domain adaptation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12359, pp. 608–624. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58568-6_36
68. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: NeurIPS (2014)
69. You, C., Li, C., Robinson, D.P., Vidal, R.: Scalable exemplar-based subspace clustering on class-imbalanced data. In: ECCV, pp. 67–83 (2018)
70. Zhang, J., Ding, Z., Li, W., Ogunbona, P.: Importance weighted adversarial nets for partial domain adaptation. In: CVPR, pp. 8156–8164 (2018)
71. Zhang, X., Fang, Z., Wen, Y., Li, Z., Qiao, Y.: Range loss for deep face recognition with long-tailed training data. In: ICCV, pp. 5409–5418 (2017)
72. Zhang, Y., Tang, H., Jia, K., Tan, M.: Domain-symmetric networks for adversarial domain adaptation. In: CVPR, pp. 5031–5040 (2019)
73. Zhang, Y., Liu, T., Long, M., Jordan, M.I.: Bridging theory and algorithm for domain adaptation. In: ICML (2019)
74. Zhao, H., Des Combes, R.T., Zhang, K., Gordon, G.: On learning invariant representations for domain adaptation. In: ICML, pp. 7523–7532. PMLR (2019)
75. Zhao, H., Zhang, S., Wu, G., Moura, J.M., Costeira, J.P., Gordon, G.J.: Adversarial multiple source domain adaptation. In: NeurIPS, pp. 8559–8570 (2018)
76. Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: Bbn: bilateral-branch network with cumulative learning for long-tailed visual recognition. In: CVPR, pp. 9719–9728 (2020)