Dual Mixup Regularized Learning for Adversarial Domain Adaptation

Yuan Wu[1], Diana Inkpen[2], and Ahmed El-Roby[1]

[1] Carleton University {yuan.wu3, Ahmed.ElRoby}@carleton.ca
[2] University of Ottawa Diana.Inkpen@uottawa.ca

Abstract. Recent advances on unsupervised domain adaptation (UDA) rely on adversarial learning to disentangle the explanatory and transferable features for domain adaptation. However, there are two issues with the existing methods. First, the discriminability of the latent space cannot be fully guaranteed without considering the class-aware information in the target domain. Second, samples from the source and target domains alone are not sufficient for domain-invariant feature extracting in the latent space. In order to alleviate the above issues, we propose a dual mixup regularized learning (DMRL) method for UDA, which not only guides the classifier in enhancing consistent predictions in-between samples, but also enriches the intrinsic structures of the latent space. The DMRL jointly conducts category and domain mixup regularizations on pixel level to improve the effectiveness of models. A series of empirical studies on four domain adaptation benchmarks demonstrate that our approach can achieve the state-of-the-art.

Keywords: domain adaptation, Mixup, regularization

1 Introduction

The development of deep neural networks has significantly improved the state of the arts for a wide variety of machine learning tasks, such as computer vision[13], speech recognition[11], and reinforcement learning [23]. However, these advancements often rely on the existence of a large amount of labeled training data. In many real-world applications, collecting sufficient labeled data is often prohibitive due to time, financial, and expertise constraints. Therefore, there is a strong motivation to train an effective predictive model which can leverage knowledge learned from a label-abundant dataset and perform well on another label-scarce domains [17]. However, due to the existence of domain shift, deep neural networks trained on one large scale labeled dataset can be weak at generalizing learned knowledge to new datasets and tasks [17].

To address the above issue, a general strategy called domain adaptation is introduced by transferring knowledge from a label-rich domain, referred as the source domain, to a label-scarce domain, referred as the target domain [1]. Unsupervised domain adaptation addresses a more challenging scenario where there is no labeled data in the target domain. A theoretical analysis of domain adaptation

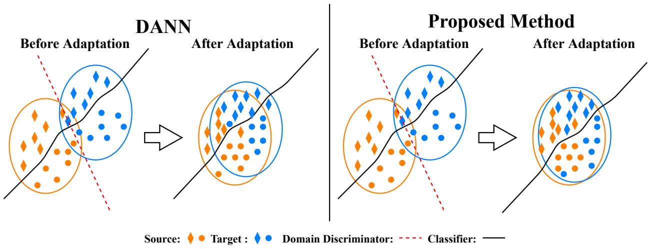


Fig. 1: Comparison of DANN and the proposed method. **Left**: DANN only tries to match the feature distribution by utilizing adversarial learning; it does not consider the class-aware information in the target domain and samples from the source and target domains may not be sufficient to ensure domain-invariance of the latent space. **Right**: Our proposed method uses category mixup regularization to enforce prediction consistency in-between samples and domain mixup regularization to explore more intrinsic structures across domains, resulting in better adaptation performance.

is introduced by [1], it suggests that in UDA tasks, the risk on the target domain can be bounded by the risk of a model on the source domain and the discrepancy between distributions of the two domains. Early UDA methods learned to reduce the discrepancy between domains in a shallow regime [7] or to re-weight source instances based on their relevance to the target domain [6]. Later on, Maximum Mean Discrepancy (MMD) [9] was proposed to measure the distribution difference between source and target domains [25, 14, 16]. More recently, the UDA models are largely built on deep neural networks, and focus on learning domain-invariant features across domains by using adversarial learning [4]. Adversarial domain adaptation models can learn discriminative and domain-invariant features across domains by playing a minimax game between a feature extractor and a domain discriminator. The domain discriminator is trained to tell whether the sample comes from the source domain or target domain, while the feature extractor is learned to fool the domain discriminator. Many recent UDA methods based on adversarial learning can achieve the state-of-the-art performance [20, 15, 30].

Even though adversarial domain adaptation method has shown impressive performance for various tasks, such as image classification [15] and semantic segmentation [20]. This approach still faces two issues: First, the adversarial domain adaptation does not take class-aware information in the target domain into consideration. Second, as we always use mini-batch stochastic gradient descent (SGD) for optimization in practice, if the batch size is small, samples from the source and target domains may not be sufficient to guarantee the domain-invariance in the latent space. Therefore, after adaptation, as shown in the left side of Figure 1, the classifier may falsely align target samples of one label with samples of a different label in the source domain, which leads to inconsistent predictions.

In this paper, we propose a dual mixup regularized learning (DMRL) method, which implements category and domain mixup regularizations on pixel level to address the aforementioned issues for unsupervised domain adaptation. Mixup has the ability of generating convex combinations of pairs of training samples and their corresponding labels. Motivated by this data augmentation technique, we propose two effective regularization mechanisms including domain-level mixup regularization and category-level mixup regularization, which play crucial roles in reducing the domain discrepancy for unsupervised domain adaptation. In particular, category mixup regularization is used to enforce consistent predictions in the latent space, which is conducted on both source and target domains, achieving a stronger discriminability of the latent space. Domain mixup regularization can reveal more mixed instances within each domain and allows the model to enrich internal feature patterns in the latent space, which can lead to a more continuous domain-invariant latent space and help match the global domain statistics across different domains. By using the two mixup-based regularization mechanisms, our model can effectively generate discriminative and domain-invariant representations. Empirical studies on four benchmarks demonstrate the performance of our approach. The contributions of our paper are summarized as follows:

- We propose a dual mixup regularized learning method which can project the source and target domains to a common latent space, and efficiently transfer the knowledge learned from the labeled source domain to the unlabeled target domain.
- The proposed regularization mechanisms (category-level mixup regularization and domain-level mixup regularization) can learn discriminative and domain-invariant representations effectively to help reduce the distribution discrepancy across different domains.
- We empirically confirm the effectiveness of our proposed method by evaluating it on four benchmark datasets. Conducting ablation studies and parameter sensitivity analysis to validate the contributions of different components in our model and evaluate how each hyperparameter influences the performance of our method.

2 Related Work

This work builds on two threads of research: interpolation-based regularization and domain adaptation. In this section, we briefly overview methods that are related to these two tasks.

2.1 Interpolation-based Regularization

Interpolation-based regularization has been recently proposed for supervised learning [29, 26], it can help models to alleviate issues such as instability in adversarial training and sensitivity to adversarial samples. In particular, Mixup

[29] is proposed to train models on virtual examples constructed as the convex combinations of pairs of inputs and labels. It can also encourage models to have a strictly linear behavior between training samples, by smoothing the models' output for a convex combination of two inputs to the convex combination of the outputs of each individual input [2]. Moreover, Mixup can also be used to guarantee consistent predictions in the data distribution [27], which can induce the separation of samples into different labels [3]. More recently, various variants of Mixup have been studied. A manifold extension to Mixup, Manifold-Mixup [26], proposes to perform interpolation in the latent space representations. [2] exploits interpolations in the latent space generated by an autoencoder to improve performance.

2.2 Domain Adaptation

The main goal of domain adaptation is to transfer the knowledge learned from a label-abundant domain to a label-scarce domain. Unsupervised domain adaptation tackles a more challenging scenario where the target domain has no labeled data at all. Deep neural network based methods have been widely studied for UDA. The Deep Domain Confusion (DDC) method leverages Maximum Mean Discrepancy (MMD) [9] metric in the last fully-connected layer to learn representations that are both discriminative and transferable [25]. [14] proposes a Deep Adaptation Network (DAN) to enhance the feature transferability by minimizing the multi-kernel MMD in several task specific layers. Asymmetric Tri-Training (ATT) exploits three different networks to generates pseudo-labels for target domain samples and utilizes these pseudo-labels to train the final classifier [19]. Joint Adaptation Networks (JAN) learn a transfer network by aligning the joint distributions of multiple domain-specific layers across different domains based on a joint maximum mean discrepancy criterion [16]. [31] proposes a Confidence Regularized Self-Training (CRST) framework to construct the soft pseudo-label, smoothing the one-hot pseudo-label to a conservative target distribution.

More recently, unsupervised domain adaptation methods are largely focusing on learning domain-invariant features by using adversarial training [4]. Generative Adversarial Network (GAN) is proposed in [8], which plays a minimax game between two networks: the discriminator is trained by minimizing the binary classification error of distinguishing the real images from the generated ones, while the generator is learned to generate high-quality images that are indistinguishable by the discriminator. Motivated by GAN, [4] proposes a domain adversarial neural network (DANN) that can learn discriminative and domain-invariant features by exploiting adversarial learning between a feature extractor and a domain discriminator. Many recent works have adopted the adversarial learning mechanism and achieved the state-of-the-art performance for unsupervised domain adaptation. The Adversarial Discriminative Domain Adaptation method (ADDA) uses an untied weight sharing strategy to align the feature distributions of source and target domains [24]. The Maximum Classification Discrepancy (MCD) utilizes different task-specific classifiers to learn a feature extractor that can generate category-related discriminative features [20]. [12] proposes Cycle-Consistent Adversarial Domain Adaptation (CyCADA) which implements domain adaptation at both pixel-level and feature-level by using cycle-consistent adversarial training. Multi-Adversarial Domain Adaptation (MADA) [18] can exploit multiplicative interactions between feature representations and category predictions to enforce adversarial learning. Generate to Adapt (GTA) provides an adversarial image generation approach that directly learns a joint feature space in which the distance between the source and target domains can be minimized [21]. Conditional Domain Adversarial Network (CDAN) conditions the domain discriminator on a multilinear map of feature representations and category predictions so as to enable discriminative alignment of multi-mode structures [15]. Consensus Adversarial Domain Adaptation (CADA) [30] enforces the source and target encoders to achieve consensus to ensure the domain-invariance of the latent space.

Our proposed DMRL can be regarded as an extension of this line of research by introducing both category and domain mixup regularizations on pixel level to solve complex, high dimensional unsupervised domain adaptation tasks.

3 Method

In this paper, we consider unsupervised domain adaptation in the following setting. We have a labeled source domain $D^s = \{(x_i^s, y_i^s)\}_{i=1}^{n^s}$ with $x_i^s \in \mathcal{X}$ and $y_i^s \in \mathcal{Y}$, and an unlabeled target domain $D^t = \{x_i^t\}_{i=1}^{n^t}$ with $x_i^t \in \mathcal{X}$. The data in two domains are sampled from two distributions P_S and P_T. P_S and P_T are assumed to be different but related (refereed as covariate shift in the literature [22]). The target task is assumed to be the same with the source task. Our ultimate goal is to utilize the labeled data in the source domain to learn a predictive model $h : \mathcal{X} \mapsto \mathcal{Y}$ which can generalize well on the target domain.

3.1 Adversarial Domain Adaptation

Motivated by the domain adaptation theory [1] and GANs [8], [4] proposes Domain Adversarial Neural Network (DANN), which can learn the domain-invariant features that are generalizable across domains. The standard DANN consists of three components: a feature extractor G, a category classifier C and a domain discriminator D. We consider a feature extractor $G : \mathcal{X} \mapsto \mathbb{R}^m$, which maps any input instance $\mathbf{x} \in \mathcal{X}$ from the input space $\mathcal{X}$ into the latent space $G(\mathbf{x}) \in \mathbb{R}^m$; a category classifier $C : \mathbb{R}^m \mapsto \mathcal{Y}$, which transforms a feature vector in the latent space into the output label space $\mathcal{Y}$; a domain discriminator $D : \mathbb{R}^m \mapsto [0, 1]$, which distinguishes the source domain data (with domain label 1) from the target domain data (with domain label 0). By training G adversarially to confuse D, DANN can learn domain-invariant features to bridge the divergence between domains. Formally, the DANN can be formulated as:

$$\min_G \max_D \quad \mathcal{L}_c(G, C) + \lambda_d \mathcal{L}_{Adv}(G, D) \tag{1}$$

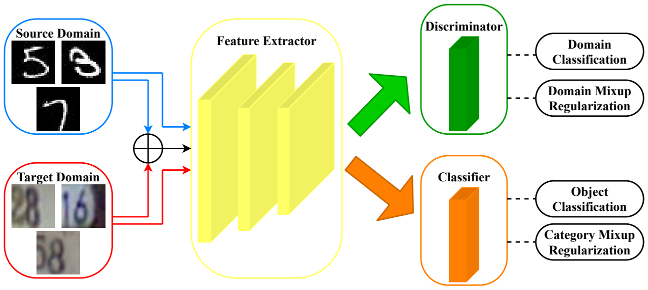


Fig. 2: The architecture of the proposed dual mixup regularized learning (DMRL) method. Our DMRL consists of two mixup-based regularization mechanisms, including category-level mixup regularization and domain-level mixup regularization, which can enhance discriminability and domain-invariance of the latent space. The feature extractor G aims to learn discriminative and domain-invariant features, the domain discriminator D is trained to tell whether the sampled feature comes from the source domain or target domain, and the classifier C is used to conduct object classification.

$$\mathcal{L}_c(G, C) = \mathbb{E}_{(\mathbf{x}^s, y^s)\sim D^s} \ell(C(G(\mathbf{x}^s)), y^s) \tag{2}$$

$$\mathcal{L}_{Adv}(G, D) = \mathbb{E}_{\mathbf{x}^s \sim D^s} \log D(G(\mathbf{x}^s)) + \mathbb{E}_{\mathbf{x}^t \sim D^t} \log(1 - D(G(\mathbf{x}^t))) \tag{3}$$

where $\ell(\cdot, \cdot)$ is the canonical cross-entropy loss, and λ_d is a trade-off hyperparameter.

3.2 Dual Mixup Regularization

In this work, we propose a dual mixup regularized learning (DMRL) method based on adversarial domain adaptation. This method conducts category and domain mixup on pixel level. In general, Mixup performs data augmentation by constructing virtual samples with convex combinations of a pair of samples and their corresponding labels: $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$:

$$\widetilde{\mathbf{x}} = \mathcal{M}_\lambda(\mathbf{x}_i, \mathbf{x}_j) = \lambda \mathbf{x}_i + (1 - \lambda)\mathbf{x}_j \tag{4}$$
$$\widetilde{y} = \mathcal{M}_\lambda(y_i, y_j) = \lambda y_i + (1 - \lambda) y_j \tag{5}$$

where λ is randomly sampled from a beta distribution $Beta(\alpha, \alpha)$ for $\alpha \in (0, \infty)$. By encouraging linear interpolation regularization in-between training samples, Mixup has been demonstrated effective in both supervised and semi-supervised learning [29, 27].

We largely enhance the ability of prior adversarial-learning-based domain adaptation methods by our proposed dual mixup regularized learning module,

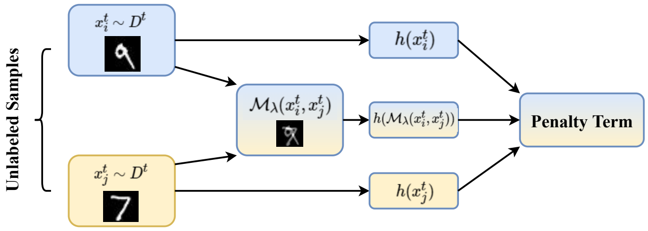


Fig. 3: The framework of unlabeled category mixup regularization.

which jointly conducts category and domain mixup on pixel level to guide adversarial learning. Figure 2 depicts the whole framework of our proposed method. For the input, images are linearly mixed by pixel-wise addition within each individual domain. Therefore, in the input space, there exists four kinds of samples: source samples, target samples, mixed source samples obtained by mixing two source samples, and mixed target samples obtained by mixing two target samples. After that, there exists two streams. For one stream, features of the source and target domains are used to align the global distribution statistics and conduct domain mixup regularization by the domain discriminator D. For the other stream, object classification and category mixup regularization are implemented by the classifier C. More details are provided in the following parts.

Category Mixup Regularization The category mixup regularization consists of two components: labeled category mixup regularization and unlabeled category mixup regularization. For the source domain, since we have the labeled samples, we directly use mixed source samples and their corresponding labels to enforce prediction consistency:

$$\mathcal{L}_s^r(G, C) = \mathbb{E}_{(\mathbf{x}_i^s, y_i^s), (\mathbf{x}_j^s, y_j^s) \sim D^s} \ell(h(\widetilde{\mathbf{x}}^s), \widetilde{y}^s) \tag{6}$$

where h denotes the composition of the feature extractor G and the classifier C, $h = G \circ C$, and it can be treated as the classification function in the input space.

For the target domain, we have no access to the label information. Therefore, mixup needs to be applied on the pseudo-labels generated by the classifier C. Specifically, we replace y_i^t and y_j^t with $h(\mathbf{x}_i^t)$ and $h(\mathbf{x}_j^t)$, which are the current predictions of C. Literally, $h(\mathbf{x}_i^t)$ and $h(\mathbf{x}_j^t)$ are termed as pseudo-labels of $(\mathbf{x}_i^t)$ and $(\mathbf{x}_j^t)$. Figure 3 presents the process of unlabeled category mixup regularization. First, we should construct convex combinations, denoted as $(\widetilde{\mathbf{x}}^t, \mathcal{M}_\lambda(h(\mathbf{x}_i^t), h(\mathbf{x}_j^t)))$, of pairs of samples $(\mathbf{x}_i^t, \mathbf{x}_j^t)$ and their virtual labels $(h(\mathbf{x}_i^t), h(\mathbf{x}_j^t))$. Then we conduct regularization by enforcing $h(\widetilde{\mathbf{x}}^t)$ to be consistent with $\mathcal{M}_\lambda(h(\mathbf{x}_i^t), h(\mathbf{x}_j^t))$ via a penalty term:

$$\mathcal{L}_t^r(G, C) = \mathbb{E}_{\mathbf{x}_i^t, \mathbf{x}_j^t \sim D^t} Dis(h(\widetilde{x}^t), \mathcal{M}_\lambda(h(\mathbf{x}_i^t), h(\mathbf{x}_j^t))) \tag{7}$$

where $Dis(\cdot,\cdot)$ denotes the penalty term which can punish the difference between $h(\widetilde{\mathbf{x}}^t)$ and $\mathcal{M}_\lambda(h(\mathbf{x}_i^t), h(\mathbf{x}_j^t))$, encouraging a linear behavior in-between training samples and λ equals to the one used in labeled category mixup regularization. In our experiments, we use $L1$-Norm function as the penalty term. In general, Category mixup regularization can smooth the output distribution by constructing neighboring samples of the training samples and enforce prediction consistency between the neighboring and training samples, which exploits the class-aware information of the target domain in the training process, leading to performance improvement.

Domain Mixup Regularization In practice, as we use mini-batch SGD in training, samples from the source and target domains alone are usually insufficient for global distribution alignment. Domain mixup has the ability of generating more intermediate samples and exploring more internal structures within each domain. Linear interpolations of each domain can be generated in the input space, and the domain mixup regularization term can be defined as follows:

$$\mathcal{L}_{Adv}^{r}(G,D) = \mathbb{E}_{\mathbf{x}_i^s,\mathbf{x}_j^s \sim D^s} \log D(G(\widetilde{\mathbf{x}}^s)) + \mathbb{E}_{\mathbf{x}_i^t,\mathbf{x}_j^t \sim D^t} \log(1 - D(G(\widetilde{\mathbf{x}}^t))) \tag{8}$$

where λ is the same as the one used in category mixup regularization. Contrary to prior adversarial-learning-based domain adaptation methods [4, 24, 15], not only the samples from source and target domains, but also the mixed samples are used to align the global distribution statistics. Finally, in our proposed domain mixup regularization applied to adversarial domain adaptation, the domain-invariance of the latent space is expected to be enhanced, and with category mixup regularization, the learned representations can be more discriminative. Formally, the DMRL method can be formulated as:

$$\min_{G,C}\max_{D} \quad \mathcal{L}_c(G,C) + \lambda_s\mathcal{L}_s^r(G,C) + \lambda_t\mathcal{L}_t^r(G,C) + \lambda_d\mathcal{L}_{Adv}(G,D) + \lambda_r\mathcal{L}_{Adv}^r(G,D) \tag{9}$$

where λ_s, λ_t, λ_d and λ_r are hyperparameters for trading off different losses.

3.3 Training Procedure

The training algorithm of DMRL which uses mini-batch SGD is presented in Algorithm 1. In each iteration, we first mix the samples in the input space for the source and target domains, separately. Then the mixed samples and their constitutions are used to guide the adversarial learning and conduct two mixup-based regularizations. The category mixup regularization can enforce consistent prediction constraint, while the domain mixup regularization can enrich the feature patterns in the latent space, so that the learned representation can be both discriminative and domain-invariant. α is a hyperparameter that controls the selection of λ. λ_s, λ_t, λ_d and λ_r are hyperparameters that balance different losses. In our experiments, we set α, λ_s and λ_r as 0.1, 0.0001 and 0.00001. According

Algorithm 1 Stochastic gradient descent training algorithm of DMRL

```
**Input:** Source domain: $D^s$, target domain: $D^t$ and batch size: $N$.
**Output:** Configurations of DMRL
**Initialize** $\alpha$, $\lambda_s$, $\lambda_t$, $\lambda_d$ and $\lambda_r$
**for** number of training iterations **do**
    $(\mathbf{x}^s, y^s) \leftarrow$ RANDOMSAMPLE$(D^s, N)$
    $(\mathbf{x}^t) \leftarrow$ RANDOMSAMPLE$(D^t, N)$
    $\lambda \leftarrow$ RANDOMSAMPLE$(Beta(\alpha, \alpha))$
    $(\widetilde{x}^s, \widetilde{y}^s) \leftarrow$ Eq. (4, 5)    #get mixed images for the source domain
    $(\widetilde{x}^t) \leftarrow$ Eq.(4)    #get mixed images for the target domain
    Calculate $l_D = \lambda_d \mathcal{L}_{Adv}(G, D) + \lambda_r \mathcal{L}^r_{Adv}(G, D)$;
    Update $D$ by ascending along gradients $\nabla l_D$.
    Calculate $loss = \mathcal{L}_c(G, C) + \lambda_s \mathcal{L}^r_s(G, C) + \lambda_t \mathcal{L}^r_t(G, C) + \lambda_d \mathcal{L}_{Adv}(G, D) + \lambda_r \mathcal{L}^r_{Adv}(G, D)$;
    Update $G$, $C$ by descending along gradients $\nabla loss$.
**end for**
```

to our parameter sensitivity analysis, the values of α, λ_s and λ_r do not influence much the adaptation performance of our approach, and these hyperparameters are kept fixed in all experiments. The value of λ_t has an influence on the adaptation performance, so λ_t is selected via tuning on the unlabeled test data for different tasks. λ_d is adapted according to the strategy from [5].

3.4 Discussion

For the adversarial-learning-based domain adaptation methods, both the domain label and category label play crucial roles in filling the gap between the source and target domains. Specifically, domain labels are used to help make global distribution alignment across different domains, and category labels can enable features to be discriminative [4, 24, 15]. These two types of information can help reduce the domain discrepancy in different aspects, and complement each other for domain adaptation. However, prior adversarial domain adaptation methods suffer two limitations: (1) The lack of class-aware information of the target domain can make the extracted features less discriminative; (2) As we use mini-batch SGD in training, samples from source and target domains do not allow models to completely explore internal feature patterns in the latent space.

In our method, we conduct two mixup-based regularizations to alleviate the above issues. First, we apply category mixup regularization on source and target domains. Specifically, for unlabeled target data, pseudo-labels are introduced. Since there are obviously false labels as target pseudo-labels, consistent prediction constraint is exploited to suppress the detrimental influence brought by the false target pseudo-labels. Moreover, category mixup regularization can smooth the output distribution of the model by encouraging the model to behave linearly in-between training samples. Then, Mixup [29], as an effective data augmentation technique, is expected to provide extra mixed samples for adversarial learning. Domain mixup regularization is used to explore more internal feature patterns in the latent space, which leads to a more continuous domain-invariant latent distribution. In summary, our proposed approach can learn discriminative and domain-invariant representations and improve the performance of different unsupervised domain adaptation tasks.

4 Experiments

We evaluate our proposed DMRL on unsupervised domain adaptation tasks of four benchmarks, validate the effectiveness of different components in detail and investigate the influences of different hyperparameters.

4.1 Setup

Dataset We conducted experiments on four domain adaptation benchmarks, **Office-31**, **ImageCLEF-DA**, **VisDA-2017** and **Digits**. Office-31 is a benchmark domain adaptation dataset containing images belonging to 31 classes from three domains: Amazon (A) with 2,817 images, Webcam (W) with 795 images and DSLR (D) with 498 images. We evaluate all methods on six domain adaptation tasks: $\mathbf{A} \rightarrow \mathbf{W}$, $\mathbf{D} \rightarrow \mathbf{W}$, $\mathbf{W} \rightarrow \mathbf{D}$, $\mathbf{A} \rightarrow \mathbf{D}$, $\mathbf{D} \rightarrow \mathbf{A}$, and $\mathbf{W} \rightarrow \mathbf{A}$.

ImageCLEF-DA is a benchmark dataset for ImageCLEF 2014 domain adaptation challenges, which contains 12 categories shared by three domains: Caltech-256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC 2012 (P). Each domain contains 600 images and 50 images for each category. The three domains in this dataset are of the same size, which is a good complementation of the Office-31 dataset where different domains are of different sizes. We build six domain adaptation tasks: $\mathbf{I} \rightarrow \mathbf{P}$, $\mathbf{P} \rightarrow \mathbf{I}$, $\mathbf{I} \rightarrow \mathbf{C}$, $\mathbf{C} \rightarrow \mathbf{I}$, $\mathbf{C} \rightarrow \mathbf{P}$, and $\mathbf{P} \rightarrow \mathbf{C}$.

VisDA-2017 is a large simulation-to-real dataset, with over 280,000 images of 12 classes in the combined training, validation, and testing domains. The source images were obtained by rendering 3D models of the same object classes as in the real data from different angles and under different lighting conditions. It contains 152,397 synthetic images. The validation and test domains comprise natural images. The validation one has 55,388 images in total. We use the training domain as the source domain and validation domain as the target domain.

For Digits domain adaptation tasks, we explore three digit datasets: **MNIST**, **USPS** and **SVHN**. Each dataset contains digit images of 10 categories (0-9). We adopt the experimental settings of CyCADA [12] with three domain adaptation tasks: MNIST to USPS (**MNIST→USPS**), USPS to MNIST (**USPS→MNIST**), and SVHN to MNIST (**SVHN→MNIST**).

Comparison Methods We compare our proposed DMRL with state-of-the-art models. For Office-31, we compare with Deep Adaptation Network (DAN) [14], Domain Adversarial Neural Network (DANN) [4], Joint Adaptation Network (JAN) [16], Generate to Adapt (GTA) [21], Multi-Adversarial Domain Adaptation (MADA)[18], Conditional Domain Adaptation Network (CDAN) [15] and Confidence Regularized Self-Training (CRST). For ImageCLEF-DA, we compare with DAN, DANN, JAN, MADA and CDAN. For VisDA-2017, We compare with DANN, DAN, JAN, GTA, Maximum Classifier Discrepancy (MCD) [20] and CDAN. To further validate our method, we also conduct experiments on digit datasets, including MNIST, USPS and SVHN, we compare with DANN, Adversarial Discriminative Domain Adaptation (ADDA)[24], Cycle-consistent Adversarial Domain Adaptation (CyCADA)[12], MCD, CDAN and Consensus Adversaria Domain Adaptation (CADA)[30].

Implementation Details We follow standard evaluation protocols for unsupervised domain adaptation [20, 15]. For Office-31 and ImageCLEF-DA datasets, we utilize ResNet50 [10] pre-trained on ImageNet [13] as the backbone. For each domain adaptation task of the Office-31 and ImageCLEF-DA datasets, we report classification results of mean ± standard error over three random trials. For VisDA-2017 dataset, we follow [20] and use ResNet101 [10] pre-trained on ImageNet [13] as the backbone. For Digits datasets, we use a modified version of Lenet architecture as the base network, and train the models from scratch. For each backbone, we use all its layers up to the second last one as the feature extractor G and replace the last full-connected layer with a task-specific fully-connected layer as the category classifier C. For discriminator, we use the same architecture as DANN [4].

We adopt mini-batch SGD with momentum of 0.9 and the learning rate annealing strategy as [5]: the learning rate is adjusted by $\eta_p = \frac{\eta_0}{(1+\theta p)^\beta}$, where p denotes the process of training epochs that is normalized to be in [0,1], and we set $\eta_0 = 0.01$, $\theta = 10$, $\beta = 0.75$, which are optimized to promote convergence and low errors on the source domain. λ_d is progressively changed from 0 to 1 by multiplying to $\frac{1-exp(-\delta p)}{1+exp(-\delta p)}$, where $\delta = 10$. For all experiments, we set the hyper-parameter α of distribution $Beta(\alpha, \alpha)$ to 0.2 as used in [29]. λ_s and λ_r are fixed as 0.0001 and 0.00001 respectively. λ_t is chosen in the range $\{0.1, 1, 2, 5, 6, 10\}$, we select it on a per-experiment basis relying on unlabeled target data.

4.2 Results

The unsupervised domain adaptation results in terms of classification accuracy in the target domain on Office-31, ImageCLEF-DA, VisDA-2017 and Digits datasets are reported in Table 1, 2 and 3 respectively, with results of comparison methods directly cited from their original papers wherever available.

Results on the Office-31 dataset are presented in Table 1. The results of ResNet-50 trained with only source domain data serve as the lower bound. Our proposed approach can obtain the best performance in four of six tasks: D → W, W → D, A → D, and D → A. For task W → A, DMRL can produce competitive accuracy comparing with the state-of-the-art. It is noteworthy that DMRL results in improved accuracies in two hard transfer tasks: A → D, and D → A. For two easier tasks: D → W and W → D, our approach achieves accuracy no less than 99.0%. Moreover, our method can achieve the best average domain adaptation accuracy on this dataset. Given the fact that the number of samples

Table 1: Accuracy (%) on Office-31 .

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet-50 [10]	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DAN [14]	80.5±0.4	97.1±0.2	99.6±0.1	78.6±0.2	63.6±0.3	62.8±0.2	80.4
DANN [4]	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
JAN [16]	85.4±0.3	97.4±0.2	99.8±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
GTA [21]	89.5±0.5	97.9±0.3	99.8±0.4	87.7±0.5	72.8±0.3	**71.4**±0.4	86.5
MADA [18]	90.0±0.1	97.4±0.1	99.6±0.1	87.8±0.2	70.3±0.3	66.4±0.3	85.2
CDAN [15]	**93.1**±0.2	98.2±0.2	**100.0**±0.0	89.8±0.3	70.1±0.4	68.0±0.4	86.6
CRST [31]	89.4±0.7	98.9±0.4	**100.0**±0.0	88.7±0.8	72.6±0.7	70.9±0.5	86.8
DMRL (Proposed)	90.8±0.3	**99.0**±0.2	**100.0**±0.0	**93.4**±0.5	**73.0**±0.3	71.2±0.3	**87.9**

per category is limited in the Office-31 dataset, these results can demonstrate that our method manages to improve the generalization ability of adversarial domain adaptation in the target domain.

Table 2: Accuracy (%) on ImageCLEF-DA.

Method	I→P	P→I	I→C	C→I	C→P	P→C	Avg
ResNet-50 [10]	74.8±0.3	83.9±0.1	91.5±0.3	78.0±0.2	65.5±0.3	91.2±0.3	80.7
DAN [14]	74.5±0.4	82.2±0.2	92.8±0.2	86.3±0.4	69.2±0.4	89.8±0.4	82.5
DANN [4]	75.0±0.6	86.0±0.3	96.2±0.4	87.0±0.5	74.3±0.5	91.5±0.6	85.0
JAN [16]	76.8±0.4	88.0±0.2	94.7±0.2	89.5±0.3	74.2±0.3	91.7±0.3	85.8
MADA [18]	75.0±0.3	87.9±0.2	96.0±0.3	88.8±0.3	75.2±0.2	92.2±0.3	85.8
CDAN [15]	76.7±0.3	90.6±0.3	97.0±0.4	90.5±0.4	74.5±0.3	93.5±0.4	87.1
DMRL (Proposed)	**77.3**±0.4	**90.7**±0.3	**97.4**±0.3	**91.8**±0.3	**76.0**±0.5	**94.8**±0.3	**88.0**

Results on the ImageCLEF-DA dataset are shown in Table 2, the results reveal several interesting observations: (1) Deep transfer learning methods outperform standard deep learning methods; this validates that domain shifts cannot be captured by deep networks [28]. (2) The three domains in the ImageCLEF-DA dataset are more balanced than those in the Office-31 dataset. We can verify whether the performance of domain adaptation models can be improved when domain size does not change, with these more balanced domains. From Table 2, we can see that our approach can outperform all comparison methods in all transfer tasks but with lower improvements compared to the results of the Office-31 dataset in term of the average accuracy, which validates that the domain size may cause domain shift [16]. (3) our proposed DMRL method can achieve a new state-of-the-art on the ImageCLEF-DA dataset, strongly confirming the effectiveness of our method in aligning the features across domains. Moreover, for three of six tasks: C → I, C → P and P → C, our method can produce results with larger rooms of improvement, which further illustrates the effectiveness of the two mixup-based regularization mechanisms.

Positive results are also obtained on the VisDA-2017 and Digits datasets, as shown in Table 3. For VisDA-2017, our approach achieves the highest accuracy among all compared methods, and exceeds the baseline of ResNet-101 pre-trained

Table 3: Accuracy (%) on Digits and VisDA-2017.

Method	MNIST→USPS	USPS→MNIST	SVHN→MNIST	Avg	Method	Synthetic→Real
No Adaptation [12]	82.2	69.6	67.1	73.0	ResNet-101 [10]	52.4
DANN [4]	90.4	94.7	84.2	89.8	DANN [4]	57.4
ADDA [24]	89.4	90.1	86.3	88.6	DAN [14]	61.1
CyCADA [12]	95.6	96.5	90.4	94.2	JAN [16]	65.7
MCD [20]	92.1	90.0	94.2	92.1	GTA [21]	69.5
CDAN [15]	93.9	96.9	88.5	93.1	MCD [20]	71.9
CADA [30]	**96.4**	97.0	90.9	95.6	CDAN [15]	73.7
DMRL (Proposed)	96.1	**99.0**	**96.2**	**97.2**	**DMRL (Proposed)**	**75.5**

on ImageNet with a great margin. For the Digits datasets, our approach can gain improvements of more than 1.5% on two tasks: USPS → MNIST and SVHN → MNIST, while for the MNIST → USPS task, we can obtain competitive results with the existing approaches.

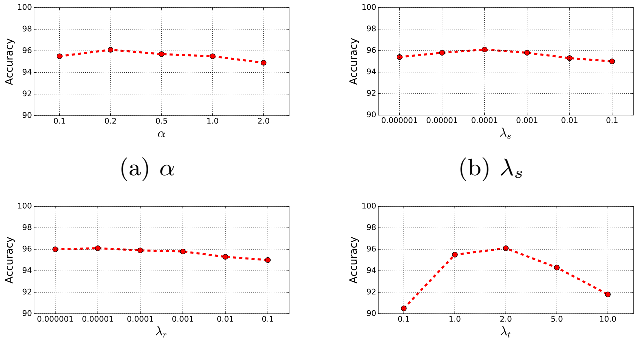
4.3 Further Analysis

Ablation Study We conduct ablation study with two domain adaptation tasks on the Digits dataset, MNIST→USPS and USPS→MNIST, to investigate the contributions of different components in DMRL. First, in order to examine the effectiveness of domain mixup (DM) and category mixup (CM), we produce two variants of DMRL: DMRL(w/o DM) and DMRL(w/o CM). Furthermore, since category mixup consists of two components: labeled category mixup (LCM) and unlabeled category mixup (UDM), DMRL(w/o LCM) and DMRL(w/o UCM) are also taken into consideration. In addition, the very basic baseline "No Adaptation", which simply trains the model on the source domain and tests on the target domain, is included in our study as well. The comparison results are presented in Table 4. The results show that both domain mixup and category mixup can contribute to our method, which verify the effectiveness of these two mixup-based regularizations. In particular, category mixup contributes more to the model than domain mixup. When evaluating two components in category mixup, we can see that the model without unlabeled category mixup produces worse classification accuracies for both tasks, demonstrating that class-aware information in the target domain can make a significant contribution in adversarial domain adaptation. All variants produce inferior results, and the full model with two regularizations produces the best results. This validates the contribution of both category and domain mixup regularization terms.

Parameter Sensitivity Analysis In this section, we discuss the sensitivity of our approach to the values of the hyperparameters α, λ_s, λ_r and λ_t. λ_s, λ_r and λ_t are used to trade-off among losses, and α constrains the selection of λ when conducting Mixup. We evaluate these hyperparameters on the Digits dataset, especially, the MNIST→USPS task. When evaluating one hyperparameter, the others are fixed to their default values (e.g., $\alpha = 2$, $\lambda_s = 0.0001$, $\lambda_r = 0.00001$ and $\lambda_t = 2$). α is tested in the range $\{0.1, 0.2, 0.5, 1.0, 2.0\}$, λ_s and λ_r are explored

Table 4: Ablation Studies.

Method	MNIST→USPS	USPS→MNIST
No Adaptation[12]	82.2	69.6
DMRL(w/o DM)	94.8	97.8
DMRL(w/o CM)	90.3	92.6
DMRL(w/o LCM)	95.0	97.9
DMRL(w/o UCM)	90.7	93.3
DMRL(full)	96.1	99.0



(a) α

(b) λ_s

(c) λ_r

(d) λ_t

Fig. 4: Parameter sensitivity analysis

in the range $\{0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1\}$, and λ_t is evaluated in the range $\{0.1, 1, 2, 5, 6, 10\}$. The experimental results are reported in Figure 4.

From Figure 4, it can be observed that the domain adaptation performance is not sensitive to the hyperparameters α, λ_s and λ_r. Consequently, we can set α, λ_s and λ_r as 0.2, 0.0001 and 0.00001 in all experiments. In addition, with the increase of λ_t, the accuracy increases dramatically and reaches the best value at $\lambda_t = 2.0$, then it decreases rapidly. The parameter sensitivity analysis illustrates that a properly selected λ_t can effectively improve the performance.

5 Conclusion

In this paper, we propose a dual mixup regularized learning (DMRL) framework for adversarial domain adaptation. By conducting category and domain mixup on pixel level, the DMRL cannot only guide the classifier in enhancing consistent predictions in-between samples, which can help avoid mismatches and enforce a stronger discriminability of the latent space, but also explore more internal structures in the latent space, which leads to a more continuous latent space. These two mixup-based regularizations can enhance and complement each other to learn discriminative and domain-invariant representations for target task. The experiments demonstrate that the proposed DMRL can effectively gain performance improvements on unsupervised domain adaptation tasks.

References

1. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine learning **79**(1-2), 151–175 (2010)
2. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.: Mixmatch: A holistic approach to semi-supervised learning. arXiv preprint arXiv:1905.02249 (2019)
3. Chapelle, O., Zien, A.: Semi-supervised classification by low density separation. In: AISTATS. vol. 2005, pp. 57–64. Citeseer (2005)
4. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: ICML. pp. 1180–1189. JMLR. org (2015)
5. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The Journal of Machine Learning Research **17**(1), 2096–2030 (2016)
6. Gong, B., Grauman, K., Sha, F.: Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In: ICML. pp. 222–230 (2013)
7. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: CVPR. pp. 2066–2073 (2012)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
9. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.J.: A kernel method for the two-sample-problem. In: Advances in neural information processing systems. pp. 513–520 (2007)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
11. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., et al.: Deep neural networks for acoustic modeling in speech recognition. IEEE Signal processing magazine **29** (2012)
12. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: ICML (2018)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
14. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: ICML. pp. 97–105. JMLR. org (2015)
15. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems. pp. 1640–1650 (2018)
16. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: ICML. pp. 2208–2217. JMLR. org (2017)
17. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2009)
18. Pei, Z., Cao, Z., Long, M., Wang, J.: Multi-adversarial domain adaptation. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
19. Saito, K., Ushiku, Y., Harada, T.: Asymmetric tri-training for unsupervised domain adaptation. In: ICML. pp. 2988–2997. JMLR. org (2017)

20. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: CVPR. pp. 3723–3732 (2018)
21. Sankaranarayanan, S., Balaji, Y., Castillo, C.D., Chellappa, R.: Generate to adapt: Aligning domains using generative adversarial networks. In: CVPR. pp. 8503–8512 (2018)
22. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of statistical planning and inference **90**(2), 227–244 (2000)
23. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. nature **529**(7587), 484 (2016)
24. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR. pp. 7167–7176 (2017)
25. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014)
26. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Courville, A., Lopez-Paz, D., Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states. arXiv preprint arXiv:1806.05236 (2018)
27. Verma, V., Lamb, A., Kannala, J., Bengio, Y., Lopez-Paz, D.: Interpolation consistency training for semi-supervised learning. arXiv preprint arXiv:1903.03825 (2019)
28. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in neural information processing systems. pp. 3320–3328 (2014)
29. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
30. Zou, H., Zhou, Y., Yang, J., Liu, H., Das, H.P., Spanos, C.J.: Consensus adversarial domain adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 5997–6004 (2019)
31. Zou, Y., Yu, Z., Liu, X., Kumar, B., Wang, J.: Confidence regularized self-training. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5982–5991 (2019)