UNLEASH THE POTENTIAL OF ADAPTATION MODELS VIA DYNAMIC DOMAIN LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we propose an embarrassing simple yet highly effective adversarial domain adaptation (ADA) method for effectively training models for alignment. We view ADA problem primarily from a neural network memorization perspective and point out a fundamental dilemma, in that the real-world data often exhibits an imbalanced distribution where the majority data clusters typically dominate and biase 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 dynamic domain labels (DDLs) to replace the original immutable domain labels on the fly. DDLs adaptively and timely transfer the model attention from over-memorized aligned data to those easily overlooked samples, which allows each sample can be well studied and fully unleashes the potential of adaption model. Albeit simple, this dynamic adversarial domain adaptation (DADA) framework with DDLs effectively promotes adaptation. We demonstrate through empirical results on real and synthetic 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 (Torralba & Efros; Yosinski et al., 2014). 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 (Ganin & Lempitsky, 2015; Ganin et al., 2016; Haeusser et al., 2017; Kang et al., 2019b; Cui et al., 2020a).

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 (Ganin et al., 2016; Chen et al., 2018; Sankaranarayanan et al., 2018; Long et al., 2018; Cui et al., 2020b). 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') (Ganin et al., 2016; Long et al., 2018; Cui et al., 2020b). However, this fixed positive-negative separation criterion neglects a fact that most real-world data exhibit imbalanced distributions: the clusters with abundant examples (*i.e.*, majority clusters) may **swamp** the clusters with few examples (*i.e.*, minority clusters). Such imbalanceness contains two aspects, intra-class long-tailed distribution and inter-class long-tailed distribution (Tan et al., 2020; Wu et al., 2019b), and is widely existed in many UDA benchmarks. For example, in DomainNet, 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 have no background scene (empty) while minority "bike" samples have real-world background instead.

Deep neural networks (DNNs), on the other hand, typically learn simple patterns first before memorizing. In other words, DNN optimization is content-aware, taking advantage of patterns shared by multiple training examples (Arpit et al., 2017). The majority domain clusters would therefore dominate the memorization of domain discriminator in DA, so that bias its decision boundary and hinder the effective adaptation. As shown in Figure 1(a), only the majority 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 biases the optimization of domain discriminator so that misleads feature extractor to learn unexpected domain-specific knowledge from majority clusters. As a result, the adapted model still can not correctly classify these under-explored samples.

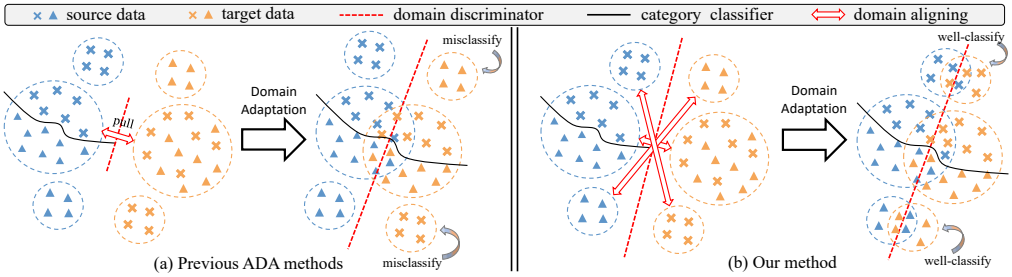


Figure 1: Motivation illustration. ($\times$, $\triangle$) denote two different classes, and (blue, orange) color mean different domains. (a) Previous DA methods tend to be dominated by those large majority clusters and neglects small minority clusters, which will biase the domain discriminator optimization,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

In this paper, we attempt to design an optimization strategy to progressively take full advantage of both majority and minority data clusters across different domains, like shown in Figure 1(b). In this way, the domain-invariant representation learning could be gradually enhanced, and the potential of adaptation model will be unleashed, leading a satisfied classification performance.

To this end, we propose to replace the original immutable domain labels with a variable and importance-aware alternative, dubbed Dynamic Domain Label (DDL). Its core idea is to adaptively reduce the importance of these dominated training data that have been aligned, and 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 dynamic domain 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 "memorized", or said, the learned feature w.r.t this sample has been domain-invariant. Then, we map this well-aligned sample to an intermediate domain label space (*i.e.*, use 0.5 as domain label), so as to reduce its optimization importance. Our contributions are summarized as follows,

- We revisit ADA problem from a deep network memorization perspective, and pinpoint the optimization defect caused by the common imbalanced data distributions.
- To alleviate this issue, we propose a novel concept of dynamic domain label (DDL) to achieve a dynamic adversarial domain adaptation (DADA), which allows each sample can be well studied to promote domain alignment without any increase in computational cost.
- As a byproduct, our work also provides a new perspective to better learn domain-invariant features in a simple dynamic manner with variable domain labels.

We thoroughly study the proposed DDL 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) (Borgwardt et al., 2006; Long et al., 2017; Wei et al., 2021) and second-/higher-order statistics (Sun et al., 2016; Peng et al., 2019; Kang et al., 2019b). Adversarial domain adaptation (ADA) methods have achieved superior performance and this paper focus on it. The pioneering works of DANN (Ganin et al., 2016) and ADDA (Tzeng et al., 2017) both employ a domain discriminator to compete with a feature extractor in a two-player mini-max game. CDAN (Long et al., 2018) improves this idea by conditioning domain discriminator on the information conveyed by the category classifier. MADA (Pei et al., 2018) uses multiple domain discriminators to capture multi-modal structures for fine-grained domain alignment. Recent GVB (Cui et al., 2020b) gradually reduces the domain-specific characteristics in domain-invariant representations via a bridge layer between the generator and discriminator. MCD (Saito et al., 2018), STAR (Lu et al., 2020) and Symnet (Zhang et al., 2019a) 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 (Johansson et al., 2019; Zhao et al., 2019; Jiang et al., 2020; Tan et al., 2020; Wu et al., 2019b). In particular, IWAN (Zhang et al., 2018) leverages the idea of re-weighting for adaptation, and RADA (Jin et al., 2021) enhances the ability of domain discriminator in DA via sample re-sampling and augmentation. Besides, the works of (Wu et al., 2019b; Li et al., 2020a; Tan et al., 2020) pay their attention on the label/subpopulation shift issue, 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.

Techniques for Training GANs. Our work is also related to the line of research which aims to leverage or modify the discriminator output to further augment the standard GAN training (Nowozin et al., 2016; Arjovsky et al., 2017; Gulrajani et al., 2017; Azadi et al., 2018; Wu et al., 2019a; Guo et al., 2020; Sinha et al., 2020). 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 Dynamic Adversarial Domain Adaptation (DADA)

Prior Knowledge Recap. 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 (Ganin et al., 2016; Long et al., 2018). 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 'positive-negative' 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 (Ganin et al., 2016; Hoffman et al., 2018; Long et al., 2018; Cui et al., 2020b). They typically use two loss functions, 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,

$$\mathcal{L}_{cls} = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathcal{L}_{ce}(C(F(x_i^s)), y_i^s), \tag{1}$$

$$\mathcal{L}_{adv} = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathcal{L}_{bce}(D(F(x_i^s)), d_i^s = 1) + \frac{1}{N_t} \sum_{i=1}^{N_t} \mathcal{L}_{bce}(D(F(x_i^t)), d_i^t = 0), \tag{2}$$

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 follows,

$$\min_{D} \mathcal{L}_{adv}, \quad \min_{F,C} \mathcal{L}_{cls} - \mathcal{L}_{adv}, \tag{3}$$

Note that, a gradient reversal layer (GRL) (Ganin et al., 2016) 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 (Shimodaira, 2000; Sugiyama et al., 2007) 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.

Motivation Re-clarification. Here we look into whether the imbalanced data distribution issue actually hinders the effective ADA training, through a t-SNE (Saito et al., 2019) visualization results. This experiment is conducted on Office31 (Saenko et al., 2010) (W→A setting) with the baseline of

DANN (Ganin et al., 2016). 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 Figure 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/memorized by the domain discriminator (the darker the color, the better the alignment/memorization), 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 is urgently required.

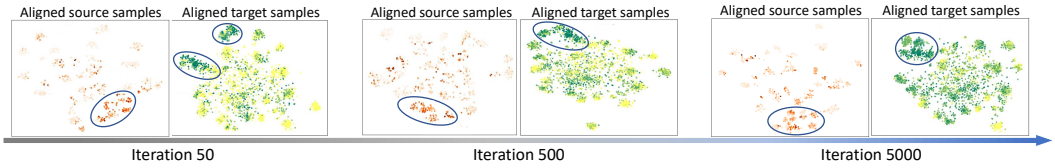


Figure 2: Red and green points denote source and target domain data, respectively. The darker the color, the better the alignment, the more possible to be mis-classified by domain discriminator.

Proposed Dynamic Domain Labels. To alleviate the optimization difficulty caused by imbalanced data distributions and thus enhance the domain-invariant representation learning, we propose a dynamic adversarial domain adaptation (DADA) framework: 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 a variable dynamic domain labels (DDLs) on the fly. In formula, we modify the domain adversarial loss $\mathcal{L}_{adv}$ of Eq. 2 to

$$\mathcal{L}_{adv} = \frac{1}{N_s}\sum_{i=1}^{N_s}\mathcal{L}_{bce}(D(F(x_i^s)), \widehat{d}_i^s) + \frac{1}{N_t}\sum_{i=1}^{N_t}\mathcal{L}_{bce}(D(F(x_i^t)), \widehat{d}_i^t), \tag{4}$$

where $\widehat{d}_i^s$ and $\widehat{d}_i^t$ are the updated 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 variable and data-adaptive. Intuitively, a reliable metric to distinguish the well-aligned majority cluster data and not aligned minority 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 in training.

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

$$\widehat{d}_i^s = (d_i^s + Round(\widetilde{d}_i^s))/2 = \begin{cases} 1, & \widetilde{d}_i^s > 0.5, \\ 0.5, & \widetilde{d}_i^s \leq 0.5. \end{cases} \tag{5}$$

$$\widehat{d}_i^t = (d_i^t + Round(\widetilde{d}_i^t))/2 = \begin{cases} 0.5, & \widetilde{d}_i^t \geq 0.5, \\ 0, & \widetilde{d}_i^t < 0.5. \end{cases} \tag{6}$$

where new domain labels of $\widehat{d}_i^s$, $\widehat{d}_i^t$ are dynamic and variable, depending on the different domain prediction results $\widetilde{d}_i^s$, $\widetilde{d}_i^t$. It can be see that we remain the raw domain labels unchanged for those correctly classified samples by D, because they have not been well aligned (*i.e.*, $\widetilde{d}_i^s > 0.5$ and $\widetilde{d}_i^t < 0.5$). We only assign a new median label (*i.e.*, 0.5) to 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 (Paszke et al., 2019) pseudo-code snippet is shown below. Our DADA with dynamic domain label (DDL) modification amounts simply to the addition of lines 9, 10 of the example code, which indicates its ease of implementation and generality.

```
# Extract features from source (s) or target (t) domain samples
feat_s, feat_t = Extractor(sample_s, sample_t)

# Get true domain labels and domain predictions
d_s, d_t = 1, 0
p_s, p_t = Domain_Discriminator(feat_s, feat_t)

# Get updated dynamic domain labels
d'_s = (d_s + torch.Round(p_s.detach()) / 2.0
d'_t = (d_t + torch.Round(p_t.detach()) / 2.0

# Compute adversarial loss with dynamic domain labels
loss_adv = torch.BCELoss(p_s, d'_s) + torch.BCELoss(p_t, d'_t)
```

Discussion: Why use 0.5 as threshold? and why use Rounding? Using 0.5 as a threshold is because considering that in the most cases (for the most UDA benchmarks), the size of source dataset and the target one are comparable, and 0.5 is an intermediate domain label space between [0, 1]. 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 domain labels more "targeted". If no rounding, the real-valued soft dynamic 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 DDL is to **softly reduce** the importance for these dominated majority samples on the fly while **progressively** transferring optimization focus to those minority data.

4 Experiments

4.1 Validation on Toy Problems

2D Random Point Classification. First, we observe the behavior of our DADA method on toy problem of *2D random point classification*, in which we use *numpy.random* (Oliphant, 2006) to generate the source and target samples that share the same label space. For the source samples, we generate point samples with 2 classes, labeled as '0' (marked as red) and '1' (marked as green), respectively. For each class, it contains 3 data clusters with different scales (*i.e.*, large head cluster has 10,000 samples, middle cluster has 5,000 samples, small tail cluster has 200 samples), this design aims to simulate the data imbalance situation in real-world, *i.e.*, the problem we focused. For the target samples, we totally generate 10,200 samples for each class. Each class has two clusters, one large head cluster with 10,000 samples and one small tail cluster with 200 samples. We compared the class decision boundary of our DADA 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. Other details are provided in **Appendix**.

As shown in Figure 3, 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. The trend of our classification boundary in the source domain has demonstrated this point. As a result, the adaptation performance on the target set of ours is obviously superior to that of *Baseline*.

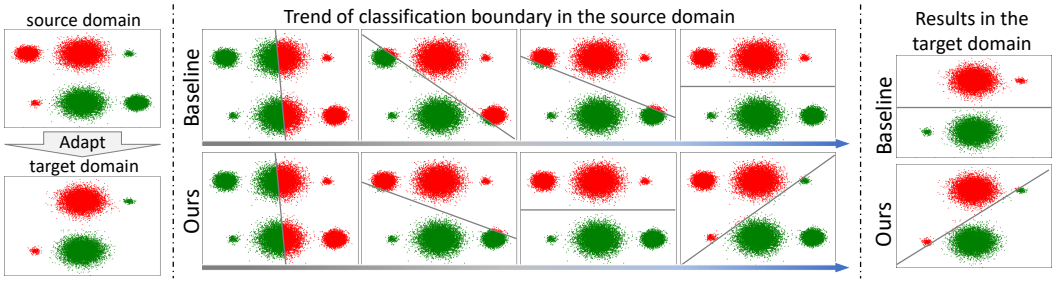
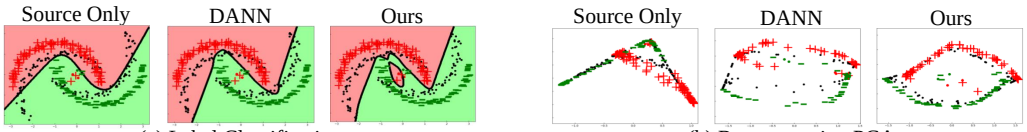


Figure 3: *2D random point classification*. Red and green points indicate the samples of class '0' and '1', respectively (left). The solid line denotes the class decision boundary, and we use "color change" to indicate its changing trend (middle). Adaptation performance on the target set is shown on right.

Inter-twinning Moons. Furthermore, we observe the behavior of our DADA with DDL on toy problem of *inter-twinning moons* (Ganin et al., 2016; Saito et al., 2018). In particular, we additionally generate some outlier samples near the center of each moon to mimic the imbalanced data distribution. For the source data, a lower moon and an upper moon are generated, and labeled as '0' and '1'. Each of them is accompanied by two extra outliers, totally 152 samples. Target data are generated by re-sampling from the source distribution. Then, we rotate each sample by 35° and remove its label to obtain an unlabeled target set. We compare our method with the model trained with source data only and DANN (Ganin et al., 2016) in the Figure 4. 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. Besides, after performing PCA, we can easily see that our method also achieves a better feature alignment in comparison with other two baselines, where target samples that denoted as black points are homogeneously spread out among source points.



(a) Label Classification (b) Representation PCA

Figure 4: 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. (a) The solid black line is the class decision boundary. (b) We also show the feature alignment situations of different schemes via a principal components analysis (PCA) transformation.

4.2 Experiments on the general UDA Benchmarks

Datasets. Except for toy tasks, we also conduct experiments on the commonly-used domain adaptation (DA) datasets, including Digit-Five (Ganin & Lempitsky, 2015), Office31 (Saenko et al., 2010), Office-Home (Venkateswara et al., 2017), VisDA-2017 (Peng et al., 2017), and DomainNet (Peng et al., 2019). 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 **Appendix**.

Implementation Details. As a plug-and-play optimization strategy, we apply our DDL on top of three representative ADA baselines, DANN (Ganin et al., 2016), CDAN (Long et al., 2018), and GVB (Cui et al., 2020b) for validation. DANN has been described in Sec. 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. All reported results are obtained from the average of multiple runs (**Appendix**).

Effectiveness of Dynamic Domain Label. Our proposed DDL 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 (Ganin et al., 2016), CDAN (Long et al., 2018), GVB (Cui et al., 2020b), 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 DDL (all +*DDL* 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 DDL, each sample can be well explored in a dynamic way, resulting in better adaptation performance.

What Happens to Domain Discriminator When Updating with DDL? 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 Figure 5. 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 (Ganin et al., 2016). As shown in Figure 5, 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

Table 1: Classification accuracy (mean ± std %) of different schemes. We evaluate the effectiveness of our DDL with different baselines, including DANN (Ganin et al., 2016), CDAN (Long et al., 2018), GVB (Cui et al., 2020b), on the Digit-Five/Office31 datasets with Cov_3FC_2 (Peng et al., 2019)/ResNet-50 (He et al., 2016) as backbone. Note that, 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 (Ganin et al., 2016)	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
+DDL	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 (Long et al., 2018)	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
+DDL	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 (Cui et al., 2020b)	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
+DDL	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 (Ganin et al., 2016)	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
+DDL	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 (Long et al., 2018)	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
+DDL	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 (Cui et al., 2020b)	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
+DDL	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**

to 'well-aligned'. Therefore, more 'mis-classified' samples by domain discriminator indicates that our method with DDL has a capability to align more samples, or said, could better cover those easy-to-miss minority clusters for alignment.

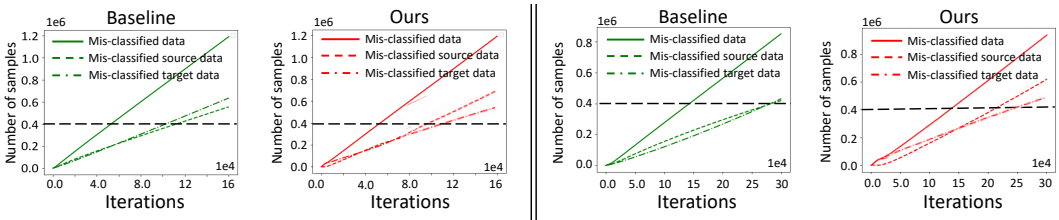


Figure 5: Trend analysis of the mis-classified cases statistics for the domain discriminator in the training. Here, baseline is DANN (Ganin et al., 2016) with ResNet-50 as backbone.

Loss Curve Comparison. Here we also show and compare the loss curves of domain discriminator for baseline DANN and our method. From Figure 6, 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 majority and minority cluster data) need to be studied/aligned during the training, which could in turn further drive better domain-invariant representations learning.

Why Not Ignore Well-Aligned Data Directly? The core idea of our dynamic adversarial domain adaptation with DDL is to transfer the model attention from over-memorized aligned data to those easily overlooked samples progressively, so as to allow each sample can 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 (Ganin et al., 2016). 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

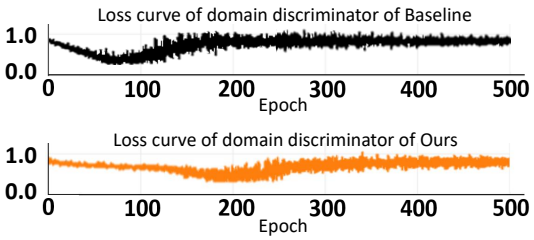


Figure 6: Domain discriminator loss curves comparison of baseline (DANN) and our method (DANN + DDL). Experiments are conducted on the setting of W→A of Office31.

such **'hard and rude'** data filtering trick is sub-optimal because it may lose some important knowledge by mistake. Differently, our DDL training strategy could **softly** and **progressively** transfer the focus of optimization from the over-aligned samples to the under-explored data.

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 (Ganin et al., 2016)	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
DANN + 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
DANN + E (Long et al., 2018)	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
DANN + IWAN (Zhang et al., 2018)	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
DANN + DDL	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
DANN + E + DDL	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

Comparison with Re-weighting based Methods. As pointed in previous researches (Kang et al., 2019a; Li et al., 2020b), 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 (Zhou et al., 2020). Besides, most sample re-weighting techniques (Yang et al., 2020) 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 DDL with some sample (re)weighting based methods, including entropy-based re-weighting (+ E) (Long et al., 2018), IWAN (Zhang et al., 2018). 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 (Zhang et al., 2018) 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 DDL strategy still outperforms all these strategies. In addition, our DDL is also complementary to these re-weighting techniques, the scheme of *DANN + E + DDL* still could achieve 1.6% gain in comparison with *DANN + E*.

DDL is Well-suited to DA Settings with Intra-class and Inter-class Imbalance. The results on DomainNet (Peng et al., 2019) 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 (Tan et al., 2020). From the Table 3, we can observe that our DDL consistently achieves gains on the different sub-settings, which demonstrates it is always effective to DA settings with the different imbalances to some extents.

Table 3: Classification accuracy (mean ± std %) on DomainNet. ResNet-101 as backbone.

Multi-Source Setting	Venue	clipart	infograph	painting	quickdraw	real	sketch	Average
MDAN (Zhao et al., 2018)	NIPS'18	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 (Peng et al., 2019)	ICCV'19	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 (Yang et al., 2020)	ECCV'20	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+E (Baseline) (Long et al., 2018)	NIPS'18	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
CDAN+E+DDL	This work	**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**

Analysis about Rounding Operation. To validate the rounding design in DDL, we experimented with *real-valued soft* dynamic domain labels (based on the probability scores without rounding) for comparison. Actually, this is the initial version of our DDL. This scheme of using real-valued soft dynamic 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 dynamic 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 DDL 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 DDL more effective and robust.

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.
MDD (Zhang et al., 2019b)	ICML'19	93.5±0.2	94.5±0.3	98.4±0.1	**100.0±.0**	74.6±0.3	72.2±0.1	88.9
TADA (Wang et al., 2019)	AAAI'19	91.6±0.3	94.3±0.3	98.7±0.1	99.8±0.2	72.9±0.2	73.0±0.3	88.4
Symnets (Zhang et al., 2019a)	CVPR'19	93.9±0.5	90.8±0.1	98.8±0.3	**100.0±.0**	74.6±0.6	72.5±0.5	88.4
SAFN (Xu et al., 2019)	ICCV'19	90.3±0.8	92.1±0.2	98.7±0.0	**100.0±.0**	73.4±0.2	71.2±0.3	87.6
DANCE (Saito et al., 2020)	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
CDAN + E (Baseline) (Long et al., 2018)	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 + DDL	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) (Cui et al., 2020b)	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 + DDL	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

4.3 Comparison with State-of-the-Arts

To compare with previous state-of-the-art domain adaptation methods, we insert our DDL optimization strategy into the recent strong domain adaptation frameworks GVB (Cui et al., 2020b) and CDAN with entropy conditioning regularization (*i.e.*, *CDAN+E* (Long et al., 2018)). Our new dynamic adversarial domain adaptation (DADA) schemes are termed as *GVB+DDL* and *CDAN+E+DDL*. 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 our baseline schemes *GVB* and *CDAN+E* reproduced by our implementation. We find *GVB+DDL* and *CDAN+E+DDL* both outperform their corresponding baselines *GVB* and *CDAN+E*, and also achieves the best performance on three datasets.

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
MDD (Zhang et al., 2019b)	ICML'19	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
Symnets (Zhang et al., 2019a)	CVPR'19	47.7	72.9	78.5	64.2	71.3	74.2	63.6	47.6	79.4	73.8	50.8	82.6	67.2
TADA (Wang et al., 2019)	AAAI'19	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
SAFN (Xu et al., 2019)	ICCV'19	54.4	73.3	77.9	65.2	71.5	73.2	63.6	52.6	78.2	72.3	58.0	82.1	68.5
BNM (Cui et al., 2020a)	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 (Saito et al., 2020)	NeurIPS'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
CDAN + E (Long et al., 2018)	NeurIPS'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
CDAN + E + DDL	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
GVB (Cui et al., 2020b)	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
GVB + DDL	This work	57.4	**76.2**	79.6	**65.9**	**74.6**	73.7	**65.9**	**55.9**	**82.9**	**75.2**	**61.0**	**84.6**	**71.1**

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 (Zhang et al., 2019b)	ICML'19	74.61
SAFN (Xu et al., 2019)	ICCV'19	76.10
DANCE (Saito et al., 2020)	NeurIPS'20	70.20
RADA (Jin et al., 2021)	ICCV'21	76.30
CDAN + E (Baseline) (Long et al., 2018)	NIPS'18	70.83
CDAN + E + DDL	This work	75.12
GVB (Baseline) (Cui et al., 2020b)	CVPR'20	75.34
GVB + DDL	This work	**76.42**

5 Conclusion

We pinpoint a optimization defect faced by existing adversarial domain adaptation (ADA) methods, which is caused by imbalanced data distribution. To address this issue, we propose a simple plug-and-play technique dubbed dynamic domain label (DDL) to achieve a dynamic adversarial domain adaptation (DADA) framework, which effectively alleviates the negative influence brought by imbalanced data distribution and significantly enhances the domain-invariant representation learning. DDL 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 DDL come without bells and whistles on all domain adaptation benchmarks we evaluated, despite embarrassingly simple.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pp. 214–223. PMLR, 2017.

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *ICML*, pp. 233–242. PMLR, 2017.

Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian Goodfellow, and Augustus Odena. Discriminator rejection sampling. *arXiv preprint arXiv:1810.06758*, 2018.

Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

Qingchao Chen, Yang Liu, Zhaowen Wang, Ian Wassell, and Kevin Chetty. Re-weighted adversarial adaptation network for unsupervised domain adaptation. In *CVPR*, pp. 7976–7985, 2018.

Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *CVPR*, pp. 3941–3950, 2020a.

Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*, pp. 12455–12464, 2020b.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pp. 1180–1189. PMLR, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

Tianyu Guo, Chang Xu, Jiajun Huang, Yunhe Wang, Boxin Shi, Chao Xu, and Dacheng Tao. On positive-unlabeled classification in gan. In *CVPR*, pp. 8385–8393, 2020.

Philip Haeusser, Thomas Frerix, Alexander Mordvintsev, and Daniel Cremers. Associative domain adaptation. In *ICCV*, pp. 2765–2773, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, pp. 1989–1998. PMLR, 2018.

Jonathan J. Hull. A database for handwritten text recognition research. *IEEE TPAMI*, 16(5).

Xiang Jiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei. Implicit class-conditioned domain alignment for unsupervised domain adaptation. In *ICML*, pp. 4816–4827. PMLR, 2020.

Xin Jin, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Re-energizing domain discriminator with sample relabeling for adversarial domain adaptation. *arXiv preprint arXiv:2103.11661*, 2021.

Fredrik D Johansson, David Sontag, and Rajesh Ranganath. Support and invertibility in domain-invariant representations. In *AISTATS*, pp. 527–536. PMLR, 2019.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2019a.

Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, pp. 4893–4902, 2019b.

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *IEEE*, 1998.

Bo Li, Yezhen Wang, Tong Che, Shanghang Zhang, Sicheng Zhao, Pengfei Xu, Wei Zhou, Yoshua Bengio, and Kurt Keutzer. Rethinking distributional matching based domain adaptation. *arXiv preprint arXiv:2006.13352*, 2020a.

Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *CVPR*, pp. 10991–11000, 2020b.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, pp. 2208–2217, 2017.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, pp. 1640–1650, 2018.

Ilya Loshchilov and Frank Hutter. Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi-Zhe Song, and Tao Xiang. Stochastic classifiers for unsupervised domain adaptation. In *CVPR*, pp. 9111–9120, 2020.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS-W*, 2011.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *NeurIPS*, 2016.

Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI*, volume 32, 2018.

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

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, pp. 1406–1415, 2019.

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

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pp. 3723–3732, 2018.

Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *CVPR*, pp. 6956–6965, 2019.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. *NeurIPS*, 2020.

Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, pp. 8503–8512, 2018.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pp. 618–626, 2017.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Samarth Sinha, Zhengli Zhao, Anirudh Goyal ALIAS PARTH GOYAL, Colin A Raffel, and Augustus Odena. Top-k training of gans: Improving gan performance by throwing away bad samples. In *NeurIPS*, volume 33, pp. 14638–14649, 2020.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, volume 30, 2016.

Shuhan Tan, Xingchao Peng, and Kate Saenko. Class-imbalanced domain adaptation: an empirical odyssey. In *ECCV*, pp. 585–602. Springer, 2020.

Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pp. 1521–1528.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pp. 7167–7176, 2017.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. Transferable attention for domain adaptation. In *AAAI*, volume 33, pp. 5345–5352, 2019.

Guoqiang Wei, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation. *CVPR*, 2021.

Yan Wu, Jeff Donahue, David Balduzzi, Karen Simonyan, and Timothy Lillicrap. Logan: Latent optimisation for generative adversarial networks. *arXiv preprint arXiv:1912.00953*, 2019a.

Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In *ICML*, pp. 6872–6881. PMLR, 2019b.

Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*, pp. 1426–1435, 2019.

Luyu Yang, Yogesh Balaji, Ser-Nam Lim, and Abhinav Shrivastava. Curriculum manager for source selection in multi-source domain adaptation. *ECCV*, 2020.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *NeurIPS*, 2014.

Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *CVPR*, pp. 8156–8164, 2018.

Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *CVPR*, pp. 5031–5040, 2019a.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I Jordan. Bridging theory and algorithm for domain adaptation. *ICML*, 2019b.

Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *NeurIPS*, pp. 8559–8570, 2018.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *ICML*, pp. 7523–7532. PMLR, 2019.

Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *CVPR*, pp. 9719–9728, 2020.

A APPENDIX

A.1 DATASETS INTRODUCTION

Here we introduce in detail the five datasets we used and their characteristics, Figure 7 shows some samples of these datasets:

- 1). *Digit-Five* consists of five different digit recognition datasets: MNIST (mn) (LeCun et al., 1998), MNIST-M (mm) (Ganin & Lempitsky, 2015), USPS (up) (Hull), SVHN (sv) (Netzer et al., 2011) and SYN (sy) (Ganin & Lempitsky, 2015). Among the five domains, there are a total of 20 DA tasks. The large differences of the two domains of SVHN and SYN with other domains make the adaptation harder. Thus, we evaluate our method on 6 transfer tasks: **mn** $\rightarrow$ **sv**, **mn** $\rightarrow$ **sy**, **sv** $\rightarrow$ **sy**, **sv** $\rightarrow$ **mn**, **sy** $\rightarrow$ **sv** and **sy** $\rightarrow$ **mn**.
- 2). *Office31* (Saenko et al., 2010) is the most widely used dataset for visual domain adaptation, with 4,652 images and 31 categories collected from three distinct domains: Amazon (A), Webcam (W) and DSLR (D). Among the three domains, there are a total of 6 DA tasks. Office31 has lots of intra-category long-tailed situations, *e.g.*, majority "bike" samples (90%) in "Amazon" domain in Office31 dataset have no background scene (empty) while minority "bike" samples have real-world background instead, which makes these minority samples look like come from "Webcam" domain.
- 3). *Office-Home* (Venkateswara et al., 2017) is a more difficult dataset (with relative large domain discrepancy) than *Office-31*. It consists of 15,500 images of 65 object classes in office and home settings. It has four dissimilar domains: Artistic images (Ar), ClipArt (Cl), Product images (Pr), and Real-World images (Rw). There are a total of 12 DA tasks.
- 4). *VisDA-2017* (Peng et al., 2017) is a simulation-to-real dataset for DA with over 280,000 images across 12 categories in the training, validation and testing domains.
- 5). *DomainNet* (Peng et al., 2019) is a recently introduced benchmark for large-scale multi-source domain adaptation (Peng et al., 2019), which includes six domains (Clipart, Infograph, Painting, Quickdraw, Real and Sketch) and 600k images with 345 classes. Note that, because the dataset of 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, for example, the "dog" class in the "clipart" domain has 70 image samples while has 782 image samples in the "real" domain. And, when changing another domain as target domain, such large source domain will also change. Besides, the label shift issue is also existed in DomainNet (Tan et al., 2020).

Figure 7: Some examples covering different domains of datasets we used.

A.2 TRAINING DETAILS

In all experiments, SGD with momentum is used as the optimizer and the cosine annealing rule (Loshchilov & Hutter, 2017) is adopted for learning rate decay. All our experiments are implemented on PyTorch and conducted on a single 12G NVIDIA 1080ti GPU.

For Digit-Five, the CNN backbone is constructed with three convolution layers and two fully connected layers, termed as Cov_3FC_2 following (Peng et al., 2019). For each mini-batch, we sample 64 images for training. The model is trained with an initial learning rate of 0.05 for totally 30 epochs.

For Office-31 and Office-Home, following (Long et al., 2018; Zhang et al., 2019a; Cui et al., 2020b), we use ResNet-50 as backbone. The initial learning rate is set to 1e-3. The input image size is

224×224 and the batch size is 36. We train the models for 500 epochs (nearly 16,000 iterations) and evaluate their adaptation performance. We use the default train/test/val split protocol as (Cui et al., 2020b; Long et al., 2018) for both the two datasets.

For VisDA-2017, following (Long et al., 2018; Cui et al., 2020b), we also use ResNet-50 as backbone. The initial learning rate is set to 1e-4. The input image size is 224×224, and the batch size is 36. We follow the train/val/test split protocol of (Cui et al., 2020b) and train the models for 150 epochs.

For DomainNet, we use ResNet-101 as the CNN backbone, the same as (Peng et al., 2019), and sample from each domain 6 images to form a mini-batch. The model is trained with an initial learning rate of 0.002 for 40 epochs.

A.3 Source Code

We have uploaded the source code that corresponds to our proposed dynamic adversarial domain adaptation (DADA) method, with dynamic domain label (DDL). Please find details and reproduce the main experimental results in the uploaded supplementary materials of 'Dynamic_Adversarial_Domain_Adaptation.zip'.

A.4 More Details about Toy Experiments

Random Point Classification. In the main manuscript, we observe the behavior of our proposed training strategy of Dynamic Adversarial Domain Adaptation (DADA) method with dynamic domain label (DDL) on toy problem of *2D random point classification*, in which we used *numpy.random* (Oliphant, 2006) to synthesize the toy source and target samples that share the same label space for validation.

For the network structure, we use the totally same architecture for two schemes of *Baseline* and *Ours*. We adopt a multilayer perceptron (MLP) (Gardner & Dorling, 1998) as the feature extractor F (refer to Eq. (1) of the main manuscript for notation), which MLP is composed of three fully connected layers with BatchNorm1d and ReLU layers for stable training. The category classifier C is an one-layer fully connected layer followed by a sigmoid function to output the classification result ('0' – red point or '1' – green point). For the domain discriminator D (refer to Eq. (2) of the main manuscript for notation), it is also composed of three fully connected layers with inserted dropout and ReLU layers for stable training following (Long et al., 2018; Cui et al., 2020b), followed by a sigmoid function to output the domain classification result. A gradient reversal layer (GRL) (Ganin & Lempitsky, 2015; Ganin et al., 2016; Long et al., 2018) is used to connect feature extractor F and domain discriminator D to achieve the adversarial function by multiplying the gradient from D with a certain negative constant during the back-propagation to the feature extractor F.

For the basic optimization hyper-parameters, we employ stochastic gradient descent (SGD) as optimizer with an initial learning rate of 0.01 train all the schemes of *Baseline* and *Ours*. Batch size is set as 100 and total training epoch is set as 10.

Inter-twinning Moons. For this toy problem, we conduct experiment fully based on the codebase [1] released by (Ganin et al., 2016), we recommend readers to get more details from their original paper.

A.5 More Experimental Results

Feature Distributions Visualization. Here, we further visualize the learned feature distributions by t-SNE (Saito et al., 2019) for *W→A* setting of Office31 in Figure 8. We observe the scheme of *Source Only* that without considering domain adaptation only works well in source domain but poorly in target domain. The adversarial training based baseline scheme of *DANN* (Ganin et al., 2016) aligns most samples in the source and target domains well. When applying the proposed dynamic domain label (DDL) technique into DANN, the scheme of *DANN + DDL (ours)* achieves a much better domain alignment results, where the clusters with the same class are more compact and less data points scatter at the boundaries between clusters. This group of visualization results validates the effectiveness of our DDL for adversarial domain adaptation.

Feature Map Visualization. Except t-SNE visualization results, in Figure 9, we also visualize the learned feature maps of *DANN (Baseline)* and *DANN + DDL (ours)* by Grad-CAM (Selvaraju et al.,

[1] https://github.com/GRAAL-Research/domain_adversarial_neural_network

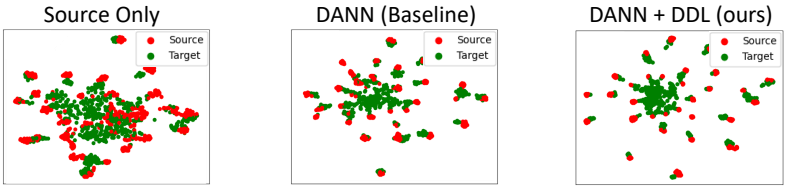


Figure 8: Visualization of t-SNE distributions, where samples are from source webcam (W) and target amazon (A) domains of Office31.

2017) w.r.t. object category classification. This group of experimental visualization aims to explore whether dynamic domain label (DDL) could help the feature extractor to learn better domain-invariant and object-focus visual representations. We see that the baseline scheme *DANN (Baseline)* is prone to ignore some discriminative regions, which impedes the transferability across domains. In contrast, with DDL, the learned feature representations could better focus on the discriminative regions that related to foreground objects, enabling a higher classification accuracy.

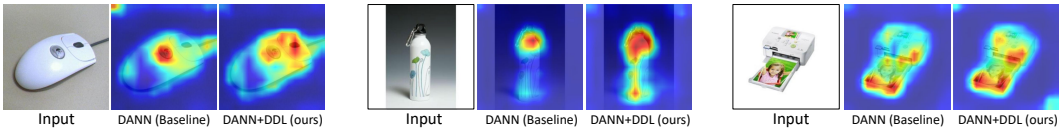


Figure 9: Visualization of inner feature maps, where samples are also from source webcam (W) and target amazon (A) domains of Office31.

Clarification of Viewing ADA from Memorization Perspective. Our work is the first work to view the adversarial domain adaptation problem from a neural network memorization perspective, and point out that the imbalanced distribution defect will make the majority data clusters dominate/biase the adaptation process. To prove that, here we deliberately provide empirical evidence to show that standard adversarial DA methods tend to systematically align majority samples before minority samples.

First, our main manuscript (the third paragraph in the introduction section) has given some references that related to deep network memorization: DNNs learn simple patterns first, before memorizing. In other words, DNN optimization is content-aware, taking advantage of patterns shared by multiple training examples (Arpit et al., 2017). Therefore, the DNN-based adversarial DA methods tend to align well those majority samples before minority samples in the adaptation process.

Moreover, we further conduct a toy experiment as evidence to support the above statement: the majority "bike" samples (90%) in "Amazon" domain of Office31 have no background scene (empty) while minority "bike" samples have real-world background instead (which makes these minority samples look like come from "Webcam" domain). We use CDAN as baseline, and train it for just 5 epochs in the "A→W" setting on Office31. Then, we test its category classification accuracy and domain distinguishment accuracy w.r.t the majority "bike" samples and the minority "bike" samples, respectively. We found that 1). the object classification average accuracy (recognized as "bike") on the former majority outperforms that on the latter minority by 37.9% (77.9% vs. 40.0%); 2). the domain distinguishment accuracy (recognized as "Amazon") outperforms by 72.2% (92.2% vs. 20%).