Margin-Based Adversarial Joint Alignment Domain Adaptation

Yukun Zuo, Hantao Yao, *Member, IEEE*, Liansheng Zhuang, *Member, IEEE*, and Changsheng Xu, *Fellow, IEEE*

***Abstract*— Domain adaptation aims to transfer the knowledge learned from a labeled source domain to an unlabeled target domain, which has different data distribution with the source domain. Most of the existing methods focus on aligning the data distribution between the source and target domains but ignore the discrimination of the feature space among categories, leading the samples close to the decision boundary to be misclassified easily. To address the above issue, we propose a Margin-based Adversarial Joint Alignment (MAJA) to constrain the feature spaces of source and target domains to be aligned and discriminative. The proposed MAJA consists of two components: joint alignment module and margin-based generative module. The joint alignment module is proposed to align the source and target feature spaces by considering the joint distribution of features and labels. Therefore, the embedding features and the corresponding labels treated as pair data are applied for domain alignment. Furthermore, the margin-based generative module is proposed to boost the discrimination of the feature space, *i.e.,* make all samples as far away from the decision boundary as possible. The margin-based generative module first employs the Generative Adversarial Networks (GAN) to generate a lot of fake images for each category, then applies the adversarial learning to enlarge and reduce the category margin for the true images and generated fake images, respectively. The evaluations on three benchmarks, *e.g.,* small image datasets, VisDA-2017, and Office-31, verify the effectiveness of the proposed method.**

***Index Terms*— Domain adaptation, joint alignment module, margin-based generative module.**

Manuscript received December 29, 2020; revised March 19, 2021; accepted May 7, 2021. Date of publication May 19, 2021; date of current version April 5, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0102205; in part by the National Natural Science Foundation of China under Grant 61902399, Grant 61721004, Grant U1836220, Grant U1705262, Grant 61832002, Grant 61720106006, Grant U20B2070, and Grant 61976199; in part by the Beijing Natural Science Foundation under Grant L201001; and in part by the Key Research Program of Frontier Sciences, Chinese Academy of Sciences (CAS), under Grant QYZDJSSW-JSC039. This article was recommended by Associate Editor J. Han. *(Corresponding author: Changsheng Xu.)*

Yukun Zuo and Liansheng Zhuang are with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China (e-mail: zykpy@mail.ustc.edu.cn; lszhuang@ustc.edu.cn).

Hantao Yao is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: hantao.yao@nlpr.ia.ac.cn).

Changsheng Xu is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: csxu@nlpr.ia.ac.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2021.3081729.

Digital Object Identifier 10.1109/TCSVT.2021.3081729

I. Introduction

In THE past ten years, considerable attention has been paid to deep learning due to its tremendous successes in various areas, such as image segmentation, object detection, and image classification. However, training the deep model requires a large amount of annotated labeled data, which are difficult to obtain and would limit the generability of the proposed methods. Recently, domain adaptation, which aims to learn a well-performing model from a source data distribution and apply it to a different target data distribution, has attracted much attention. Since the domain shift between source and target domains exists, the model trained on the labeled source dataset cannot work well on the unlabeled target dataset. Based on the assumption that the source and target domains contain the same categories, the goal of domain adaptation is to transfer the knowledge from the source domain to the target domain with a small generalization error in the target domain.

Recently, a lot of methods have been proposed for domain adpation [1]–[7]. Most of these methods focus on reducing the domain shift by aligning the visual distributions between source and target domains, *e.g.,* statistical-based methods [8]–[10], adversarial learning-based methods [11]–[13], and reconstruction-based methods [14], [15]. Although the above methods are effective, they still exist two disadvantages for domain adaptation. Firstly, they only apply the marginal feature distribution to align the source domain and the target domain, and ignore the category information. Since the source and target domains share the common categories in domain adaptation, the category information can also be used for aligning two domains. Consequently, using the joint distribution of features and categories to align the source and target domains can be more effective than merely using marginal features distribution. Secondly, existing methods focus on aligning the data distribution between the source and target domains, and ignore the discrimination of the feature space among categories, which leads to the samples close to the decision boundary to be misclassified easily, as shown in Figure 1(a). Therefore, considering the joint distribution of features and categories and enlarging the feature discrimination are two crucial factors for domain adaptation.

By considering the issues mentioned above, we propose a novel Margin-based Adversarial Joint Alignment (MAJA) approach, as shown in Figure 2. The MAJA consists of two components: joint alignment module and margin-based

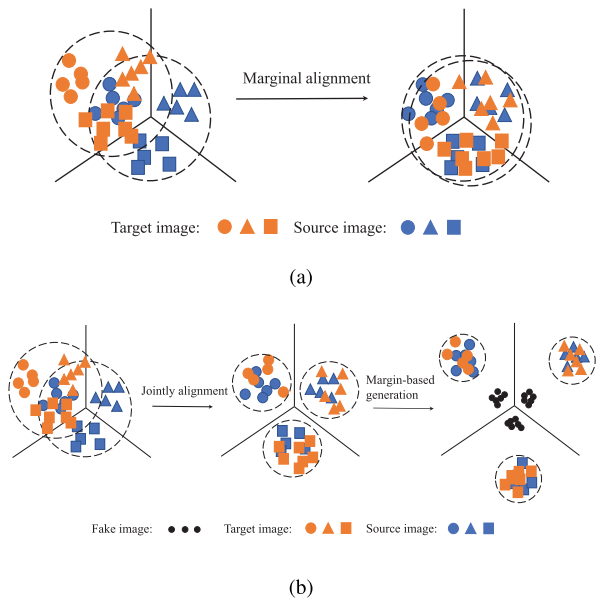


Fig. 1. (a) The previous methods only align the marginal distribution between the source and target domains. (b) The joint alignment module aligns the source and target domains, and the margin-based generation module enlarges the discrepancy among categories.

generative module. The joint alignment module is proposed to align the source and target feature spaces by considering the joint distribution of features and categories. Since the source and target domains share the same categories, using the additional category information can further reduce the domain shift, as shown in Figure 1(b). Based on the fact that the generated fake images are different from the true images in the source and target domains, using the fake images can help to adjust the decision boundary via adversarial learning. Therefore, using the margin-based generative module can enlarge the discrepancy among categories, as shown in Figure 1(b).

The Margin-based Adversarial Joint Alignment (MAJA) is implemented by adding the proposed joint alignment module and margin-based generative module upon the existing domain adaptation methods, *e.g.,* self-ensembling [16], which can be treated as a robust baseline model for domain adaptation. To consider the joint distribution of features and categories, we treat the features and labels as pair data. Given the pair data, the joint alignment module utilizes the minimax game to update the feature extractor $\mathcal{F}$ and domain classifier $\mathcal{D}_1$ for aligning the source and target domains, as shown in Figure 2. For the margin-based generative module, we apply the generator $\mathcal{G}$ to generate the realistic and large category margin fake images that cannot be distinguished by the discriminator $\mathcal{D}_2$ and student network $\mathcal{S}$. Since the fake images are different from the source images, the student network $\mathcal{S}$ is optimized by maximizing and minimizing the category margin for true images and fake images, respectively. The category margin is defined as the difference between the predicted probability of ground-truth class and the largest probability in prediction vector except the ground-truth class. The student network $\mathcal{S}$ will continue to learn more discriminative features of the source images in order to distinguish the fake images from the real source images. By jointly optimizing the joint alignment module and margin-based generative module, the feature extractor can align the features for source and target domains and enhance the feature discrimination.

The main contributions can be summarized as follows:

- We demonstrate that considering the joint distribution of features and categories is an effective way to align the source and target domains than merely using the distribution of features.
- We prove that using the generated fake images to enlarge the discrepancy among categories can boost the performance of domain adaptation.
- The evaluations on three benchmarks, *e.g.,* small image datasets, VisDA-2017 [17], and Office-31 [18], verify the effectiveness of our proposed model.

II. Related Work

In this section, we give a brief review of the related work with our model in domain adaptation, semi-supervised learning and discrimiantive feature learning.

A. Domain Adaptation

Recently, a lot of methods have been proposed for domain adaptation, which can be divided into three groups: statistical-based methods, adversarial learning methods and reconstruction-based methods.

1) Statistical-Based Methods: Since the data distributions of the source domain and target domain are different, calculating statistics of feature distribution can be used to minimize the domain discrepancy. For example, some methods [9], [19]–[22] utilize the Maximum Mean Discrepancy (MMD) to align the high-dimensional features in the source and target domains. DAN [23] proposes a multiple kernel variant of MMD for generalizing deep convolutional neural network to the domain adaptation scenario. JAN [9] aligns the joint distributions of multiple domain-specific layers across domains based on a Joint Maximum Mean Discrepancy (JMMD) criterion. Deep CORAL [24] aligns the source and target domains based on the second-order statistics. Furthermore, the higher-order moment is proposed to align the source and target domains with Central Moment Discrepancy (CMD) [25]. Different from the above methods, RTN [26] applies a residual function to model the domain shift, and Maximum Classifier Discrepancy (MCD) [27] uses the difference of two separate classifiers to embed the domain invariant features.

2) Adversarial Learning Methods: With the great success of Generative Adversarial Networks (GAN) [28] in image generation, adversarial learning [29] has been applied in domain adaption to align the source and target domains. DANN [11] utilizes the minimax game between feature extractor and domain classifier to infer the domain invariant feature. Domain classifier attempts to distinguish the source and target features, and feature extractor attempts to confuse domain classifiers by generating the domain-invariant features. MSTN [12] considers the category information and aligns the centroids of each category between the source and target domains.

DMRL [30] jointly conducts category and domain mixup regularizations on pixel level to improve the effectiveness of models. Recently, the consistency-based methods [31], [32] have achieved good performance and apply cluster assumption in domain adaptation to learn more stable and discriminative features. Batch Spectral Penalization (BSP) [33] considers the relation between transferability and discriminability by the largest singular value of batch features. MDD [13] extends the $\mathcal{H}\Delta\mathcal{H}$ distance to margin disparity discrepancy, which can be transformed into an adversarial learning algorithm for domain adaptation.

3) Reconstruction-Based Methods: Image reconstruction has been widely used to encoder useful information in unsupervised learning. Therefore, it is natural to apply unsupervised reconstruction for domain adaptation. DRCN [14] obtains discriminative and transferable features via supervised classification of labeled source data and unsupervised reconstruction of unlabeled target data. DSN [15] separates feature space into domain-private subspace and domain-share subspace to keep the individual characteristics of each domain, and reconstructs the input sample by using both the private and share representations. CyCADA [34] proposes Cycle-Consistent Adversarial Domain Adaptation to adapt representations at both the pixel-level and feature-level while enforcing semantic consistency.

B. Semi-Supervised Learning

We next introduce two types of methods widely used in semi-supervised learning, such as GANs for semi-supervised learning and Self-ensembling for semi-supervised learning.

1) GANs: With the success of generative adversarial networks (GANs) in image generation, GAN is also used in semi-supervised learning. For example, Feature Matching [35] replaces the binary discriminator with $(K + 1)$-class classifier, where K is the number of categories. The (K + 1)-class classifier classifies the labeled images into the first K source classes, and also classifies the unlabeled images as any of the first K classes. Since the generated fake images do not belong to any source classes, they should be classified as the $(K + 1)$-th class. By generating images located in low-density areas with a "bad" GAN, the method [36] can achieve better generalization by pushing decision boundary through these regions. MarginGAN [37] uses the adversarial learning of margin to generate "bad" images to increase the tolerance of incorrect pseudo labels.

2) Self-Ensembling: Self-ensembling has achieved great success in semi-supervised learning. Temporal Ensembling [38] maintains an exponential moving average of label predictions of each training sample and makes subsequent predictions consistent with the average. Instead of averaging label predictions, Mean Teacher [39] keeps the weights in teacher network being an exponential moving average of the weights in the student network, and constrains the student network to have the consistent outputs with the teacher network under different perturbations. SE [16] integrates the mean-teacher model to domain adaptation and achieves good performance.

C. Discriminative Feature Learning

Some methods [20], [40]–[43] pay efforts to learn more discriminative features for representation learning. CAN [20] proposes Contrastive Domain Discrepancy to measure the difference between conditional data distributions across domains to obtain discriminative target features for domain adaptation. DML [40] utilizes two metric learning stages with different objectives for feature learning. D-CNN [41] imposes a metric learning regularization term on the CNN features to enhance the discrimination of the proposed model. RIFD-CNN [42] adds a rotation-invariant and Fisher discrimination regularizer to achieve rotation-invariance, small within-class scatter and large between-class separation for object detection. CAT [43] proposes Cluster Alignment with a Teacher to incorporate the discriminative clustering structures in both domains.

D. Discussion With Previous Work

Although the previous methods [9], [11], [31] have achieved great performance in domain adaptation, they only consider the marginal distribution between the source domain and the target domain. Different from the previous methods, the proposed Margin-based Adversarial Joint Alignment (MAJA) approach considers other two major additional constraints: joint alignment module and margin-based generative module. Firstly, the joint distribution of features and categories is considered in joint alignment module to align the source and target domains. Secondly, the margin-based generative module can enlarge the discrepancy among categories via adversarial learning between the true images and the generated fake images.

III. Methodology

Domain adaptation aims to classify images belonging to the target domain with the help of the labeled source images. Note that the target domain has the same categories as the source domain, but has a different data distribution. Formally, defining the datasets as $\mathcal{D} = \{\mathcal{D}_s, \mathcal{D}_t\}$, where $\mathcal{D}_s = \{\mathcal{X}_s, \mathcal{Y}_s\}$ is the source dataset, and $\mathcal{D}_t = \{\mathcal{X}_t, \mathcal{Y}_t\}$ is the target dataset. $\mathcal{X}$ and $\mathcal{Y}$ denote the images and corresponding labels, respectively. Based on the assumption that the source categories and target categories are the same, domain adaptation is to infer the category label $\mathcal{Y}_t$ for the image $\mathcal{X}_t$ by making full use of the source dataset $\mathcal{D}_s$.

In this work, we propose a novel Margin-based Adversarial Joint Alignment (MAJA) method to align the source and target domains. As shown in Figure 2, the proposed MAJA consists of two components: joint alignment module and margin-based generative module. MAJA is implemented by adding the proposed joint alignment module and margin-based generative module upon the existing domain adaptation methods, *i.e,* self-ensembling. The self-ensembling model, which can be treated as the baseline method for domain adaptation, is applied to boost the representations by constructing the consistency constraints for unlabeled target images. Based on the self-ensembling module, the joint alignment module is proposed to align the source and target domains by considering the joint distribution of features and categories. Furthermore, the Margin-based generative module is applied to enlarge

the feature discrepancy among categories with the help of generated fake images. In the following, we give a detailed description of each component.

A. Self-Ensembling

Recently, the self-ensembling methods have been obtained powerful performance in semi-supervised learning [35], [36], and also work well in domain adaptation [37]. Therefore, we treat the self-ensembling model as the baseline domain adaptation method in this work. The self-ensembling model consists of two types of networks: student network $\mathcal{S}$ and teacher network $\mathcal{T}$. The network architecture of the teacher network is the same as that of the student network. During training, self-ensembling only updates the weights of the student network, and the weights of the teacher network are updated by the exponential moving average of the weights in the student network. Assuming $\theta_s(k)$ and $\theta_t(k)$ are the weights of the student network and teacher network at step k, respectively. The updating of the weights of the teacher network can be formulated as follows,

$$\theta_t(k) = \mu\theta_t(k-1) + (1-\mu)\theta_s(k), \tag{1}$$

where μ is a smoothing coefficient hyperparameter.

Given the labeled source dataset $\mathcal{D}_s$, self-ensembling optimizes the student network $\mathcal{S}$ by minimizing the supervised classification loss $\mathcal{L}_{cls}$:

$$\mathcal{L}_{cls} = -\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_s}[y^\top \log \mathcal{S}(x)], \tag{2}$$

where x and y are the source image and the corresponding label.

The unlabeled target images and their augmentations are fed into the student and teacher networks to extract the feature descriptions, respectively. Self-ensembling further constructs a constraint that the outputs of unlabeled target sample x_t and its augmentation $\bar{x}_t$ should be consistent. Therefore, we minimize the consistency constraints by:

$$\mathcal{L}_{con} = \mathop{\mathbb{E}}_{x\sim\mathcal{X}_t}[||\mathcal{S}(x) - \mathcal{T}(\bar{x})||^2], \tag{3}$$

where $\mathcal{S}(x)$ and $\mathcal{T}(\bar{x})$ denote the outputs of student network and teacher networks, respectively.

B. Joint Alignment Module

The joint alignment module aims to align the source and target domains by considering the joint distribution of features and categories. As a consequence, we need to generate the feature space and category space for each image. Therefore, we divide the student network and teacher network into two sub-components: feature extractor $\mathcal{F}$ and classifier network $\mathcal{C}$. Given the images x_s and x_t, we apply the feature extractor $\mathcal{F}$ to extract the source and target features $\mathcal{F}(x_s)$ and $\mathcal{F}(x_t)$, which are further fed into the classifier $\mathcal{C}$ to predict its category. The obtained category probabilities are denoted as $\mathcal{C}(\mathcal{F}(x_s))$ and $\mathcal{C}(\mathcal{F}(x_t))$. Since the ground-truth label for target images x_t is unavailable, we can generate its pseudo label $\hat{y}_t$ by choosing the class with the highest probability. The goal of joint alignment module is to align the source domain and target domain with the joint distribution $\mathcal{P}(\mathcal{F}(x), y)$ between features $\mathcal{F}(x)$ and category y. For the labeled source sample x_s, we utilize the corresponding label y_s to construct the joint distribution $\mathcal{P}_s(\mathcal{F}(x_s), y_s)$. For the unlabeled target sample x_t, we utilize the pseudo label $\hat{y}_t$ to obtain the joint distribution $\mathcal{P}_t(\mathcal{F}(x_t), \hat{y}_t)$.

Given the source dataset $\mathcal{D}_s$ and target dataset $\mathcal{D}_t$, we firstly generate the joint distribution sample sets $\mathcal{A}_s$ and $\mathcal{A}_t$ for source and target domains, respectively. The source joint distribution sample set $\mathcal{A}_s$ is defined as $\mathcal{A}_s = \{(\mathcal{F}(x_s^1), y_s^1), (\mathcal{F}(x_s^2), y_s^2), \ldots, (\mathcal{F}(x_s^{n_s}), y_s^{n_s})\}$, where n_s is the number of source images, and y_s^i is the label. Similarly, the target joint distribution sample set $\mathcal{A}_t = \{(\mathcal{F}(x_t^1), \hat{y}_t^1), (\mathcal{F}(x_t^2), \hat{y}_t^2), \ldots, (\mathcal{F}(x_t^{n_t}), \hat{y}_t^{n_t})\}$, where n_t is the number of target images, and $\hat{y}_t^i$ is the pseudo label. After obtaining the $\mathcal{A}_s$ and $\mathcal{A}_t$, the domain classifer $\mathcal{D}_1$ is trained to distinguish these two joint distributions. The objective of the domain classifier is formulated as Eq. (4):

$$\mathcal{L}_{\mathcal{D}_1} = -\{\mathop{\mathbb{E}}_{\mathbf{j_s}\sim\mathcal{A}_s}[\log(\mathcal{D}_1(\mathbf{j_s})] + \mathop{\mathbb{E}}_{\mathbf{j_t}\sim\mathcal{A}_t}[\log(1-\mathcal{D}_1(\mathbf{j_t})]\}, \tag{4}$$

where $\mathbf{j_s}$ and $\mathbf{j_t}$ represent the joint distribution samples $(\mathcal{F}(x), y)$ in $\mathcal{A}_s$ and $\mathcal{A}_t$, respectively.

By fixing the trained domain classifier $\mathcal{D}_1$, we next optimize the feature extractor $\mathcal{F}$ to make the domain classifier $\mathcal{D}_1$ not be able to distinguish the source joint distribution $\mathcal{P}_s(\mathcal{F}(x_s), y_s)$ and target joint distribution $\mathcal{P}_t(\mathcal{F}(x_t), \hat{y}_t)$, which can align the visual spaces for source and target domains. The objective can be formulated as follows,

$$\mathcal{L}_{\mathcal{F}_{JA}} = -\mathop{\mathbb{E}}_{(\mathcal{F}(x_t),\hat{y}_t)\sim\mathcal{A}_t}[\log(\mathcal{D}_1((\mathcal{F}(x_t), \hat{y}_t))]. \tag{5}$$

By combining Eq. (4) and Eq. (5), the joint alignment module optimizes the feature extractor $\mathcal{F}$ and domain classifier $\mathcal{D}_1$ with a minmax game,

$$\begin{aligned}\min_{\mathcal{F}}\max_{\mathcal{D}_1}\mathcal{L}_{JA} = &\mathop{\mathbb{E}}_{\mathbf{j_s}\sim\mathcal{A}_s}[\log(\mathcal{D}_1(\mathbf{j_s})] \\ &+ \mathop{\mathbb{E}}_{(\mathcal{F}(x_t),\hat{y}_t)\sim\mathcal{A}_t}[\log(1-\mathcal{D}_1((\mathcal{F}(x_t), \hat{y}_t))].\end{aligned} \tag{6}$$

Finally, when the target feature distribution is similar to the source feature distribution, the target pseudo labels obtained by the classifier would be correct, which can further make the domain classifier not distinguish the source joint distribution and the target joint distribution.

C. Margin-Based Generative Module

Although existing methods can effectively align the source feature space and target feature space, the feature spaces of different categories are easily confused. In this work, we propose a margin-based generative module to enlarge the discrepancy among categories. In the following, we first define *Category Margin*, and then introduce the margin-based generative module.

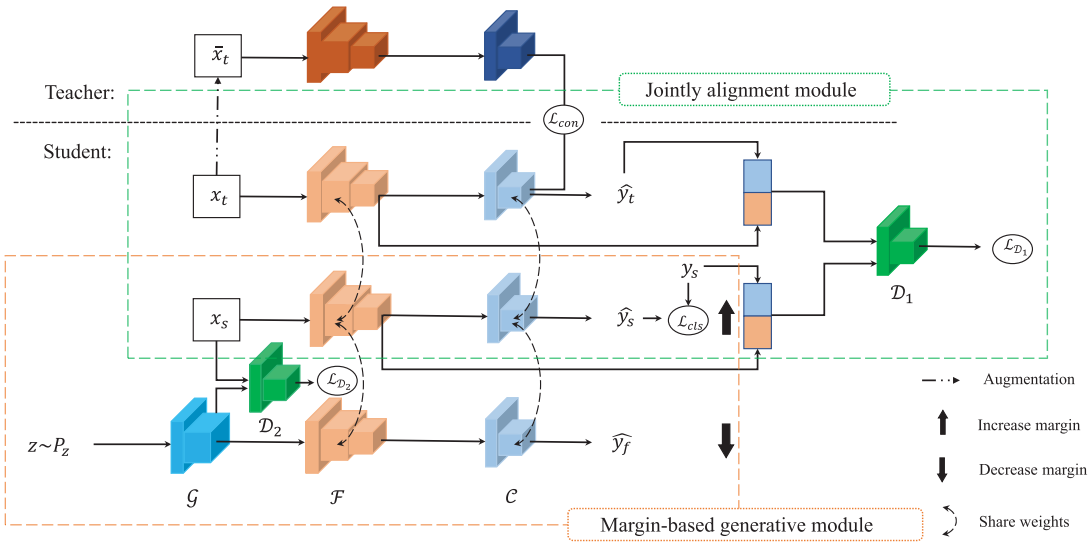


Fig. 2. An overview of our Margin-based Adversarial Joint Alignment method. x_s, x_t, and $\bar{x}_t$ represent source sample, target sample, and augmented target sample, respectively. MAJA consists of two modules: joint alignment module and margin-based generative module. For the joint alignment module, we firstly obtain the pseudo label $\hat{y}_t$ for target sample x_t through the student network $\mathcal{S}$. Next, the source joint distribution sample $(\mathcal{F}(x_s), y_s)$ and the target joint distribution sample $(\mathcal{F}(x_t), \hat{y}_t)$ are fed into the domain classifier $\mathcal{D}_1$ to distinguish two domains. Furthermore, the feature extractor $\mathcal{F}$ is optimized to fool the domain classifier $\mathcal{D}_1$. For the margin-based generative module, the generator $\mathcal{G}$ is applied to generate the fake images that cannot be distinguished by the discriminator $\mathcal{D}_2$ and the student network $\mathcal{S}$. By fixing the trained generator $\mathcal{G}$, the discriminator $\mathcal{D}_2$ is updated to distinguish the fake images and the source images. Finally, the student network $\mathcal{S}$ is optimized to increase the category margin of the source images and decrease the category margin of the fake images. The teacher network $\mathcal{T}$ is used to supervise the student network for the unlabeled target sample x_t through different data augmentations.

1) Category Margin: Given a sample (x, y), where x and y are the image and label, the *category margin* is defined as the difference between the y-th predicted probability and the largest predicted probability except the y-th class. Formally, the *category margin* is defined as:

$$CM(x, y) = \mathbf{p}_y(x) - \max_{m \neq y} \mathbf{p}_m(x), \tag{7}$$

where $\mathbf{p}_y(x)$ denotes the predicted probability of the y-th class for the image x, and $CM(x, y)$ is the corresponding category margin. The larger margin, the more confident that the sample (x, y) is correctly classified. The margin close to 0 indicates that the predicted probability for x is uncertain.

We can increase the category margin by minimizing the cross entropy of the prediction:

$$\mathcal{L}_{en} = -\mathbf{y}^\top \log \mathbf{p}(x). \tag{8}$$

Furthermore, using the inverse cross entropy of the prediction can reduce the category margin,

$$\mathcal{L}_{ien} = -\mathbf{y}^\top \log(\mathbf{1} - \mathbf{p}(x)), \tag{9}$$

where $\mathbf{y}$ represents one-hot vector of y and $\mathbf{p}$(x) indicates the prediction probability vector of x.

2) Category Margin in Generative Module: The Margin-based generative module is divided into three components: generator $\mathcal{G}$, discriminator $\mathcal{D}_2$, and student network $\mathcal{S}$. The generator $\mathcal{G}$ is used to generate realistic and large category margin fake images that help to adjust the decision boundary and enlarge the discrepancy among categories. The discriminator $\mathcal{D}_2$ aims to distinguish the source images and fake images. Optimizing the student network $\mathcal{S}$ aims to increase the margin for the source images and decrease the margin for fake images. By performing the adversarial learning among generator $\mathcal{G}$, discriminator $\mathcal{D}_2$, and network S, the generated source features are discriminative enough.

For the discriminator $\mathcal{D}_2$, it aims to distinguish source images x_s and fake images $\hat{x}_s$ generated by the generator $\mathcal{G}$. By assigning the labels for source images and fake images as 1 and 0, the discriminator $\mathcal{D}_2$ can be optimized as follows,

$$\mathcal{L}_{\mathcal{D}_2} = -\{ \mathop{\mathbb{E}}_{x \sim \mathcal{X}_s} [\log(\mathcal{D}_2(x))] + \mathop{\mathbb{E}}_{z \sim \mathbf{P}_z} [\log(1 - \mathcal{D}_2(\mathcal{G}(z)))]\}, \tag{10}$$

where $\mathbf{P}_z$ denotes a simple distribution, *e.g.,* normal or uniform distribution.

For the student network $\mathcal{S}$, it is used to increase the margin of source images and decrease the margin of fake images. Since the source images have the ground-truth labels, we minimize the cross-entropy of the prediction probability to increase the margin,

$$\mathcal{L}_{\mathcal{S}_{cls}} = - \mathop{\mathbb{E}}_{(x,y) \sim \mathcal{D}_s} [\mathbf{y}^\top \log \mathcal{S}(x)]. \tag{11}$$

For the fake images generated by the generator $\mathcal{G}$, there is no ground-truth labels. We thus take the class having the highest probability as the pseudo label, and further reduce the margin by minimizing the inverse cross entropy between prediction probability and the pseudo label,

$$\mathcal{L}_{\mathcal{S}_{ien}} = - \mathop{\mathbb{E}}_{z \sim \mathbf{P}_z} [\hat{\mathbf{y}}_f^\top \log(1 - \mathcal{S}(\mathcal{G}(z))], \tag{12}$$

where $\hat{\mathbf{y}}_f$ is the pseudo label one-hot vector.

For the generator $\mathcal{G}$, it fools the discriminator $\mathcal{D}_2$ and the student network $\mathcal{S}$ to generate the realistic fake images. Given the discriminator $\mathcal{D}_2$, the generator $\mathcal{G}$ randomly samples from the distribution $\mathbf{P}_z$ to generate fake images that cannot be distinguished by the discriminator $\mathcal{D}_2$. Therefore, the objective is:

$$\mathcal{L}_{\mathcal{G}_{GAN}} = - \mathop{\mathbb{E}}_{z\sim\mathbf{P}_z}[\log(\mathcal{D}_2(\mathcal{G}(z)))]. \tag{13}$$

By fixing the student network $\mathcal{S}$, we optimize the generator $\mathcal{G}$ to constrain that the probability output of the student network has large margin,

$$\mathcal{L}_{\mathcal{G}_{en}} = - \mathop{\mathbb{E}}_{z\sim\mathbf{P}_z}[\hat{\mathbf{y}}_f^\top \log \mathcal{S}(\mathcal{G}(z)]. \tag{14}$$

By combining the above objective functions, the margin-based generative module optimizes Eq. (15) via a minimax problem,

$$\begin{aligned}\min_{\mathcal{G}} \max_{\mathcal{D}_2,\mathcal{S}} \mathcal{L}_{MG} &= \mathop{\mathbb{E}}_{x\sim\mathcal{X}_s}[\log(\mathcal{D}_2(x))] + \mathop{\mathbb{E}}_{z\sim\mathbf{P}_z}[\log(1-\mathcal{D}_2(\mathcal{G}(z)))] \\ &+ \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_s}[\mathbf{y}^\top \log \mathcal{S}(x)] + \mathop{\mathbb{E}}_{z\sim\mathbf{P}_z}[\hat{\mathbf{y}}_f^\top \log(1-\mathcal{S}(\mathcal{G}(z)].\end{aligned} \tag{15}$$

D. Training Procedure

The final model is the combination of the joint alignment module and the margin-based generative module. We iteratively update each component of the final model, *e.g.,* feature extractor $\mathcal{F}$, classifier $\mathcal{C}$, generator $\mathcal{G}$, domain classifier $\mathcal{D}_1$, and discriminator $\mathcal{D}_2$.

We firstly update the feature extractor $\mathcal{F}$ and the classifier $\mathcal{C}$ by minimizing the following equation,

$$\begin{aligned}\min_{\mathcal{F},\mathcal{C}} \mathcal{L}_{\mathcal{F}\&\mathcal{C}} &= \mathop{\mathbb{E}}_{x\sim\mathcal{X}_t}[||\mathcal{C}(\mathcal{F}(x)) - \mathcal{T}(\bar{x})||^2] - \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_s}[\mathbf{y}^\top \log \mathcal{C}(\mathcal{F}(x))] \\ &- \mathop{\mathbb{E}}_{(\mathcal{F}(x_t),\hat{y}_t)\sim\mathcal{A}_t}[\log(\mathcal{D}_1((\mathcal{F}(x_t),\hat{y}_t)))] \\ &- \mathop{\mathbb{E}}_{z\sim\mathbf{P}_z}[\hat{\mathbf{y}}_f^\top \log(1-\mathcal{C}(\mathcal{F}(\mathcal{G}(z))))].\end{aligned} \tag{16}$$

Next, we update the generator $\mathcal{G}$ to fool the discriminator $\mathcal{D}_2$ and the student network $\mathcal{S}$,

$$\min_{\mathcal{G}} \mathcal{L}_{\mathcal{G}} = -\{\mathop{\mathbb{E}}_{z\sim\mathbf{P}_z}[\log(\mathcal{D}_2(\mathcal{G}(z)))] + \mathop{\mathbb{E}}_{z\sim\mathbf{P}_z}[\hat{\mathbf{y}}_f^\top \log \mathcal{S}(\mathcal{G}(z))]\}. \tag{17}$$

Since the domain classifier $\mathcal{D}_1$ is used to distinguish the joint distributions in the source domain and target domain, which can be optimized with Eq. (18),

$$\min_{\mathcal{D}_1} \mathcal{L}_{\mathcal{D}_1} = -\{\mathop{\mathbb{E}}_{\mathbf{j_s}\sim\mathcal{A}_s}[\log(\mathcal{D}_1(\mathbf{j_s}))] + \mathop{\mathbb{E}}_{\mathbf{j_t}\sim\mathcal{A}_t}[\log(1-\mathcal{D}_1(\mathbf{j_t}))]\}. \tag{18}$$

Finally, we optimize the discriminator $\mathcal{D}_2$ to distinguish source images and fake images generated by the generator $\mathcal{G}$ with Eq. (19),

$$\min_{\mathcal{D}_2} \mathcal{L}_{\mathcal{D}_2} = -\{\mathop{\mathbb{E}}_{x\sim\mathcal{X}_s}[\log(\mathcal{D}_2(x))] + \mathop{\mathbb{E}}_{z\sim\mathbf{P}_z}[\log(1-\mathcal{D}_2(\mathcal{G}(z)))]\}. \tag{19}$$

By iteratively performing the above update steps, we can obtain the final feature extractor $\mathcal{F}$ and classifier $\mathcal{C}$ that can be applied to classify the target images.

IV. Experiments

A. Dataset

We conduct the evaluations on three benchmarks to demonstrate the effectiveness of our proposed method, *e.g.,* Small image datasets, VisDA-2017, and Office-31.

1) Small Image Datasets: For Small image datasets, we choose SVHN [52], MNIST [53], CIFAR-10 [54], STL [55], Syn Digits [2], GTSRB [56] and Syn-Signs [57] for evaluations. SVHN is the Street View House Numbers Dataset, whose images are collected from the house number photographed in Google Street View. MNIST is a well-known greyscale hand-written digit dataset created by the National Institute of Standards and Technology (NIST). Unlike the previous digital dataset, CIFAR-10 is a 10-class dataset of real objects, including airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. We remove the "frog" class in CIFAR-10 for unsupervised domain adaptation. STL contains the same categories as CIFAR-10 except for "Monkey". Syn-Digits is a synthetic counterpart of the SVHN dataset, and GTSRB is a German traffic sign dataset containing 43 traffic signals. Syn-Signs is a synthetic image dataset based on GTSRB.

2) VisDA-2017: VisDA-2017 dataset is a large-scale cross-domain dataset, which contains 280K images from twelve categories. We use the training set as the source domain and the validation set as the target domain. The source domain including 152,397 images is synthetic 2D renderings of 3D models generated from different angles and with different lighting conditions. The target domain consisting of 55,388 images is collected from COCO [58].

3) Office-31: Office-31 is a well-known domain adaptation dataset which contains 31 categories in three domains: Amazon, Webcam, and DSLR. The Office-31 dataset contains 4,110 images, of which Amazon (A) domain contains 2,817 images, Webcan (W) domain contains 795 images, and DSLR contains 498 digit SLR pictures.

B. Implementation Detail

For small image datasets, we utilize a 12-block residual network [59] with Shake-Shake regularization [60] as the backbone network similar to MarginGAN, and replace the last fully-connected (FC) layer with the task-specific FC layer as classifier $\mathcal{C}$. The generator $\mathcal{G}$ and discriminator $\mathcal{D}_2$ are derived from the infoGAN [61]. For domain classifier $\mathcal{D}_1$, we use three fully connected layers: (fea_dim + n_class) $\rightarrow$ 500 $\rightarrow$ 500 $\rightarrow$ 1, where fea_dim and n_class represent the dimensions

TABLE I

COMPARISON WITH THE EXISTING METHODS ON SMALL IMAGE DATSETS. THE BEST PERFORMANCE IS HIGHLIGHTED IN **BOLD**

Methods	SVHN ↓ MNIST	MNIST ↓ SVHN	CIFAR ↓ STL	STL ↓ CIFAR	Syn Digits ↓ SVHN	Syn Signs ↓ GTSTB
RevGrad [2]	73.91	35.67	66.12	56.91	91.09	88.65
DCRN [14]	81.97	40.05	66.37	58.65	-	-
G2A [44]	84.70	36.4	-	-	-	-
ADDA [45]	76.00	-	-	-	-	-
ATT [46]	86.20	52.8	-	-	93.1	96.2
SBADA-GAN [47]	76.14	61.08	-	-	-	-
ADA [48]	97.6	-	-	-	91.86	97.66
SE [16]	**99.55**	92.64*	83.92	76.39	**96.66**	98.63
VADA [31]	94.5	73.3	78.3	71.4	-	-
SWD [49]	98.9	-	-	-	-	98.6
MMEN [50]	98.8	-	-	-	-	-
SEMA(SE + MAJA)	99.29	**96.49***	**86.63**	**78.72**	96.29	**98.65**

*presents using class balance loss and specific intensity augmentation.

TABLE II

COMPARISON WITH THE EXISTING METHODS ON VISDA-2017. THE BEST PERFORMANCE IS HIGHLIGHTED IN **BOLD**

Methods	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	mean
RevGrad [2]	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
DAN [23]	68.1	15.4	76.5	87.0	71.1	48.9	82.3	51.5	88.7	33.2	88.9	42.2	62.8
JAN [9]	75.7	18.7	82.3	86.3	70.2	56.9	80.5	53.8	92.5	32.2	84.5	54.5	65.7
MCD [27]	87.0	60.9	83.7	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
ADR[51]	87.8	79.5	83.7	65.3	92.3	61.8	88.9	73.2	87.8	60.0	85.5	32.3	74.8
SE [16]	95.9	87.4	85.2	58.6	96.2	95.7	90.6	80.0	94.8	90.8	88.4	47.9	84.3
SEMA(SE + MAJA)	95.8	85.5	83.0	69.2	94.7	96.4	91.2	80.3	93.2	93.3	87.1	45.7	**84.6**

of feature and label, respectively. Batch Normalization is inserted in fully connected layers. For SVHN ↔ MNIST, CIFAR-10 ↔ STL, Syn-Digits → SVHN tasks, we resize the images to 32×32. For Syn-Signs → GTSRB task, we resize images to 40×40. We utilize the same policy "CT+TF" as described in SE [16], where "CT" represents confidence thresholding and "TF" means using translation and horizontal flip augmentation. We adopt SGD optimizer with learning rate of 0.05, momentum of 0.9, weight_decay of 0.0002, and nesterov of True for feature extractor $\mathcal{F}$ and classifier $\mathcal{C}$. We use Adam optimizer with learning rate of 0.0002, beta1 of 0.5 and beta2 of 0.999 for generator $\mathcal{G}$, domain classifier $\mathcal{D}_1$, and discriminator $\mathcal{D}_2$. We set smoothing coefficient hyper-parameter μ in exponential moving average between student network and teacher network is 0.99.

For VisDA-2017, we use ResNet-101 [59] pre-trained on ImageNet [62] as the backbone network, and replace the last FC layer with the task-specific FC layer as classifier. We resize all images to 160×160. We adopt Adam optimizer with learning rate of 0.0001 for feature extractor $\mathcal{F}$ and classifier $\mathcal{C}$. The optimizers of generator $\mathcal{G}$, domain classifier $\mathcal{D}_1$, and discriminator $\mathcal{D}_2$ are same as in small image datasets. For Office-31, We adopt ResNet-50 pre-trained on ImageNet as the backbone network. Other experiment settings of Office-31 are same as in VisDA-2017.

C. Comparison With Existing Methods

In this section, we make the comparison between the proposed method and existing methods on three benchmarks, and summarize the results in Table I, Table II and Table III.

1) Small Image Datasets: For the small image datasets, the MAJA is implemented based on SE, named SEMA. We compare SEMA with the existing methods, including RevGrad [2], DCRN [14], G2A [44], ADDA [45], ATT [46], SBADA-GAN [47], ADA [48], SE [16], VADA [31], SWD [49] and MMEN [50]. The detailed results are shown in Table I. From Table I, we can find that the proposed SEMA model achieves the best performance except SVHN → MNIST and Syn-Digits → SVHN settings. The reason is that the target images have a small domain gap with the source images and the baseline SE model obtains the high performance, *e.g.,* 99.55% and 96.66% for the setting of SVHN → MNIST and Syn-Digits → SVHN, respectively. Since the fake images influence the parameters of BN layer in backbone network, which could have negative impacts for the predictions of the target samples. Therefore, using the generated fake images may degrade the performance, leading to the SEMA obtain a worse performance than SE. We also observe that the proposed SEMA obtains an obvious improvement upon the challenging settings, *e.g.,* obtaining 2.71% and 2.33% improvement for the CIFAR-10 → STL and STL → CIFAR-10 tasks, respectively.

2) VisDA-2017: Since the images belonging to the Small image datasets have small scales, we further evaluate the SEMA on a large scale VisDA-2017 dataset. We compare SEMA with RevGrad [2], DAN [23], JAN [9], MCD [27], ADR [51], SE [16], and summarize the related results in Table II. Note that the baseline Self-ensembling (SE) wins the first place in the VisDA-2017 competition. By aligning the joint distribution of the source domain and the target domain, and enlarging the discrimination of the feature space, SEMA

TABLE III

COMPARISON WITH THE EXISTING METHODS ON OFFICE-31. THE BEST PERFORMANCE IS HIGHLIGHTED IN **BOLD**. THE "A", "D", AND "W" ARE THE ABBREVIATION OF "AMAZON", "DSLR", AND "WEBCAM", RESPECTIVELY

Methods	A→D	D→A	W→A	D→W	W→D	A→W	mean
GFK [63]	74.5	63.4	61.0	95.0	98.2	72.8	77.5
DAN [23]	78.6	63.6	62.8	97.1	99.6	80.5	80.4
RevGrad [2]	79.7	68.2	67.4	96.9	99.1	82.0	82.2
ADDA [45]	77.8	69.5	68.9	96.2	98.4	86.2	82.9
JAN [9]	85.1	69.2	70.7	96.7	99.7	86.0	84.6
DMRL [30]	93.4	73.0	71.2	**99.0**	**100.0**	90.8	87.9
SE [16]	83.0	69.1	69.0	97.6	98.7	85.1	83.8
CAN [20]	**94.6**	77.2	75.7	97.2	99.4	93.3	89.6
SEMA(SE + MAJA)	83.6	70.4	70.9	98.3	99.4	87.1	85.0
CAMA(CAN + MAJA)	93.9	**77.4**	**75.8**	98.5	99.8	**94.9**	**90.1**

TABLE IV

ABLATION EXPERIMENTS ON OFFICE-31. THE BEST PERFORMANCE IS HIGHLIGHTED IN **BOLD**. THE ABBREVIATION "SE", "JA", AND "MG" REPRESENTS THE SELF-ENSEMBLING MODEL, JOINT ALIGNMENT MODULE, AND MARGIN-BASED GENERATIVE MODULE, RESPECTIVELY. "MAJA" IS THE PROPOSED MODEL."+" DENOTES THE COMBINATION OPERATION

Methods	A→D	D→A	W→A	D→W	W→D	A→W	mean
SE	83.0	69.1	69.0	97.6	98.7	85.1	83.8
SE+JA	**83.9**	**70.4**	69.8	97.7	99.1	85.5	84.4
SE+MG	83.7	68.5	69.4	97.7	**99.5**	86.3	84.2
SEMA(SE + MAJA)	83.6	**70.4**	**70.9**	**98.3**	99.4	**87.1**	**85.0**

obtains a higher performance than Self-ensembling (SE), *e.g.,* improving the performance from 84.3% to 84.6%.

3) Office-31: Office-31 consists of three domains: Amazon (A), Webcam (W), DSLR (D), and contains six domain adaptation tasks: A → D, D → A, W → A, D → W, W → D, and A → W. For Office-31, we also treat the Contrastive Adaptation Network (CAN) as baseline to combine the proposed Margin-based Adversarial Joint Alignment module, named CAMA. Since the results of CAN varies greatly, we reimplement CAN with the released code and obtain the average results of CAN. We compares SEMA and CAMA with existing methods, and summarize the related results in Table III. As shown in Table III, CAMA obtains the best performance and adding the proposed MAJA module can further boost the performance upon the baseline, *e.g.,* SEMA obtains 1.2% improvement compared with SE, and CAMA achieve higher performance than CAN by 0.5%.

Based on the above comparison and analysis, we can conclude that considering the joint distribution of feature and category information can enlarge the discrepancy of feature spaces and boost the performance of domain adaptation.

D. Ablation Study

As discussed above, the proposed model consists of two critical components: joint alignment module and margin-based generative module. We thus evaluate the effect of each component.

1) Effect of Joint Alignment Module: The joint alignment module aims to align the joint distribution of features and categories. We first evaluate the effect of joint alignment module, and summarize the results in Table IV. In Table IV, "SE" represents the self-ensembling model, "JA" indicates the joint alignment module, "SE+JA" means the combination of the self-ensembling model and the joint alignment module. From Table IV, we can observe that considering the joint distribution can further boost the performance, *e.g.,* "SE+JA" model obtains 0.6% improvement upon the "SE" model. The improvement can demonstrate the necessity and effectiveness of joint alignment module for aligning the source and target domains.

In joint alignment module, we claim that using the joint distribution of features and categories is superior to merely using the distribution of features. We thus conduct experiments D→A and W→D to verify the effectiveness of joint distribution. As shown in Table V, "SE+JA" obtains a higher performance than "SE+MA" which represents incorporating marginal alignment module to the self-ensembling model, *e.g.,* improving the performance from 69.3% to 70.4% and 98.9% to 99.1%, respectively. The higher performance shows that aligning joint distribution is superior to align marginal distribution. From Table V, we also observe that the "SE+MA" obtains only 0.2% improvement upon the "SE" model in D→A task. Compared with the slight improvement obtained by "SE+MA", the large improvement for "SE+JA", *e.g.,* 1.3% improvement, can further demonstrate the effectiveness of using the joint distribution for domain adaptation.

2) Effect of Margin-Based Generative Module: Margin-based generative module aims to enlarge the feature discrimination with the help of the generated fake images. We then evaluate the effect of margin-based generative module, and summarize the results in Table IV. "MG" denotes the model with margin-based generative module, "SE+MG" means the combination of self-ensembling model and margin-based generative module. From Table IV, we can see that adding the margin-based generative module can

TABLE V

COMPARISON ON D → A AND W → D. "SE", "MA", AND "JA" REPRESENTS THE SELF-ENSEMBLING MODEL, MARGINAL ALIGNMENT MODULE, AND JOINT ALIGNMENT MODULE, RESPECTIVELY. "+" DENOTES THE COMBINATION OPERATION

Methods	D→A	W→D
SE	69.1	98.7
SE+MA	69.3	98.9
SE+JA	**70.4**	**99.1**

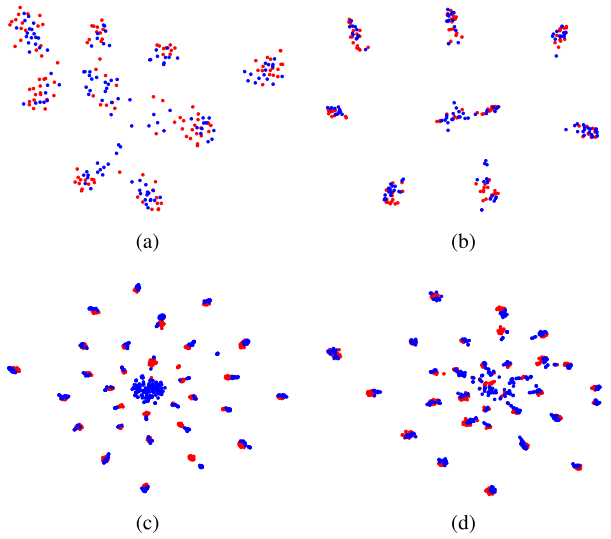


Fig. 3. The visualization of features about SE and SEMA in CIFAR-10→STL(C→S) and Amazon→Webcam(A→W) task. Red and blue dots represent the source features and target features, respectively. We can observe that the proposed SEMA can enlarge the discrepancy among categories and align the source and target features.

boost the performance, *e.g.,* "SE+MG" improves the mean performance from 83.8% to 84.2%. The higher performance proves that using margin-based generative module can obtain more discriminative features in the target domain. Furthermore, combining joint alignment module and margin-based generative module obtains the highest performance compared with merely using one, *e.g.,* SEMA obtains 0.6% and 0.8% improvement upon the "SE+JA" and "SE+MG" models, respectively. Therefore, we can conclude that the joint alignment module and margin-based generative module are two complementary components, and combining them is a reasonable choice for domain adaptation.

3) Visualization: We further illustrate the effectiveness of the proposed method by visualizing the features in CIFAR-10→STL and Amazon→Webcam task. For the visualization, we firstly apply the feature extractor $\mathcal{F}$ to generate the corresponding visual descriptions, and then utilize T-SNE [64] to visualize the feature distributions in Figure 3. From Figure 3, we can observe that using the adversarial learning between the generated images and the source images, and aligning the joint distributions of the source domain and target domain, the target features of different classes aligned with the source features are more discriminative.

Fig. 4. Generated fake images in VisDA-2017 dataset.

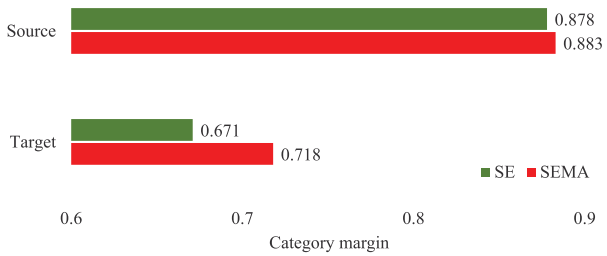


Fig. 5. The mean category margin of the predictions about SE and SEMA in the source and target domain for CIFAR-10→STL task.

One core of the margin-based generative model is that we apply the GAN to generate some fake images used to enhance the feature discrimination. We thus visualize the generated fake images of VisDA-2017 dataset as shown in Figure 4. It can be seen that the generated images are realistic and cannot be distinguished belonging to any category. Therefore, these fake images can be utilized to adjust the decision boundary and constrain the source features of each category to be far away from each other.

4) Category Margin: To demonstrate that the proposed MAJA module can enlarge the category margin of source and target categories, we calculate the mean category margin of the predictions about the model with and without MAJA module in the source and target domain for CIFAR-10 → STL task. From Figure 5 we can see that, the mean category margin of SEMA module is higher than that of SE in both the source and target domains, which proves that using the MAJA module can boost the performance after enlarging the category margin.

V. CONCLUSION

In order to deal with the challenges in domain adaptation, we propose a Margin-based Adversarial Joint Alignment (MAJA) method. We propose a joint alignment module, by considering the joint distribution of feature and categories to align the joint distributions of source and target domains. Furthermore, we use margin-based generative module to boost the discrimination of the feature space by enlarging the dis-

crepancy among categories. The evaluations on three benchmarks prove the effectiveness of our method.

In this work, we only consider the interaction between the generative module and the source domain. In the future, we will integrate the generative module into the target domain to obtain better experimental results.

REFERENCES

[1] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, nos. 1–2, pp. 151–175, May 2010.

[2] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, 2015, pp. 1180–1189.

[3] H. Zhao, R. T. Des Combes, K. Zhang, and G. Gordon, "On learning invariant representations for domain adaptation," in *Proc. ICML*, 2019, pp. 7523–7532.

[4] Y. Zuo, H. Yao, and C. Xu, "Category-level adversarial self-ensembling for domain adaptation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2020, pp. 1–6.

[5] P. Morerio, R. Volpi, R. Ragonesi, and V. Murino, "Generative pseudo-label refinement for unsupervised domain adaptation," 2020, *arXiv:2001.02950*. [Online]. Available: http://arxiv.org/abs/2001.02950

[6] Y. Wang *et al.*, "Domain-specific suppression for adaptive object detection," 2021, *arXiv:2105.03570*. [Online]. Available: https://arxiv.org/abs/2105.03570

[7] Y. Zuo, H. Yao, and C. Xu, "Attention-based multi-source domain adaptation," *IEEE Trans. Image Process.*, vol. 30, pp. 3793–3803, 2021.

[8] C. Cortes, M. Mohri, and A. M. Medina, "Adaptation based on generalized discrepancy," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 1–30, 2019. [Online]. Available: http://jmlr.org/papers/v20/15-192.html

[9] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 2208–2217.

[10] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2058–2065.

[11] Y. Ganin, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, pp. 2030–2096, Jan. 2016.

[12] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5419–5428.

[13] Y. Zhang, T. Liu, M. Long, and M. I. Jordan, "Bridging theory and algorithm for domain adaptation," 2019, *arXiv:1904.05801*. [Online]. Available: http://arxiv.org/abs/1904.05801

[14] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 597–613.

[15] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 343–351.

[16] G. French, M. Mackiewicz, and M. Fisher, "Self-ensembling for visual domain adaptation," in *Proc. ICLR*, Jun. 2018, pp. 1–5.

[17] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, "VisDA: The visual domain adaptation challenge," 2017, *arXiv:1710.06924*. [Online]. Available: http://arxiv.org/abs/1710.06924

[18] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2010, pp. 213–226.

[19] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu, "Visual domain adaptation with manifold embedded distribution alignment," in *Proc. ACM Multimedia Conf. Multimedia Conf.*, 2018, pp. 402–410.

[20] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4893–4902.

[21] L. Zhang *et al.*, "Unsupervised domain adaptation using robust class-wise matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 5, pp. 1339–1349, May 2019.

[22] W. Deng, L. Zheng, Y. Sun, and J. Jiao, "Rethinking triplet loss for domain adaptation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 29–37, Jan. 2021.

[23] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[24] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 443–450.

[25] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (CMD) for domain-invariant representation learning," 2017, *arXiv:1702.08811*. [Online]. Available: http://arxiv.org/abs/1702.08811

[26] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 136–144.

[27] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3723–3732.

[28] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[29] X. Xu, H. He, H. Zhang, Y. Xu, and S. He, "Unsupervised domain adaptation via importance sampling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4688–4699, Dec. 2020.

[30] Y. Wu, D. Inkpen, and A. El-Roby, "Dual mixup regularized learning for adversarial domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 540–555.

[31] X. Guo, W. Chen, and J. Yin, "A simple approach for unsupervised domain adaptation," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 1–8.

[32] A. Kumar *et al.*, "Co-regularized alignment for unsupervised domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9345–9356.

[33] X. Chen, S. Wang, M. Long, and J. Wang, "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation," in *Proc. ICML*, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds. Long Beach, CA, USA: PMLR, Jun. 2019, pp. 1081–1090. [Online]. Available: http://proceedings.mlr.press/v97/chen19i.html

[34] J. Hoffman *et al.*, "Cycada: Cycle-consistent adversarial domain adaptation," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 1–5.

[35] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[36] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov, "Good semi-supervised learning that requires a bad gan," in *Proc. NIPS*, 2017, pp. 6510–6520.

[37] J. Dong and T. Lin, "Margingan: Adversarial training in semi-supervised learning," in *Proc. NIPS*. Del Rey, CA, USA: Curran Associates, 2019, pp. 10440–10449. [Online]. Available: http://papers.nips.cc/paper/9231-margingan-adversarial-training-in-semi-supervised-learning.pdf

[38] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," 2016, *arXiv:1610.02242*. [Online]. Available: https://arxiv.org/abs/1610.02242

[39] A. Tarvainen and H. Valpola, "Weight-averaged consistency targets improve semi-supervised deep learning results," *CoRR*, vol. 1703, p. 01780, Apr. 2018.

[40] G. Cheng, P. Zhou, and J. Han, "Duplex metric learning for image set classification," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 281–292, Jan. 2018.

[41] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018.

[42] G. Cheng, J. Han, P. Zhou, and D. Xu, "Learning rotation-invariant and Fisher discriminative convolutional neural networks for object detection," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 265–278, Jan. 2019.

[43] Z. Deng, Y. Luo, and J. Zhu, "Cluster alignment with a teacher for unsupervised domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9944–9953.

[44] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8503–8512.

[45] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7167–7176.

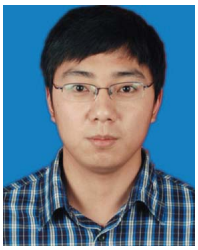
[46] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," in *Proc. ICML*, 2017, pp. 2988–2997.

[47] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo, "From source to target and back: Symmetric bi-directional adaptive GAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8018–8099.

[48] P. Haeusser, "Associative domain adaptation," in *Proc. ICCV*, 2017, pp. 2765–2773.

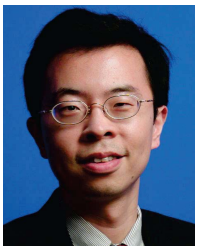
[49] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10285–10295.
[50] C. Tao, F. Lv, L. Duan, and M. Wu, "MiniMax entropy network: Learning category-invariant features for domain adaptation," 2019, *arXiv:1904.09601*. [Online]. Available: http://arxiv.org/abs/1904.09601
[51] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, "Adversarial dropout regularization," 2017, *arXiv:1711.01575*. [Online]. Available: http://arxiv.org/abs/1711.01575
[52] Y. Netzer, T. Wang, and A. Coates, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS*, Jan. 2011, pp. 1–8.
[53] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[54] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.
[55] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. AISTATS*, 2011, pp. 215–223.
[56] J. Stallkamp, M. Schlipsing, and J. Salmen, "The german traffic sign recognition benchmark: A multi-class classification competition," in *Proc. IJCNN*, 2011, pp. 1453–1460.
[57] B. Moiseev, "Evaluation of traffic sign recognition methods trained on synthetically generated data," in *Proc. ACIVS*, 2013, pp. 576–583.
[58] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," 2014, *arXiv:1405.0312*. [Online]. Available: http://arxiv.org/abs/1405.0312
[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, Mar. 2016, pp. 770–778.
[60] X. Gastaldi, "Shake-shake regularization," 2017, *arXiv:1705.07485*. [Online]. Available: http://arxiv.org/abs/1705.07485
[61] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.
[62] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009, pp. 248–255.
[63] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066–2073.
[64] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

Yukun Zuo received the B.S. degree in information security from the University of Science and Technology of China in 2018, where he is currently pursuing the Ph.D. degree. His research interests include computer vision and machine learning.



Hantao Yao (Member, IEEE) received the B.S. degree from Xidian University, Xi'an, China, in 2012, and the Ph.D. degree from the Institute of Computing Technology, University of Chinese Academy of Sciences, China, in 2018. After graduation, he worked as a Post-Doctoral with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, from 2018 to 2020. He is currently an Assistant Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. He was a recipient of the National Postdoctoral Program for Innovative Talents. His current research interests include zero-shot learning, person tracking and detection, and person re-identification.

Liansheng Zhuang (Member, IEEE) received the bachelor's and Ph.D. degrees from the University of Science and Technology of China (USTC), China, in 2001 and 2006, respectively. In 2011, he was nominated to join the Star Tracker Project of Microsoft Research of Asia (MSRA). He was a Vendor Researcher with the Visual Computing Group, Microsoft Research, Beijing. From 2012 to 2013, he was a Visiting Research Scientist with the Department of EECS, University of California at Berkeley, Berkeley. He is currently an Associate Professor with the School of Information Science and Technology, USTC. His research interests include computer vision, and machine learning. He is a member of ACM and CCF.



Changsheng Xu (Fellow, IEEE) is currently a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include multimedia content analysis/indexing/retrieval, pattern recognition, and computer vision. He has hold 50 granted/pending patents and published over 400 refereed research articles in these areas. He has served as an Associate Editor, a Guest Editor, the General Chair, the Program Chair, the Area/Track Chair, and the TPC member for over 20 IEEE and ACM prestigious multimedia journals, conferences and workshops, including IEEE TRANSACTION ON MULTIMEDIA, *ACM Transaction on Multimedia Computing, Communications and Applications*, and ACM Multimedia Conference. He is a IAPR Fellow and the ACM Distinguished Scientist.