UNSUPERVISED DOMAIN ADAPTATION

Hongxuan Li
kongyiji_is_coding@163.com

ABSTRACT

The main idea of transfer learning is enhancing inter-domain transferability and intra-domain discriminability.

1 Why transfer learning can learn?

A domain is defined as a pair consisting of a distribution $\mathcal{D}$ generating the inputs $\boldsymbol{x}$ and a labeling function $f : \boldsymbol{x} \to [0, 1]$. We only consider two domains, a source domain and a target domain. We denote by $\langle \mathcal{D}_S, f_S \rangle$ the sourcec domain and $\langle \mathcal{D}_T, f_T \rangle$ the target domain. A hypothesis is a function $h : \boldsymbol{x} \to \{0, 1\}$. The probability according to the distribution $\mathcal{D}_S$ that a hypothesis disagrees with a labeling function f (which can also be a hypothesis) is defined as

$$\epsilon_S(h, f) = \mathrm{E}_{\boldsymbol{x}\sim\mathcal{D}_S}[|h(\boldsymbol{x}) - f(\boldsymbol{x})|]$$

The shorthand $\epsilon_S(h) = \epsilon_S(h, f_S)$ is applied to denote the source error of a hypothesis. A bounds on the target domain generalization performance of a classifier trained in the source domain is expected to be developed.

1.1 L_1 divergence

L_1 is a nature measurement of divergence for distributions:

$$d_1(\mathcal{D}, \mathcal{D}') = 2 \sup_{B\in\mathcal{B}} |\Pr_{\mathcal{D}}[B] - \Pr_{\mathcal{D}'}[B]| \tag{1}$$

where $\mathcal{B}$ is the set of measurable subsers under $\mathcal{D}$ and $\mathcal{D}'$. This measure is applied to make an initial bound on the target error of a classifier:

$$\begin{aligned}\epsilon_T(h) \leq & \epsilon_S(h) + d_1\left(\mathcal{D}_S, \mathcal{D}_T\right) \\ & + \min\left\{\mathrm{E}_{\mathcal{D}_S}\left[|f_S(\boldsymbol{x}) - f_T(\boldsymbol{x})|\right], \mathrm{E}_{\mathcal{D}_T}\left[|f_S(\boldsymbol{x}) - f_T(\boldsymbol{x})|\right]\right\}\end{aligned} \tag{2}$$

However, L_1 divergence $d_1\left(\mathcal{D}_S, \mathcal{D}_T\right)$ is bound to require arbitrarily large samples to detect the change even between distributions whose total variation distance is large. L_1 divergence between real-valued distributions cannot be computed from finite samples and therefore is not useful to us when investigating representations for domain adaptation on real-world data.

1.2 $\mathcal{H}$-divergence

The key part of $\mathcal{H}$-distance theory[1, 2] is that we do not need such a powerful measurement as L_1 divergence, and the $\mathcal{H}$-distance can be measured only with respect to functions in the hypothesis class. The $\mathcal{H}$-divergence between $\mathcal{D}$ and $\mathcal{D}'$ is defined as:

$$\begin{aligned} d_{\mathcal{H}}(\mathcal{D}, \mathcal{D}') &= 2 \sup_{h \in \mathcal{H}} |\Pr_{\mathcal{D}}[I(h)] - \Pr_{\mathcal{D}'}[I(h)]| \\ &\leq \hat{d_{\mathcal{H}}}(\mathcal{U}, \mathcal{U}') + 4\sqrt{\frac{d\log(2m) + \log(\frac{2}{\delta})}{m}} \end{aligned} \tag{3}$$

where $\mathcal{H}$ is a hypothesis space on $\boldsymbol{x}$ with VC dimension d and denote by $I(h)$ the set for which $h \in \mathcal{H}$ is the characteristic function;that is, $\boldsymbol{x} \in I(h) \Leftrightarrow h(\boldsymbol{x}) = 1$. $\mathcal{U}$ and $\mathcal{U}'$ are samples of size m from $\mathcal{D}$ and $\mathcal{D}'$ respectively and $\hat{d_{\mathcal{H}}}(\mathcal{U}, \mathcal{U}')$ is the empirical $\mathcal{H}$-divergence between samples, the for any $\delta \in (0, 1)$, with probability at least 1 - δ (over the choice of samples), the above inequality holds.

The uniform convergence theory tells us that if $\mathcal{H}$ has bounded capacity (*e.g.*, a finite VC-dimension d), the empirical divergence $\hat{d_{\mathcal{H}}}(\mathcal{U}, \mathcal{U}')$ convergences uniformly to true $\mathcal{H}$-divergence for hypothesis space $\mathcal{H}$. The inequality above demonstrates that the $\mathcal{H}$-divegence resolves the problems associated with the L_1 divergence:

- For hypothesis space $\mathcal{H}$ of finite VC dimension, the $\mathcal{H}$-divergence can be estimated with finite samples.
- $\mathcal{H}$ is never larger than L_1 divergence, and is in general smaller when $\mathcal{H}$ has finite VC dimension.

Recalling the relationship between sets and their characteristic functions, it should be clear that computing the $\mathcal{A}$-distance is closely related to learning a classifier. In fact they are identical. The set $A_h \in \mathcal{H}$ which maximizes the $\mathcal{H}$-distance between $\mathcal{D}_S$ and $\mathcal{D}_T$ has a characteristic h. The h is the classifier which achieves minimum error on binary classification probelm of discriminating between points generated by the two distributions. Then for a symmetric hypothesis class $\mathcal{H}$ (one where for every $h \in \mathcal{H}$, the inverse hypothesis 1-h is also in $\mathcal{H}$. Thus the binary function class is also a symmetric hypothesis class) and samples $\mathcal{U}, \mathcal{U}'$ of size m, the empirical $\mathcal{H}$-distance is

$$\begin{aligned} \hat{d}_{\mathcal{H}}(\mathcal{U}, \mathcal{U}') &= 2(\Pr_{\mathcal{U}}[I(h)] - \Pr_{\mathcal{U}'}[I(h)]) \\ &= 2\left(1 - \min_{h \in \mathcal{H}} \left[\frac{1}{m} \sum_{\boldsymbol{x}: h(\boldsymbol{x})=0} I[\boldsymbol{x} \in \mathcal{U}] + \frac{1}{m} \sum_{\boldsymbol{x}: h(\boldsymbol{x})=1} I[\boldsymbol{x} \in \mathcal{U}']\right]\right) \end{aligned} \tag{4}$$

It is a NP-hard problem to approximate the error of the optimal hyperplane classifier for arbitary distributions. Although the proxy for $\mathcal{A}$-distance approximated by minimzing convex upper bound with a linear classifier can not provide a valide upper bound on the errorit can provide a useful insights to the $\mathcal{A}$-distance.

1.3 $\mathcal{H}\Delta\mathcal{H}$-divergence

$\mathcal{H}\Delta\mathcal{H}$-divergence[1, 2] is a classifier induced divergence, which is defined as:

$$g \in \mathcal{H}\Delta\mathcal{H} \Leftrightarrow g(\boldsymbol{x}) = h(\boldsymbol{x}) \oplus h'(\boldsymbol{x}) \text{ for some } h, h' \in \mathcal{H}$$

$\mathcal{H}\Delta\mathcal{H}$ is symmetric difference hypothesis space where every hypothesis $g \in \mathcal{H}\Delta\mathcal{H}$ is the disagreements between two hypothesis in $\mathcal{H}$. Leveraging the triangle inequality for classification $\epsilon(f_1, f_2) \leq \epsilon(f_1, f_3) + \epsilon(f_2, f_3)$, we have:

$$\begin{aligned}
\epsilon_T(h) &\leq \epsilon_T(h^\star) + \epsilon_T(h, h^\star) \\
&\leq \epsilon_T(h^\star) + \epsilon_S(h, h^\star) + |\epsilon_T(h, h^\star) - \epsilon_S(h, h^\star)| \\
&\leq \epsilon_T(h^\star) + \epsilon_S(h, h^\star) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) \\
&\leq \epsilon_T(h^\star) + \epsilon_S(h) + \epsilon_S(h^\star) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) \\
&= \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \lambda \\
&\leq \epsilon_S(h) + \frac{1}{2} \hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + 4\sqrt{\frac{2d\log(2m') + \log(\frac{2}{\delta})}{m'}} + \lambda \\
&\leq \hat{\epsilon}_S(h) + \sqrt{\frac{4}{m}\left(d\log\frac{2em}{d} + \log\frac{4}{\delta}\right)} + \frac{1}{2} \hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + 4\sqrt{\frac{2d\log(2m') + \log(\frac{2}{\delta})}{m'}} + \lambda
\end{aligned} \tag{5}$$

where $h^\star$ is the ideal joint hypothesis which minimizes the combined error:

$$h^\star = \arg\min_{h \in \mathcal{H}} \epsilon_S(h) + \epsilon_T(h) \tag{6}$$

and λ is the the combined error of the ideal joint hypothesis:

$$\lambda = \epsilon_S(h^\star) + \epsilon_T(h^\star) \tag{7}$$

The last step bounds the true $\epsilon_S(h)$ with its empirical estimate $\hat{\epsilon}_S(h)$. Our final bound on the target error is in terms of the empirical source error, the empirical $\mathcal{H}\Delta\mathcal{H}$ (*e.g.* $\mathcal{A}$-distance) between unlabeled samples from the domains, and the combined error of the best single hypothesis for both domains. The $\mathcal{A}$-distance and empirical source error can be computed from finite samples of unlabeled data, allowing us to directly estimate the error of a source-trained classifier on the target domain.

2 How do transfer learning learn?

According Eq.(5), there are three terms need to be taken into consideration:

- $\hat{\epsilon}_S(h)$ - the empirical source classifier error
- $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T)$ - the empirical $\mathcal{A}$-distance
- λ - the combined error of the ideal joint hypothesis

$\hat{\epsilon}_S(h)$ is the disagreement between h and the source classifier f_S. Obviously this term can be small enough since we have access to labels of source samples. The key for transfer learning methods learning process is how to minimize $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T)$ and λ.

2.1 Global Domain Adaptation

Global domain adaptation methods often treat the the combined error of the ideal joint hypothesis λ as a constant, and focus on minimizing the $\mathcal{A}$-distance = $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T)$, which actually measures the global domain distribution discrepancy. Some global domain adaptation methods can be summarized as follows:

- Statistic moment matching - CORAL[3], DAN(MK-MMD)[4], JAN(JMMD)[5]
- Adversarial based - DANN[6]

For statistic moment matching, Maximum Mean Discrepancy(MMD) and Correlation Alignment(CORAL) are widely used two statistic metric in global domain adaptation. MMD was initially introduced as a test statistic for the

hypothesis testing on whether two sets of samples are generated by the same distribution. The CORAL aligns the second-order statistics(covariance matrices) across domains. The working flow is shown in Fig. 1.

As shown in Fig. 2, in adversarial training, a domain classifier is trained to tell whether the sample comes from source domain or target domain. The feature extractor is trained to minimize the classification loss and maximize the domain confusion loss. Thus, the learned feature representations can be indistinguishable across domains.

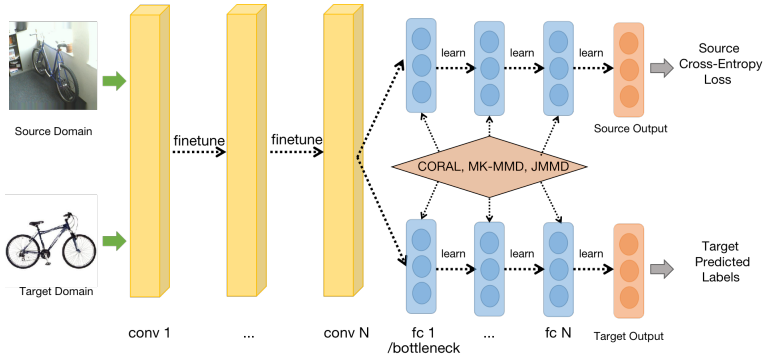


Figure 1: Domain-specific layers conduct statistic moment matching for domain-invariant feature representations.

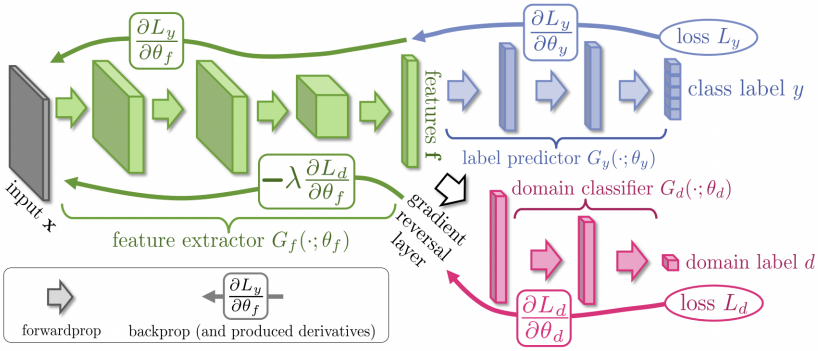


Figure 2: Adversarial global domain adaptation working flow.

2.2 Local Domain Adaptation

As show in Fig. 3, different from previous methods aligning the domains from a global view, local domain adaptation (*i.e.* subdomain adaptation adaptation) captures the fine-grained information for each class in target domain.

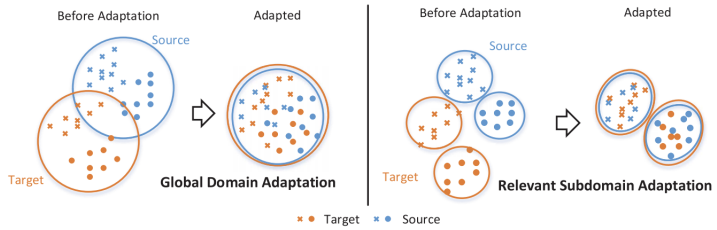


Figure 3: Comparisons between global domain adaptation and local domain adaptation.

The boosting for local domain adaptation starts with feature spectral analysis by [7] , which demystifies the transferability and discriminability of feature representations learned by DA methods.

Recalling that the term λ measures the discriminability of features especially in the target domain. The local domain adaptation mainly minimizes λ, since when the within-class samples are well aligned, the global domain discrepancy

(*i.e.* $\mathcal{A}$-distance = $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T)$) is also minimized. The upper bound for λ is usually optimized by local domain adaptation as follows:

$$\begin{aligned}
\lambda &= \min_{\forall h \in \mathcal{H}} \epsilon_{\mathcal{S}}\left(h, f_{\mathcal{S}}\right) + \epsilon_T(h, f_T) \\
&\leq \min_{\forall h \in \mathcal{H}} \epsilon_{\mathcal{S}}\left(h, f_{\mathcal{S}}\right) + \epsilon_T\left(h, f_{\mathcal{S}}\right) + \epsilon_T\left(f_{\mathcal{S}}, f_T\right) \\
&\leq \min_{\forall h \in \mathcal{H}} \epsilon_{\mathcal{S}}\left(h, f_{\mathcal{S}}\right) + \epsilon_T\left(h, f_{\hat{T}}\right) + \epsilon_T\left(f_{\mathcal{S}}, f_{\hat{T}}\right) + \epsilon_T\left(f_{\mathcal{S}}, f_{\hat{T}}\right) + \epsilon_T\left(f_T, f_{\hat{T}}\right) \\
&= \min_{\forall h \in \mathcal{H}} \epsilon_{\mathcal{S}}\left(h, f_{\mathcal{S}}\right) + \epsilon_T\left(h, f_{\hat{T}}\right) + 2\epsilon_T\left(f_{\mathcal{S}}, f_{\hat{T}}\right) + \epsilon_T\left(f_T, f_{\hat{T}}\right)
\end{aligned}$$

where $f_{\hat{T}}$ is the pseudo-labeling function for target domain. The pseudo labels and soft labels play a crucial role in transducing the dicriminative information from labeled source domain to the unlabeled target domain. There are four terms for local domain adaptation to handle:

- $\epsilon_{\mathcal{S}}\left(h, f_{\mathcal{S}}\right)$ - the source error of a hypothesis
- $\epsilon_T\left(h, f_{\hat{T}}\right)$ - the target error of a hypothesis with pseudo-labeling function
- $\epsilon_T\left(f_{\mathcal{S}}, f_{\hat{T}}\right)$ - the disagreement between the source labeling function and pseudo-labeling function
- $\epsilon_T\left(f_T, f_{\hat{T}}\right)$ - denotes the degree to which the target samples are falsely labeled on

The first term $\epsilon_{\mathcal{S}}\left(h, f_{\mathcal{S}}\right)$ can be minimized by training the classifier with labeled source samples. However, considering the second term $\epsilon_T\left(h, f_{\hat{T}}\right)$, only minimizing the first term can cause overfitting problem, where the optimal gap between $f_{\mathcal{S}}$ and $f_{\hat{T}}$ exists. The two terms can be solved by retarding the convergence speed of the source classification loss for better adaptation performance. For the third term$\epsilon_T\left(f_{\mathcal{S}}, f_{\hat{T}}\right)$, note that when the target samples in class c is well-aligned with the source samples in class c by local domain adaptation, the predicted results on target samples with source labeling function $f_{\mathcal{S}}$ can be the same as the pseudo target labeling function $f_{\hat{T}}$. The last term $\epsilon_T\left(f_T, f_{\hat{T}}\right)$ can be minimized by selecting more reliable pseudo-labeled samples in the target domain. Some local domain adaptation methods can be summarized as follows:

- MMD based methods - DSAN[8], CAN[9]
- Adversarial based methods - MADA[10], CDAN[11]
- Semantic alignment methods - MSTN[12], PFAN[13]
- Manifold alignment methods - DRMEA[14], GCAN[15]
- Normalization based methods - RSDA[16], DSBN[17]
- Self-pace based methods - SPCAN[18]

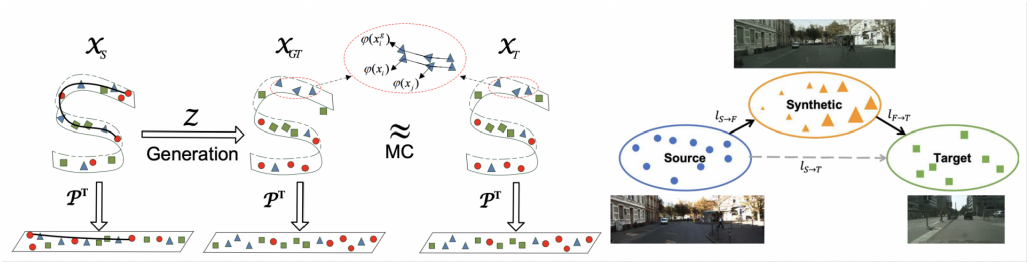
3 Domain Adaptation for Object Detection

In unrestricted object detection, it is usually assumed that the distribution of instances in both train (source domain) and test (target domain) set are identical. Unfortunately, this assumption is easily violated, and domain changes in object detection arise with variations in viewpoint, background, object appearance, scene type and illumination. As shown in Fig. 4, different weather conditions can cause domain shift.

As shown in Fig. 5(b), to tackle these problems, Hsu *et al.* [19] proposes a progressive learning strategy by generating an intermediate synthetic domain to gradually align the source and target domain. This assumption is similar with a traditional domain adaptation method proposed by Zhang *et al.* [20]. They all assume that if a new target domain can be generated from source domain, the domain shift can be eliminated. In [19], the synthetic domain was generated by GAN, where the image conditions can be differ from source domain, causing domain difference. The synthetic domain is seen as an intermediate state of target domain, and it is also unlabeled for training stage.

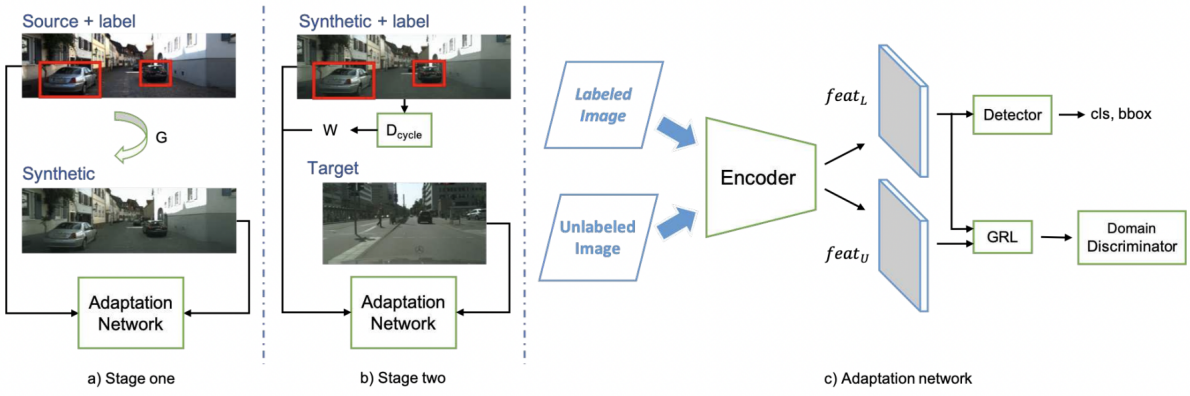
As shown in Fig. 6(c), the labeled image(source domain) and unlabeled image(synthetic or target domain) are input into the encoder to extract features. The *feat_L* is the source feature which is used to learn supervised object detection with the detector network (*e.g.* Faster R-CNN). Meanwhile, the synthetic or target domain feature *feat_U* together with *feat_L* are used to confuse the domain discriminator. The Gradient Reverse Layer (GRL) is firstly proposed by [6]. With the two training phases in Fig. 6(a)(b), the encoder network can extract domain-invariant features for the Detector for better object detection performance.

Figure 4: The train and test data can differ from sceneries, weather, lighting conditions and the image appearance with respect to the camera being used.



(a) MCTL illustration (b) PDA illustration

Figure 5: Illustration of MCTL and PDA.



a) Stage one b) Stage two c) Adaptation network

Figure 6: PDA working flow.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*. 2007.

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 2010.

[3] Baochen Sun and Kate Saenko. Deep CORAL: correlation alignment for deep domain adaptation. In *ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision*, 2016.

[4] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, 2015.

[5] Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning*, 2017.

[6] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, 2015.

[7] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International Conference on Machine Learning*, 2019.

[8] Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong, and Qing He. Deep subdomain adaptation network for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[9] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[10] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI Conference on Artificial Intelligence*, 2018.

[11] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*. 2018.

[12] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *International Conference on Machine Learning*, 2018.

[13] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[14] You-Wei Luo, Chuan-Xian Ren, Pengfei Ge, Ke kun Huang, and Yu-Feng Yu. Unsupervised domain adaptation via discriminative manifold embedding and alignment. 2020.

[15] Xinhong Ma, Tianzhu Zhang, and Changsheng Xu. GCAN: graph convolutional adversarial network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[16] Xiang Gu, Jian Sun, and Zongben Xu. Spherical space domain adaptation with robust pseudo-label loss. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[17] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *IEEE International Conference on Computer Vision*, 2019.

[18] Weichen Zhang, Dong Xu, Wanli Ouyang, and Wen Li. Self-paced collaborative and adversarial network for unsupervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[19] Han-Kai Hsu, Wei-Chih Hung, Hung-Yu Tseng, Chun-Han Yao, Yi-Hsuan Tsai, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[20] Lei Zhang, Shanshan Wang, Guang-Bin Huang, Wangmeng Zuo, Jian Yang, and David Zhang. Manifold criterion guided transfer learning via intermediate domain generation. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.