Select, Label, and Mix: Learning Discriminative Invariant Feature Representations for Partial Domain Adaptation

Aadarsh Sahoo[1] **Rameswar Panda**[1] **Rogerio Feris**[2] **Kate Saenko**[2,3] **Abir Das**[1]
[1] IIT Kharagpur, [2] MIT-IBM Watson AI Lab, [3] Boston University
[sahoo_aadarsh@, abir@cse.]iitkgp.ac.in

Abstract

Partial domain adaptation which assumes that the unknown target label space is a subset of the source label space has attracted much attention in computer vision. Despite recent progress, existing methods often suffer from three key problems: negative transfer, lack of discriminability, and domain invariance in the latent space. To alleviate the above issues, we develop a novel 'Select, Label, and Mix' (SLM) framework that aims to learn discriminative invariant feature representations for partial domain adaptation. First, we present an efficient "select" module that automatically filters out the outlier source samples to avoid negative transfer while aligning distributions across both domains. Second, the "label" module iteratively trains the classifier using both the labeled source domain data and the generated pseudo-labels for the target domain to enhance the discriminability of the latent space. Finally, the "mix" module utilizes domain mixup jointly with the other two modules to explore more intrinsic structures across domains leading to a domain-invariant latent space for partial domain adaptation. Experiments on two datasets demonstrate the superiority of our framework over state-of-the-art methods. Project page: https://cvir.github.io/projects/slm.

1 Introduction

Partial domain adaptation (PDA), which assumes that the target label space is a subset of the source label space in unsupervised domain adaptation [8, 24], has received increasing research attention recently. Several methods have been proposed by training domain discriminators with weighting [2, 3, 27, 4], using residual correction blocks [14, 15], or reweighting source samples [1, 6, 7, 11, 25]. However, (1) most of the existing methods still suffer from negative transfer due to presence of outlier source domain classes, which cripples domain-wise transfer with untransferable knowledge; (2) in absence of labels, they often neglect the class-aware information in target domain which fails to guarantee the discriminability of the latent space; and (3) given filtering of the outliers, limited number of samples from source and target domain are not alone sufficient to learn domain invariant features for such a complex problem. As a result, a domain classifier may falsely align unlabeled target samples with samples of a different class in the source domain, leading to inconsistent predictions.

To address these challenges, we propose a novel end-to-end **Select, Label, and Mix (SLM)** framework for learning discriminative invariant features while preventing negative transfer in PDA. Our framework consists of three unique modules working in concert, *i.e.*, select, label and mix. First, the select module facilitates the identification of relevant source samples preventing the negative transfer. To be specific, our main idea is to learn a model (referred to as selector network) that outputs probabilities of binary decisions for selecting or discarding each source domain sample before aligning source and target distributions using an adversarial discriminator [9]. As these decision functions are discrete and non-differentiable, we rely on Gumbel Softmax sampling [12] to learn

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

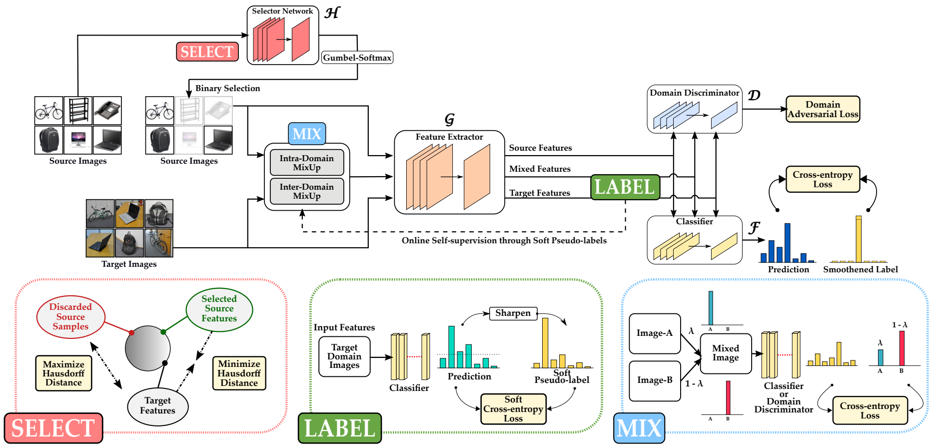

Figure 1: Illustration of our proposed framework. During training, for a mini-batch of images, all the three modules are trained jointly and during testing, we evaluate performance using classification accuracy of the network $\mathcal{F}(\mathcal{G}(.))$ on target domain data $\mathcal{D}_{target}$. See Section 2 for more details. Best viewed in color.

the policy jointly with network parameters through standard back-propagation, without resorting to complex reinforcement learning settings, as in [6, 7]. Second, we develop an efficient self-labeling strategy that iteratively trains the classifier using both labeled source domain data and generated soft pseudo-labels for target domain to enhance the discriminabilty of the latent space. Finally, the mix module utilizes both intra-domain and inter-domain mixup regularization [26] to generate convex combinations of pairs of training samples and their corresponding labels in both domains. Our proposed modules are simple yet very effective which explore three unique aspects for the first time in partial domain adaptation setting in an end-to-end manner. Specifically, in each mini-batch, our framework simultaneously eliminates negative transfer by removing outlier source samples and learns discriminative invariant features by labeling and mixing samples.

2 Proposed Method

Partial domain adaptation aims to mitigate the domain shift and generalize the model to an unlabelled target domain with a label space which is a subset of that of the labelled source domain. Formally, we define the set of labelled source domain samples as $\mathcal{D}_{source} = \{(\mathbf{x}_i^s, y_i)\}_{i=1}^{N_S}$ and unlabelled target domain samples as $\mathcal{D}_{target} = \{\mathbf{x}_i^t\}_{i=1}^{N_T}$, with label spaces $\mathcal{L}_{source}$ and $\mathcal{L}_{target}$, respectively, where $\mathcal{L}_{source} \supsetneq \mathcal{L}_{target}$. N_S and N_T represent the number of samples in source and target domain respectively. Our goal is to develop an approach with the above given data to improve the performance of a model on $\mathcal{D}_{target}$. Figure 1 illustrates an overview of our approach. Our framework consists of a feature extractor $\mathcal{G}$, a classifier network $\mathcal{F}$, a domain discriminator $\mathcal{D}$, and a selector network $\mathcal{H}$ for discarding outlier source samples ("Select") to mitigate negative transfer in partial domain adaptation. Our approach also comprises of two additional modules namely "Label" and "Mix" that works in conjunction with the "Select" module to ensure the discriminability and domain invariance of the latent space. Given a mini-batch of source and target domain images, all the components are optimized jointly in an iterative manner. The individual modules are discussed below.

Select Module. This module stands in the core of our framework with an aim to get rid of the outlier source samples in order to minimize negative transfer. Instead of using different heuristically designed criteria for weighting source samples, we develop a novel selector network $\mathcal{H}$, that takes source domain images as input and makes instance-level binary predictions to obtain *relevant* source samples for adaptation, as shown in Figure 2. Specifically, the selector $\mathcal{H}$ provides a discrete binary output of either a 0 (*discard*) or 1 (*select*) for each source sample, *i.e.*, $\mathcal{H} : \mathcal{D}_{source} \to \{0, 1\}$. However, the discrete decision policy makes the network non-differentiable and therefore difficult to optimize via standard backpropagation. To resolve non-differentiability, we adopt Gumbel-Softmax trick [12, 19] and draw samples from a categorical distribution parameterized by α_0, α_1, where α_0, α_1 are the output logits of the selector for a sample to be selected and discarded respectively. The selector $\mathcal{H}$ takes a batch of source images (size b) as input, and outputs a two-dimensional matrix $\beta \in \mathbb{R}^{b\times 2}$, where each row corresponds to $[\alpha_0, \alpha_1]$ for an image. We then draw i.i.d. samples G_0, G_1 from $Gumbel(0, 1) = -log(-log(U))$, where $U \sim$ Uniform$[0, 1]$ and generate discrete samples in forward pass as: $X = \arg\max_i[\log \alpha_i + G_i]$ resulting in hard binary predictions, while during backward pass, we approximate gradients using continuous softmax relaxation as:

$$\mathcal{Y}_i = \frac{\exp((\log \alpha_i + G_i)/\tau)}{\sum_{j \in \{0,1\}} \exp((\log \alpha_j + G_j)/\tau)} \quad \text{for } i \in \{0,1\} \tag{1}$$

where G_i's are i.i.d samples from standard Gumbel distribution $Gumbel(0,1)$ and τ denotes temperature of softmax. As τ approaches 0, $\mathcal{Y}_i$ becomes one-hot and discrete.

Learning to Discard the Outliers. We propose a novel Hausdorff distance-based triplet loss function for the select module which ensures that the selector network learns to distinguish between the outlier and the non-outlier distribution in the source domain. For a given batch of source images $\mathcal{D}^b_{source}$ and target images $\mathcal{D}^b_{target}$, each of size b, the selector results in two subsets of source samples $\mathcal{D}^b_{sel} = \{x \in \mathcal{D}^b_{source} : \mathcal{H}(x) = 1\}$ and $\mathcal{D}^b_{dis} = \{x \in \mathcal{D}^b_{source} : \mathcal{H}(x) = 0\}$. The idea is to pull the *selected* source samples $\mathcal{D}^b_{sel}$ & target samples $\mathcal{D}^b_{target}$ closer while pushing *discarded* source samples $\mathcal{D}^b_{dis}$ & $\mathcal{D}^b_{target}$ apart in the feature space of $\mathcal{G}$. To achieve this, we formulate the loss function as follows:

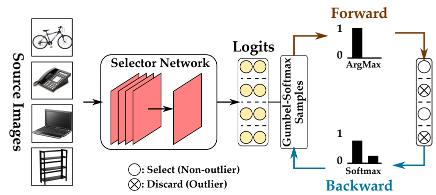

Figure 2: Learning with Gumbel Softmax Sampling.

$$d_{sel} = d_H(\mathcal{G}(\mathcal{D}^b_{sel}), \mathcal{G}(\mathcal{D}^b_{target})) \quad d_{dis} = d_H(\mathcal{G}(\mathcal{D}^b_{dis}), \mathcal{G}(\mathcal{D}^b_{target}))$$
$$\mathcal{L}_{select} = \lambda_s max(d_{sel} - d_{dis} + margin, 0) + \mathcal{L}_{reg} \tag{2}$$

where $d_H(X,Y)$ represents the average Hausdorff distance between the set of features X and Y. $\mathcal{L}_{reg} = \lambda_{reg1} \sum_{x \in \mathcal{D}^b_{source}} \mathcal{H}(x) \log(\mathcal{H}(x)) + \lambda_{reg2} \{\sum_{\hat{p}} l_{ent}(\hat{p}) - l_{ent}(\hat{p}_m)\}$, with l_{ent} being the entropy loss, $\hat{p}$ is the Softmax prediction of $\mathcal{F}(\mathcal{G}(\mathcal{D}_{target}))$ and $\hat{p}_m$ is mean prediction for the target domain. $\mathcal{L}_{reg}$ is a regularization to restrict $\mathcal{H}$ from producing trivial all-0 or all-1 outputs as well as ensuring confident and diverse predictions by $\mathcal{F}(\mathcal{G}(.))$ for $\mathcal{D}_{target}$. Note that given the selection $\mathcal{D}^b_{sel}$, we forward only the selected samples to the successive modules.

Label Module. While select module helps in removing source domain outliers, it fails to guarantee discriminability of the latent space due to absence of class-aware information in target domain. To this end, we propose a label module that provides additional self-supervision for target domain samples. Specifically, we generate soft pseudo-labels [28] for target domain samples that efficiently attenuates the unwanted deviations caused by false and noisy one-hot pseudo-labels. For a target domain sample $\mathbf{x}^t_k \in \mathcal{D}_{target}$, the soft-pseudo-label $\hat{y}^t_k$ and the corresponding loss is computed as follows:

$$\hat{y}^{t(i)}_k = \frac{p(i|\mathbf{x}^t_i)^{\frac{1}{\alpha}}}{\sum_{j=1}^{|\mathcal{L}_{source}|} p(j|\mathbf{x}^t_i)^{\frac{1}{\alpha}}} \quad \mathcal{L}_{label} = \mathbb{E}_{\mathbf{x}^t_i \in \mathcal{D}^b_{target}} l_{ce}(\mathcal{F}(\mathcal{G}(\mathbf{x}^t_i)), \hat{y}^t_i) \tag{3}$$

where $p(j|\mathbf{x}^t_i)$ is the softmax probability of the classifier for class j given $\mathbf{x}^t_i$ as input, and α controls the softness of the label. $\mathcal{D}^b_{target}$ is a batch of target samples, $l_{ce}(.)$ represents the cross-entropy loss.

Mix Module. With limited samples per batch and after discarding the outlier samples, it becomes more challenging in preventing over-fitting and learning domain invariant representation using only select and label modules. Thus, we apply MixUp [26] on the selected source samples and the target samples for discovering ingrained structures in establishing domain invariance. Given $\mathcal{D}^b_{sel}$ from select module and $\mathcal{D}^b_{target}$ with corresponding labels $\hat{y}^t$ from label module, we perform convex combinations of images belonging to these two sets on pixel-level in three different ways namely, *inter-domain*, *intra-source* domain and *intra-target* domain to obtain $\mathcal{D}^b_{inter\_mix}$, $\mathcal{D}^b_{intra\_mix\_s}$, and $\mathcal{D}^b_{intra\_mix\_t}$ respectively. Given the new augmented images, we utilize these three sets in training both the classifier $\mathcal{F}$ and the domain discriminator $\mathcal{D}$ as follows:

$$\mathcal{L}_{mix\_dom} = \mathbb{E}_{\mathbf{x}_i \sim \mathcal{D}^b_{inter\_mix}} [\lambda \log(\mathcal{D}(\mathcal{G}(\mathbf{x}_i))) + (1-\lambda)\log(1-\mathcal{D}(\mathcal{G}(\mathbf{x}_i)))]$$
$$+ \mathbb{E}_{\mathbf{x}_i \sim \mathcal{D}^b_{intra\_mix\_s}} \log(\mathcal{D}(\mathcal{G}(\mathbf{x}_i))) + \mathbb{E}_{\mathbf{x}_i \sim \mathcal{D}^b_{intra\_mix\_t}} \log(1-\mathcal{D}(\mathcal{G}(\mathbf{x}_i)))$$
$$\mathcal{L}_{mix\_cls} = \mathbb{E}_{(\mathbf{x}_i, y_i) \in \mathcal{D}^b_{mix}} l_{ce}(\mathcal{F}(\mathcal{G}(\mathbf{x}_i)), y_i), \quad \mathcal{L}_{mix} = \mathcal{L}_{mix\_cls} + \mathcal{L}_{mix\_dom} \tag{4}$$

where $\mathcal{D}^b_{mix} = \mathcal{D}^b_{inter\_mix} \cup \mathcal{D}^b_{intra\_mix\_s} \cup \mathcal{D}^b_{intra\_mix\_t}$. $\mathcal{L}_{mix\_dom}$ and $\mathcal{L}_{mix\_cls}$ represent loss for domain discriminator and classifier respectively.

Optimization. Besides the above three modules that are tailored for PDA, we use standard supervised loss on the labeled source data and domain adversarial loss as follows:

$$\mathcal{L}_{sup} = \mathbb{E}_{(\mathbf{x}_i, y_i) \in \mathcal{D}^b_{sel}} l_{ce}(\mathcal{F}(\mathcal{G}(\mathbf{x}_i)), y_i)$$
$$\mathcal{L}_{adv} = \mathbb{E}_{\mathbf{x}^s \sim \mathcal{D}^b_{sel}} w^s \log(\mathcal{D}(\mathcal{G}(\mathbf{x}^s))) + \mathbb{E}_{\mathbf{x}^t \sim \mathcal{D}^b_{target}} w^t \log(1 - \mathcal{D}(\mathcal{G}(\mathbf{x}^t))) \tag{5}$$

where $\mathcal{L}_{adv}$ is entropy-conditioned domain adversarial loss with weights w^s and w^t for source and target domain respectively [17]. We integrate all the modules into one framework, as shown in Figure 1 and minimize all losses together to train the networks jointly for partial domain adaptation.

3 Experiments

Datasets and Settings. We use Office31 [22], and VisDA-2017 [20] for experiments. We follow [14] and select target categories for each transfer task under PDA setting. We use ResNet-50 [10] as feature extractor and ResNet-18 as selector network, initialized with ImageNet [21] pretrained weights. In Eqn. 2 we set $\lambda_s, \lambda_{reg1}$ and λ_{reg2} as $0.01, 10.0$ and 0.1, respectively, and a margin value of 100.0. We set $\tau = 1.0$ in Eqn. 1, $\alpha = 0.1$ in Eqn. 3 and are annealed down to 0 during training. We report average classification accuracy and standard deviation over 3 random trials.

Table 1: Performance on Office31 and VisDA-2017. We highlight the **best** and $\underline{\text{second best}}$ method on each transfer task under PDA setting.

	Office31							VisDA-2017		
Method	**A → W**	**D → W**	**W → D**	**A → D**	**D → A**	**W → A**	**Average**	**R → S**	**S → R**	**Average**
ResNet-50 [10]	$76.5_{\pm0.3}$	$99.2_{\pm0.2}$	$97.7_{\pm0.1}$	$87.5_{\pm0.2}$	$87.2_{\pm0.1}$	$84.1_{\pm0.3}$	88.7	64.3	45.3	54.8
DAN [16]	$53.6_{\pm0.7}$	$62.7_{\pm0.5}$	$57.8_{\pm0.6}$	$47.7_{\pm0.5}$	$61.2_{\pm0.6}$	$69.7_{\pm0.5}$	58.8	68.4	47.6	58.0
DANN [9]	$62.8_{\pm0.6}$	$71.6_{\pm0.4}$	$65.6_{\pm0.5}$	$65.1_{\pm0.7}$	$78.9_{\pm0.3}$	$79.2_{\pm0.4}$	70.5	73.8	51.0	62.4
ADDA [23]	$75.7_{\pm0.2}$	$95.4_{\pm0.2}$	$\underline{99.9}_{\pm0.1}$	$83.4_{\pm0.2}$	$83.6_{\pm0.1}$	$84.3_{\pm0.1}$	87.0	–	–	–
RTN [18]	75.3	97.1	98.3	66.9	85.6	85.7	84.8	72.9	50.0	61.5
CDAN+E [17]	$80.5_{\pm1.2}$	$99.0_{\pm0.0}$	$98.1_{\pm0.0}$	$77.1_{\pm0.9}$	$93.6_{\pm0.1}$	$91.7_{\pm0.0}$	90.0	–	–	–
JDDA [5]	$73.5_{\pm0.6}$	$93.1_{\pm0.3}$	$89.3_{\pm0.2}$	$76.4_{\pm0.4}$	$77.6_{\pm0.1}$	$82.8_{\pm0.2}$	82.1	–	–	–
CAN [13]	$84.4_{\pm0.0}$	$92.0_{\pm1.4}$	$94.7_{\pm1.7}$	$84.9_{\pm0.9}$	$85.6_{\pm1.0}$	$86.4_{\pm0.8}$	88.0	–	–	–
PADA [3]	$86.3_{\pm0.4}$	$\underline{99.3}_{\pm0.1}$	$\mathbf{100.0}_{\pm0.0}$	$90.4_{\pm0.1}$	$91.3_{\pm0.2}$	$92.6_{\pm0.1}$	93.3	$\underline{76.5}$	53.5	65.0
SAN [2]	$93.9_{\pm0.5}$	$\underline{99.3}_{\pm0.5}$	$99.4_{\pm0.1}$	$94.3_{\pm0.3}$	$94.2_{\pm0.4}$	$88.7_{\pm0.4}$	95.0	69.7	49.9	59.8
IWAN [27]	$89.2_{\pm0.4}$	$\underline{99.3}_{\pm0.3}$	$99.4_{\pm0.2}$	$90.5_{\pm0.4}$	$\underline{95.6}_{\pm0.3}$	$94.3_{\pm0.3}$	94.7	71.3	48.6	60.0
ETN [4]	$93.4_{\pm0.3}$	$\underline{99.3}_{\pm0.1}$	$99.2_{\pm0.2}$	$95.5_{\pm0.4}$	$95.4_{\pm0.1}$	$91.7_{\pm0.2}$	95.8	–	–	–
DRCN [14]	88.1	**100.0**	**100.0**	86.0	$\underline{95.6}$	$\underline{95.8}$	94.3	73.2	58.2	$\underline{65.7}$
RTNet [7]	$95.1_{\pm0.3}$	$\mathbf{100.0}_{\pm0.0}$	$\mathbf{100.0}_{\pm0.0}$	$97.8_{\pm0.1}$	$93.9_{\pm0.1}$	$94.1_{\pm0.1}$	96.8	–	–	–
$\text{RTNet}_{\text{adv}}$ [7]	$96.2_{\pm0.3}$	$\mathbf{100.0}_{\pm0.0}$	$\mathbf{100.0}_{\pm0.0}$	$97.6_{\pm0.1}$	$92.3_{\pm0.1}$	$95.4_{\pm0.1}$	96.9	–	–	–
BA^3US [15]	$\underline{99.0}_{\pm0.3}$	$\mathbf{100.0}_{\pm0.0}$	$98.7_{\pm0.0}$	$\mathbf{99.4}_{\pm0.0}$	$94.8_{\pm0.1}$	$95.0_{\pm0.1}$	$\underline{97.8}$	–	–	–
SLM (Ours)	$\mathbf{99.8}_{\pm0.2}$	$\mathbf{100.0}_{\pm0.0}$	$99.8_{\pm0.3}$	$\underline{98.7}_{\pm0.0}$	$\mathbf{96.1}_{\pm0.1}$	$\mathbf{95.9}_{\pm0.0}$	**98.4**	$\mathbf{77.5}_{\pm0.8}$	$\mathbf{91.7}_{\pm0.8}$	**84.6**

Results and Analysis. Table 1 shows the results. On Office31, as expected, the popular UDA methods including the recent CAN [13], fail to outperform the simple no adaptation model (ResNet-50), which implies that they suffer from negative transfer (see upper section in Table 1). Overall, our **SLM** framework outperforms all the existing PDA methods by achieving the best results on **4 out of 6** transfer tasks. Among PDA methods, BA^3US [15] is the most competitive. However, **SLM** still outperforms it (97.8% vs 98.4%) due to our two novel components working in concert with the removal of outliers: enhancing discriminability of the latent space via iterative pseudo-labeling of target domain samples and learning domain-invariance through mixup regularizations. On VisDA-2017 dataset, our approach achieves new state-of-the-art result, outperforming the next competitive method by a large margin of about $\mathbf{18.9\%}$. Our approach is able to distill more positive knowledge from the synthetic to the real domain despite significant domain gap across them.

Effectiveness of Individual Modules. On Office-31, we find that the Select only module improves the vanilla performance by 5.6%, while addition of Label and Mix modules progressively improves the result to obtain the best performance of 98.4% ($89.3 \xrightarrow{\text{S}} 94.9 \xrightarrow{\text{S+L}} 96.0 \xrightarrow{\text{S+L+M}} 98.4$). This corroborates the fact that both discriminability and invariance of the latent space plays a crucial role in partial domain adaptation in addition to the removal of source domain outlier samples.

Distance between Domains. We compute the Wasserstein distance between the probability distribution of the target samples (T) with that of the selected (S_{sel}) and discarded samples (S_{dis}) by the selector network [7]. Assuming $\texttt{dist}(\text{S}_{all}, \text{T})$ to be equal to 1.000, we find that $\texttt{dist}(\text{S}_{sel}, \text{T})$ is smaller than $\texttt{dist}(\text{S}_{all}, \text{T})$ (0.999 & 0.893), while $\texttt{dist}(\text{S}_{dis}, \text{T})$ is greater than $\texttt{dist}(\text{S}_{all}, \text{T})$ (1.013 & 1.144) on two randomly sampled adaptation tasks from Office31 (A → D & W → A). This results indicate that the samples selected by our selector network are closer to the target domain while the discarded samples are very dissimilar to the target domain.

4 Conclusion

In this paper, we propose an end-to-end framework for learning discriminative invariant feature representation while preventing negative transfer in partial domain adaptation. While our “select” module facilitates the identification of relevant source samples for adaptation, “label” module enhances discriminability of the latent space utilizing pseudo-labels, with “mix” module using mixup regularizations jointly with the former two strategies to enforce domain invariance in latent space.

Acknowledgements. This work was partially supported by the SERB Grant SRG/2019/001205.

References

[1] Silvia Bucci, Antonio D'Innocente, and Tatiana Tommasi. Tackling partial domain adaptation with self-supervision. In *International Conference on Image Analysis and Processing*, pages 70–81. Springer, 2019.

[2] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Partial transfer learning with selective adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2724–2732, 2018.

[3] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018.

[4] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2985–2994, 2019.

[5] Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3296–3303, 2019.

[6] Jin Chen, Xinxiao Wu, Lixin Duan, and Shenghua Gao. Domain adversarial reinforcement learning for partial domain adaptation. *arXiv preprint arXiv:1905.04094*, 2019.

[7] Zhihong Chen, Chao Chen, Zhaowei Cheng, Boyuan Jiang, Ke Fang, and Xinyu Jin. Selective transfer with reinforced transfer network for partial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12706–12714, 2020.

[8] Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

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

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Jian Hu, Hongya Tuo, Chao Wang, Lingfeng Qiao, Haowen Zhong, and Zhongliang Jing. Multi-weight partial domain adaptation. In *BMVC*, page 5, 2019.

[12] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[13] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

[14] Shuang Li, Chi Harold Liu, Qiuxia Lin, Qi Wen, Limin Su, Gao Huang, and Zhengming Ding. Deep residual correction network for partial domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[15] Jian Liang, Yunbo Wang, Dapeng Hu, Ran He, and Jiashi Feng. A balanced and uncertainty-aware approach for partial domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 123–140. Springer, 2020.

[16] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning Transferable Features with Deep Adaptation Networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[17] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional Adversarial Domain Adaptation. In *Neural Information Processing Systems*, pages 1640–1650, 2018.

[18] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in neural information processing systems*, pages 136–144, 2016.

[19] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

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

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[22] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[23] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[24] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[25] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1426–1435, 2019.

[26] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[27] Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8156–8164, 2018.

[28] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5982–5991, 2019.