DUAL-CONSISTENCY SELF-TRAINING FOR UNSUPERVISED DOMAIN ADAPTATION

Jie Wang[⋆]*, Chaoliang Zhong*[⋆]*, Cheng Feng*[⋆]*, Jun Sun*[⋆]*, Masaru Ide*[†] *and Yasuto Yokota*[†]

[⋆] Fujitsu R&D Center, Co., LTD, Beijing, China
[†] Fujitsu Laboratories, Kawasaki, Japan

2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506074

ABSTRACT

Unsupervised domain adaptation (UDA) is a challenging task characterized by unlabeled target data with domain discrepancy to labeled source data. Many methods have been proposed to learn domain invariant features by marginal distribution alignment, but they ignore the intrinsic structure within target domain, which may lead to insufficient or false alignment. Class-level alignment has been demonstrated to align the features of the same class between source and target domains. These methods rely extensively on the accuracy of predicted pseudo-labels for target data. Here, we develop a novel self-training method that focuses more on accurate pseudo-labels via a dual-consistency strategy involving modelling the intrinsic structure of the target domain. The proposed dual-consistency strategy first improves the accuracy of pseudo-labels through voting consistency, and then reduces the negative effects of incorrect predictions through structure consistency with the relationship of intrinsic structures across domains. Our method has achieved comparable performance to the state-of-the-arts on three standard UDA benchmarks.

***Index Terms*—** Consistency, Self-training, Unsupervised Domain Adaptation

1. INTRODUCTION

Unsupervised domain adaptation (UDA) is a popular task used to transfer the model capability from a source domain with labeled data to a target domain with only unlabeled data [1, 2, 3, 4]; however, domain discrepancy or domain shift makes it challenging. The mainstream UDA methods focus on learning domain invariant features [3] with moment matching or adversarial training. Moment matching can minimize the difference in feature covariances [5] or the maximum mean discrepancy of features across domains [2, 6]. Adversarial training adopts a domain discriminator to confuse the feature extractor (encoder) either through a gradient reversal layer [3] or through alternating updates to the discriminator and encoder [7, 8, 9]; however, bridging the cross-domain gap without considering the structures within the domain may lead to under or negative transfer [10]. To deal with this issue, class-wise alignment methods have been proposed, such as multi-adversarial training [10], conditional discriminator [11] and semantic alignment [12, 13, 14], however, they are vulnerable to error accumulation due to wrongly predicted labels for target data, thus pseudo-labels from the classifier prediction are essential in these methods.

Pseudo-labels are not only used in class-level alignment, but are also frequently used in self-training [15], especially in semi-supervised tasks [16]. Self-training can optimize the entire network in a supervised manner with target data and their pseudo-labels, which not only yields better decision boundaries, but also renders the feature embeddings across domains. Consequently, self-training can also be regarded as a class-wise domain adaptation method. Similarly, the accuracy of pseudo-labels plays an important role in self-training. Since pseudo-labels are probably incorrect, some coping strategies have been put forward, such as introducing a learnable confusion matrix to correct the wrong labels [16], adopting a self-ensembling teacher model to improve the predictions [17], using entropy to quantify the uncertainty [11], setting a threshold to select samples whose pseudo-labels are more likely to be correct [14], or choosing the top K high-probability samples as the reliable ones [18]. These methods are either sensitive to additional net module [16] and hyper-parameters [14, 18], or adopt the model predictions as the only knowledge for pseudo-labels [11, 17].

In the present work, we propose the use of dual-consistency self-training (DCS) for UDA, consisting of a voting consistency via hard binary weighting to select prediction-consistent target samples and a structure consistency to take the intrinsic structure of the target domain into consideration (Fig. 1). These two consistency constraints facilitate the effective usage of pseudo-labels.

Our contributions can be summarized as follows:

- We propose a novel dual-consistency self-training method for UDA. The proposed dual-consistency score is the product of the voting-consistency score voted by classifier prediction and nearest neighbor-based prediction and a structure-consistency score calculated as the cosine similarity of features across domains.
- Our method has been evaluated on three popular cross-domain benchmarks and has achieved comparable results to the state-of-the-art approaches.

2. PROPOSED METHOD

Given a labeled source domain $\mathcal{D}_s = \{(x_s^i, y_s^i)\}_{i=1}^{n_s}$ and an unlabeled target domain $\mathcal{D}_t = \{x_t^j\}_{j=1}^{n_t}$, the aim of UDA is to train a model with relatively high performance in the target domain. In our framework, a feature extractor network G extracts the high-level feature from the data x_s or x_t, a classifier network C undertakes the K-class classification task on the feature space, and a domain discriminator network D that connects to G via a gradient reversal layer discriminates by the input feature domain. The classifier C outputs probability vectors $\mathbf{p}_s, \mathbf{p}_t \in \mathbb{R}^K$ for x_s and x_t, respectively. The discriminator D provides a probability value indicating the confidence of the domain source for the input feature.

2.1. Voting consistency

We first develop a voting consistency via a hard binary weighting to select certain pseudo-labeled target samples for self-training to adapt the classifier to the target domain. The certainty is decided by the consistency of two different predictions: the classifier prediction $y_{t,c}$ and the nearest neighbor-based prediction $y_{t,d}$. Then the voting-consistency score for target sample x_t^j is defined as

$$v(x_t^j) = \begin{cases} 1 & \text{if } y_{t,c}^j = y_{t,d}^j, \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where the classifier prediction $y_{t,c}^j = arg\max\ \mathbf{p}_t^j$.

The nearest neighbor-based prediction refers to the nearest source class prototype to a specific target sample. The source class prototype is defined as the average feature embedding of the source samples with the same ground-truth labels. For example, the class prototype for source class k is denoted as:

$$\lambda_s^k = \frac{1}{n_s^k} \sum_{x_s^i \in \mathcal{D}_s^k} G\left(x_s^i\right) \tag{2}$$

where $\mathcal{D}_s^k$ denotes the set of samples labeled as class k in the source domain and n_s^k represents the number of corresponding samples. The class prototypes are updated in each iteration by an exponential moving average technique [19]. For example, the prototype of class k is updated using

$$\lambda_{s(I)}^k \leftarrow \rho\lambda_{s(I)}^k + (1-\rho)\lambda_{s(I-1)}^k \tag{3}$$

where I denotes the current training iteration, the decay coefficient ρ adjusts the velocity of the update, and $\lambda_{s(0)}$ is initialized as zero.

Then the squared Euclidean distance is used to decide the nearest neighbor-based prediction $y_{t,d}^j = arg\min\ \mathbf{d}_t^j$:

$$\mathbf{d}_t^{j,k} = ||G(x_t^j) - \lambda_s^k||^2, k = \{1, 2, ..., K\} \tag{4}$$

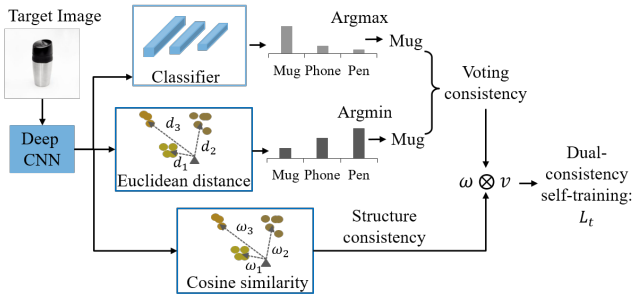


Fig. 1. The proposed dual-consistency self-training (L_t). Given a target image (triangle), the first voting consistency (v) decides whether to choose it or not according to the consistency between the classifier prediction and the nearest neighbor-based prediction, and the next structure consistency uses cosine similarity as its soft weight (ω).

where $\mathbf{d}_t^{j,k}$ is the k^{th} element of $\mathbf{d}_t^j \in \mathbb{R}^K$ which represents the distance vector of the target sample x_t^j to all source class prototypes.

2.2. Structure consistency

Considering that the pseudo-labels of selected target samples after voting are still not guaranteed to be correct, we further introduce a soft weighting involving the intrinsic structure of the target domain to provide a structure-consistency score. The feature space models the intrinsic structure, so we leverage the relationship of features across domains to give a structure hint. Specifically, the cosine similarity between the feature of a target sample and the corresponding source class prototype with the same label k is adopted as the structure-consistency score to reduce the effects of incorrect predictions, i.e, $\omega_{x_t^j} = CS(G(x_t^j), \lambda_s^k)$ where $CS(.,.)$ represents the cosine similarity function.

Hence, the voting-consistency and the structure-consistency scores are multiplied to give the final dual-consistency score. Then the loss of dual-consistency self-training for the target domain is defined as

$$L_t = -\frac{1}{n_t} \sum_{j=1}^{n_t} \omega_{x_t^j} v(x_t^j) \log(p_{x_t^j}) \tag{5}$$

where $\omega_{x_t^j}$ denotes the cosine similarity, and $p_{x_t^j}$ represents the predicted probability of the pseudo-label $y_{t,c}^j$ for x_t^j.

2.3. Semantic alignment based on voting consistency

The accuracy of the pseudo-labels not only affects the self-training, but also influences the target class prototypes. If samples near the decision boundary are wrongly labeled, they pull the class centroid away from the correct position. Hence, the voting consistency is used to get a sub-dataset of target domain $\hat{\mathcal{D}}_t = \{\hat{x}_t^j, \hat{y}_t^j\}_{j=1}^{\hat{n}_t}$ where $v(\hat{x}_t^j) = 1$ is used to calculate

a more accurate class prototype for each class. Similar to the source domain, the target prototype for class k is formulated and updated as

$$\lambda_t^k = \frac{1}{\hat{n}_t^k} \sum_{\hat{x}_t^j \in \hat{\mathcal{D}}_t^k} G\left(\hat{x}_t^j\right) \tag{6}$$

$$\lambda_{t(I)}^k \leftarrow \rho \lambda_{t(I)}^k + (1-\rho)\lambda_{t(I-1)}^k \tag{7}$$

where $\hat{x}_t^j$ is the target sample in the selected target domain $\hat{\mathcal{D}}_t$ with pseudo-label k, $\hat{n}_t^k$ represents the number of samples in class k, and $\lambda_{t(0)}$ is also initialized as zero.

Then, the semantic alignment loss is defined to minimize the distance between the source prototype and the corresponding target prototype as

$$L_{sa}\left(\mathcal{D}_s, \hat{\mathcal{D}}_t\right) = \frac{1}{K} \sum_{k=1}^{K} \left\| \lambda_s^k - \lambda_t^k \right\|^2. \tag{8}$$

Overall, the loss function of our solution is described as

$$L = \varphi_1 L_t + \varphi_2 L_{sa} + L_s + L_{adv} \tag{9}$$

where φ_1 and φ_2 balance self-training loss and semantic alignment loss with the standard cross-entropy loss for labeled source data L_s and the adversarial training loss L_{adv} via a gradient reversal layer as in [3].

3. EXPERIMENTS AND RESULTS

The proposed dual-consistency self-training (DCS) is compared with state-of-the-art domain adaptation methods: Deep Adaptation Network (DAN) [2], Domain Adversarial Neural Network (DANN) [3], Adversarial Discriminative Domain Adaptation (ADDA) [8], Joint Adaptation Network (JAN) [6], Multi-Adversarial Domain Adaptation (MADA) [10], Conditional Domain Adversarial Network (CDAN) [11], Moving Semantic Transfer Network (MSTN) [12], Cluster Alignment with Teacher (CAT) [13], Self-adaptive Re-weighted Adversarial Domain Adaptation (SRDA) [20], Adversarial-Learned Domain Adaptation (ALDA) [21], and Selective Pseudo-Labelling (SPL) [18].

We conduct experiments on three popular UDA datasets: *Office-31* [22], *Office-Home* [23], and *ImageCLEF-DA* [24]. **Office-31** contains 4,110 images under 31 categories in the office environment collected from three distinct domains: *Amazon* (A, images downloaded from on-line merchants), *Webcam* (W, low-resolution images recorded using a web camera), and *DSLR* (D, high-resolution images recorded using a digital SLR camera). **Office-Home** is a more complex dataset with 15,588 images under 65 categories that are common in office and home scenarios. This dataset is collected from four different domains: Artistic images (Ar), Clipart (Cl), Product images (Pr) and Real-World images (Rw).

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet-50 [26]	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DAN [2]	80.5	97.1	99.6	78.6	63.6	62.8	80.4
DANN [3]	82.0	96.9	99.1	79.7	68.2	67.4	82.2
ADDA [8]	86.2	96.2	98.4	77.8	69.5	68.9	82.9
JAN [6]	85.4	97.4	99.8	84.7	68.6	70.0	84.3
MADA [10]	90.0	97.4	99.6	87.8	70.3	66.4	85.2
CDAN [11]	94.1	98.6	**100.0**	92.9	71.0	69.3	87.7
MSTN [12]	91.3	**98.9**	**100.0**	90.4	72.7	65.6	86.5
CAT [13]	94.4	98.0	**100.0**	90.8	72.2	70.2	87.6
ALDA [21]	95.6	97.7	**100.0**	**94.0**	72.2	72.5	88.7
SRDA [20]	95.2	98.6	**100.0**	91.7	74.5	73.7	89.0
SPL [18]	92.7	98.7	99.8	93	**76.4**	**76.8**	89.6
w/o voting	95.1	98.2	99.9	91.8	74.8	74.5	89.0
w/o structure	93.9	98.5	99.9	92.6	73.6	75.7	89.0
DCS (Ours)	**95.7**	98.4	**100.0**	93.4	75.9	76.4	**90.0**

Table 1. Accuracy (%) of different unsupervised domain adaptation methods on Office-31 (ResNet-50). The best is in bold and the second-best is underlined.

ImageCLEF-DA is released as a benchmark dataset for ImageCLEF 2014 domain adaptation challenge containing 12 common categories shared by public datasets and each is considered as a domain. Experiments are conducted on three domains: Caltech-256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC 2012 (P).

We follow the standard evaluation protocols [3, 11], repeat three random experiments for each task, and report the average classification accuracy and standard variation. We compare the average classification accuracy with other methods whose results are directly from original papers or reproduced literature.

We implement our method on Pytorch [25], and use ResNet-50 [26] pre-trained with ImageNet [27] as the backbone. The discriminator consists of three fully connected layers with dropout as reported elsewhere [11, 21]. The discriminator and classifier layers are trained from scratch with learning rate 10 times that of the lower layers through back-propagation. We adopt mini-batch SGD with momentum of 0.9 and the learning rate annealing strategy as described in [3, 11]. The total number of iterations is 20,000 to guarantee the convergence. The decay coefficient ρ in Eqs. 3 and 7 is set to 0.7, and the coefficients φ_1 and φ_2 in Eq. 9 are set to 0.1 and 10 respectively.

We summarize the experimental results of *Office-31* in Table 1. Our method achieves the highest average accuracy of 90.0% among all the compared approaches. As shown in Table 2, our method establishes a new record on *Office-Home* with an improvement of 0.3% on the average accuracy. Table 3 displays the experimental results on *ImageCLEF-DA*. Similarly, DCS obtains the best average accuracy of 90.5%.

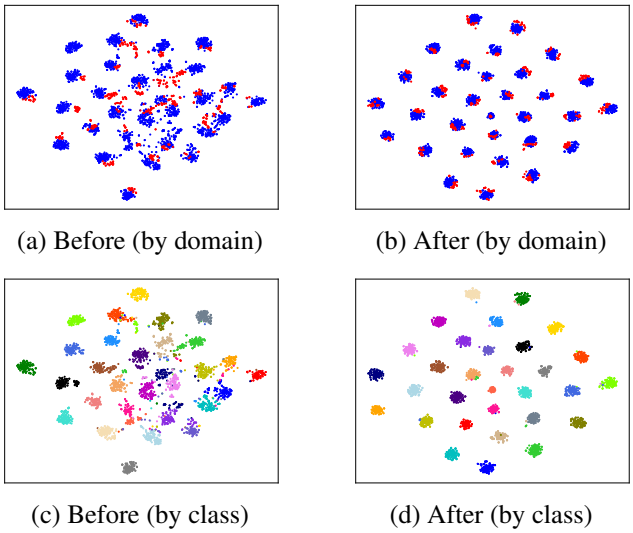
The proposed dual-consistency self-training consists of voting consistency and structure consistency. We demon-

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
ResNet-50 [26]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [2]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN [3]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN [6]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN [11]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
MSTN [12]	49.8	70.3	76.3	60.4	68.5	69.6	61.4	48.9	75.7	70.9	55.0	81.1	65.6
ALDA [21]	53.7	70.1	76.4	60.2	72.6	71.5	56.8	51.9	77.1	70.2	56.3	82.1	66.6
SRDA [20]	55.5	73.5	78.7	60.7	74.1	73.1	59.5	**55.0**	80.4	72.4	**60.3**	84.3	68.9
SPL [18]	54.5	**77.8**	**81.9**	65.1	**78.0**	**81.1**	**66.0**	53.1	**82.8**	69.9	55.3	**86.0**	71.0
DCS (Ours)	**56.3**$^{\pm 0.0}$	76.6$^{\pm 0.5}$	81.2$^{\pm 0.1}$	**67.6**$^{\pm 0.1}$	75.5$^{\pm 0.0}$	77.1$^{\pm 0.1}$	65.1$^{\pm 0.0}$	54.9$^{\pm 0.1}$	81.6$^{\pm 0.3}$	**74.8**$^{\pm 0.1}$	**60.3**$^{\pm 0.0}$	84.6$^{\pm 0.1}$	**71.3**

Table 2. Accuracy (%) of different unsupervised domain adaptation methods on Office-Home (ResNet-50)

Method	I→P	P→I	I→C	C→I	C→P	P→C	Avg
ResNet-50 [26]	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DAN [2]	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DANN [3]	75.0	86.0	96.2	87.0	74.3	91.5	85.0
JAN [6]	76.8	88.4	94.8	89.5	74.2	91.7	85.8
MADA [10]	75.0	87.9	96.0	88.8	75.2	92.2	87.7
CDAN [11]	77.7	90.7	97.7	91.3	74.2	94.3	85.8
MSTN [12]	77.3	91.3	96.8	91.2	77.7	95.0	88.2
CAT [13]	77.2	91.6	95.5	91.3	75.3	93.6	87.3
SRDA [20]	78.3	91.3	96.7	90.5	78.1	96.2	88.5
SPL [18]	78.3	**94.5**	96.7	**95.7**	**80.5**	**96.3**	90.3
DCS (Ours)	**80.2**	94.3	**97.8**	94.1	80.3	96.1	**90.5**

Table 3. Accuracy (%) of different unsupervised domain adaptation methods on ImageCLEF-DA (ResNet-50)



(a) Before (by domain) (b) After (by domain)

(c) Before (by class) (d) After (by class)

Fig. 2. (Best viewed in color) The t-SNE visualization of deep features on task A → W of *Office-31*. DCS improves the alignment between source domain and target domain (blue: A, red: W) as shown in (b), and renders the target samples discriminative as illustrated in (d).

strate the contributions of these two components with the average accuracy of three random experiments on *Office-31*. As shown in Table 1, removing any component impairs the performance. The performance is decreased by 1.0% in the absence of voting consistency (w/o voting) or structure consistency (w/o structure). Furthermore, we demonstrate the selected pseudo-labels via voting consistency have higher accuracy than classifier prediction or nearest neighbor-based prediction. Taking task A → W for example, the accuracy of selected pseudo-labels surpasses those of classifier prediction and nearest neighbor-based prediction by 1.4% and 1.3% , respectively.

We visualize the feature embeddings of our approach using t-SNE [28] and compare the output with the baseline method (Fig. 2) on task A → W on *Office-31*. Apparently, feature distributions of different domains are quite different without domain adaptation, and target features show poor discriminative structures (Fig. 2(a)). After using our method, the source and target features are close to each other and show a good cluster structure (Fig. 2(b)), and target features become discriminative (Fig. 2(d)).

4. CONCLUSION

In the present work, we present an end-to-end self-training method to tackle the classifier adaptation in unsupervised domain adaptation. The proposed dual-consistency strategy fully considers the intrinsic structure of the target domain, leveraging both classifier prediction and nearest neighbor-based prediction to conduct a binary weighting and further using the cosine similarity to the source class prototype as a soft weight to reduce the negative effects of incorrectly predicted labels. Our method achieves competitive results in three standard UDA benchmarks. In future, we intend to work on the development of a more powerful strategy to improve the precision of rejection of wrongly-predicted labels.

5. REFERENCES

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan, "A theory of learning from different domains," *Machine Learning*, vol. 79, no. 1, pp. 151–175, May 2010.

[2] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan, "Learning transferable features with deep adaptation networks," in *ICML*, 2015.

[3] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky, "Domain-adversarial training of neural networks," *JMLR*, vol. 17, pp. 2096–2030, 2016.

[4] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *ECCV*, 2016.

[5] Baochen Sun and Kate Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *ECCV Workshops*, 2016.

[6] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan, "Deep transfer learning with joint adaptation networks," in *ICML*, 2017.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.

[8] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, "Adversarial discriminative domain adaptation," in *CVPR*, 2017.

[9] Sicheng Zhao, Guangzhi Wang, Shanghang Zhang, Yang Gu, and Kurt Keutzer, "Multi-source distilling domain adaptation," in *AAAI*, 2020.

[10] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang, "Multi-adversarial domain adaptation," in *AAAI*, 2018.

[11] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan, "Conditional adversarial domain adaptation," in *NeurIPS*, 2018, pp. 1645–1655.

[12] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen, "Learning semantic representations for unsupervised domain adaptation," in *ICML*, 2018.

[13] Zhijie Deng, Yucen Luo, and Jun Zhu, "Cluster alignment with a teacher for unsupervised domain adaptation," in *ICCV*, 2020.

[14] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang, "Progressive feature alignment for unsupervised domain adaptation," in *CVPR*, 2019.

[15] Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *ECCV*, 2018.

[16] Xinzhe Li, Qianru Sun, Yaoyao Liu, Shibao Zheng, Qin Zhou, Tat-Seng Chua, and Bernt Schiele, "Learning to self-train for semi-supervised few-shot classification," in *NeurIPS*, 2019.

[17] Antti Tarvainen and Harri Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *NeurIPS*, 2017.

[18] Qian Wang and Toby P Breckon, "Unsupervised domain adaptation via structured prediction based selective pseudo-labeling," in *AAAI*, 2020.

[19] Geoffrey French, Michal Mackiewicz, and Mark Fisher, "Self-ensembling for visual domain adaptation," in *ICLR*, 2018.

[20] Shanshan Wang and Lei Zhang, "Self-adaptive re-weighted adversarial domain adaptation," in *IJCAI*, 2020.

[21] Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai, "Adversarial-learned loss for domain adaptation," in *AAAI*, 2020.

[22] Kate Saenko and Brian Kulis, "Adapting visual category models to new domains," in *ECCV*, 2010.

[23] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *CVPR*, 2017.

[24] Barbara Caputo, Henning Müller, Jesus Martinez-Gomez, Mauricio Villegas, Burak Acar, Novi Patricia, Neda Marvasti, Suzan Üsküdarlı, Roberto Paredes, Miguel Cazorla, Ismael Garcia-Varea, and Vicente Morell, "Imageclef 2014: Overview and analysis of the results," in *International Conference of the Cross-Language Evaluation Forum for European Languages*, 2014.

[25] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary Devito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," in *NeurIPS-Workshops*, 2017.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "Imagenet large scale visual recognition challenge," *IJCV*, 2015.

[28] Laurens Van Der Maaten and Hinton Geoffrey, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2605, pp. 2579–2605, 2008.