Adaptive Feature Norm for Unsupervised Subdomain Adaptation

Ashiq Imran and Vassilis Athitsos

September 27, 2021

Adaptive Feature Norm for Unsupervised Subdomain Adaptation

Ashiq Imran, and Vassilis Athitsos

Department of Computer Science and Engineering
University of Texas at Arlington, Arlington, TX, USA
ashiq.imran@mavs.uta.edu
athitsos@uta.edu

Abstract. In many real-world problems, obtaining labeled data for a specific machine learning task is expensive. Unsupervised Domain Adaptation (UDA) aims at learning a good predictive model for the target domain using labeled information from the source but only unlabeled samples from the target domain. Most of the previous methods tackle this issue with adversarial methods that contain several loss functions and converge slowly. Recently, subdomain adaptation, which focuses on nuances of the distribution of the relevant subdomains, is getting more and more attention in the UDA field. This paper proposes a technique that uses the adaptive feature norm with subdomain adaptation to boost the transfer gains. Subdomain adaptation can enhance the ability of deep adaptation networks by capturing the fine-grained features from each category. Additionally, we have incorporated an adaptive feature norm approach to increase transfer gains. Our method shows state-of-the-art results on the popular visual classification datasets, including Office-31, Office Home, and Image-CLEF datasets.

Keywords: Domain Adaptation · Transfer Learning · Object Recognition

1 Introduction

Deep Neural Networks have shown remarkable performance in various domains in the field of computer vision. To achieve good performance, they typically require a vast amount of labeled data. Training larger and deeper networks is complicated if the size of a dataset is small. Additionally, collecting well-annotated data is costly and time-consuming. A popular way to regularize these networks is to simply use a pre-trained model trained on a different dataset and use the model for the target dataset. However, if the data distribution between source and target domains is different, it may lead to adverse effects and hamper the generalization ability of the models [3]. Unsupervised Domain Adaptation (UDA) focuses on transferring knowledge from a labeled source domain to an unlabeled target domain, and a large amount of research tries to achieve this by exploring domain-invariant representations to bridge the gap. Traditional machine-learning paradigms, like supervised learning, tend to train models to predict the outcome for unseen data. These models do not necessarily optimize performance if there is enough difference between the test and training data [21]. According to Tzeng *et al.* [23], while generically trained deep networks have a reduced dataset bias, there still exists a domain shift between different datasets, and it is required to adapt the features appropriately. [1] suggests that a fair domain adaptation method should be based on features that are near similar for the source and target domains while reducing the prediction error in the source domain as much as possible. However, domain adaptation can have a domain shift problem. For example, the target domain may contain images from different imaging device (e.g. webcam vs. dslr camera), resulting in different styles in photos. This means the object recognition model trained from the source domain requires to be adapted to the target domain. Therefore, to reduce the domain shift problem, the two domains marginal distributions need to be as similar as possible. The primary goal of UDA is to learn domain-invariant feature representations that can reduce the domain shift. According to existing studies, domain-invariant representations can be captured through several methods, e.g., Maximum Mean Discrepancy [30, 10, 14], divergence-based methods [1, 18], correlation distance [20], etc. Addtionally, several adversarial based methods have been applied [5, 22, 8, 25, 19] to minimize an approximate domain discrepancy.

Recent studies have shown that, compared to shallow networks, deep networks can learn more transferable features for domain adaptation by extracting domain-invariant features [28, 10, 11, 20]. The main observation from the previous domain adaptation methods is that the domain classifier should be confused maximally so that the source classifier treats the samples from the target domain in a similar fashion. Additionally, most successful methods have come up with such ways that can make the domain classifier more confused. Most of the previous domain adaptation methods consider aligning the source and target distributions globally. We adapt a subdomain based approach to learn the domain transfer. A subdomain consists of samples within the same class. This method will lead to a scenario where all the data from different domains will be confused, and discriminative structures can be mixed up[31]. The main advantage of the subdomains over domains is the local domain shift instead of the global domain shift. Because of the local domain shift, the learners precisely may align the distribution of relevant subdomains within the same category in the source and target domains. An illustrative example of the difference between Domain Adaptation and Subdomain adaptation is depicted in Figure 1(a). After global domain adaptation, the resulting distributions of the two domains are quite similar, but the data in different subdomains are too adjacent to be correctly classified. The distributions of relevant subdomains can be aligned properly, hence the nuances of the information can be exploited for domain adaptation.

According to [26], larger norms enable more informative trasferability. Recent studies on the compression technique [27] support the above claim and suggest smaller norms contain less information during the inference. Inspired from the two studies as mentioned above, we incorporate the step-wise adaptive feature norm approach in subdomain space.

Xu *et al.* [26] demonstrates that progressively adapting the feature norms of two domains to a broad range of values can boost domain transfer. We present the local maximum mean discrepancy based method with adaptive progressively feature norm on subdomain space. For effective UDA, our goal is to endorse positive transfer and circumvent negative transfer.

In summary, the main contributions of our work are:

1. We propose an innovative stepwise adaptive feature norm-based approach for unsupervised subdomain adaptation. This approach employs to learn task-specific features in a progressive manner, which assists in aligning relevant subdomains in unsupervised scenarios.
2. We demonstrate a local MMD [31] based method with stepwise adaptive feature norm to achieve state-of-the-art results on Office-31, Office-Home, and ImageCLEF datasets.
3. We comapare our results with both adversarial and non-adversarial methods to show the efficacy of our work.

2 Related Work

Domain adaptation problem has been widely studied in the computer vision research community. Various methods have been employed to generalize the model across different domains by mitigating the domain shift problem. This section will discuss the relevant work in domain adaptation, subdomain adaptation, and maximum mean discrepancy.

2.1 Domain Adaptation

Domain adaptation can be a way to mitigate domain shift issues and reduce the effort of recollecting and retraining a model by transferring knowledge between tasks and domains. Domain adaptation can be defined as the task of training a model on labeled data from a source domain while minimizing test error on a target domain, where no labels for the target domain are available at training time. Several types of methods have been employed for unsupervised domain adaptation. Discrepancy based methods explore domain-invariant structures by reducing some specific statistic distances between the two domains. Maximum Mean Discrepancy (MMD) [2] has been adopted in many approaches [30, 10] for domain adaptation. It enables the model to learn transferable features by reducing the MMD of their kernel embedding. Some other methods extended MMD [12, 13] to measure the source and target data's joint distributions. In our case, we consider local MMD measures discrepancy of relevant subdomains between source and target domains. Adversarial DAs [5, 22, 19] are widely applied in this field. They involve a sub-network as the domain discriminator to distinguish features of alternate domains, whereas learners try to generate features that confuse the domain classifier.

2.2 Subdomain Adaptation

A significant amount of research for subdomain adaptation has been published recently. Multiadversarial domain adaptation (MADA) captures the multimode structures to enable fine-grained alignment of various data distributions [16]. CDAN [11] captures the adversarial domain adaptation on discriminative information to enable alignment of multimodal distributions. Moving the semantic transfer network (MSTN) [25] captures the semantic representation for unlabeled target samples by aligning the source and target centroid. Another method [9] creates multiple diverse feature spaces and aligns the source and target distributions in each of them separately while encouraging that alignments agree with each other with regard to the class predictions on the unlabeled target samples. All these methods have adopted adversarial loss. Compared to our work, we have adopted a discrepancy based strategy with stepwise adaptive feature norm approach which is more straightforward and can perform better than these previous methods.

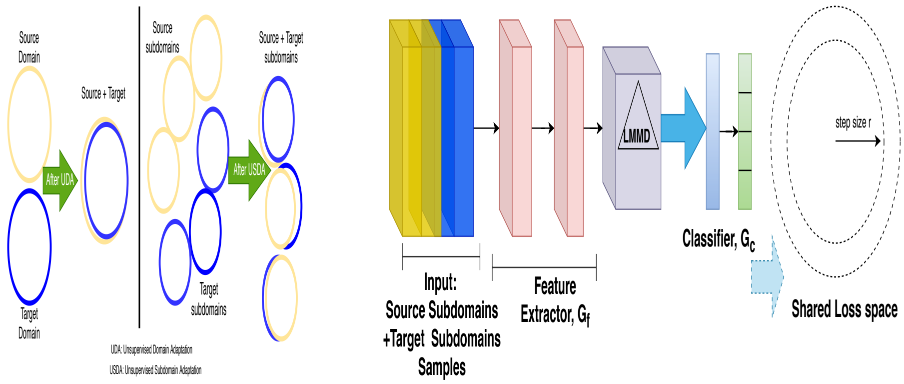
2.3 Maximum Mean Discrepancy

Among discrepancy based methods, MMD is one of the most popular metrics of training for domain invariant features. In Deep Adaptation Network (DAN) architecture [10], the authors train the first layers of the model commonly with the source and target domains; after that, they train individual task-specific layers while minimizing MMD between layers. Additionally, MMD has been extended by [12, 13]. However, most previous work considers global MMD measures to reduce discrepancies between the source and target samples. Our work is based on local MMD [31], which measures the discrepancy in relevant subdomains between the source and target domains.

Compared to the previous technique, we use a non-adversarial based subdomain adaptation method and incorporate adaptive feature norms within the subdomains to perform domain transfer. So, instead of just relying on a particular discrepancy metric, we take an additional approach as an adaptive feature norm. In our framework (Figure 1(b)), we have shown that a progressive feature-norm-based loss function in a shared subdomain space can boost the domain adaptation performance.

3 Method

In unsupervised domain adaptation, we are given a source domain $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ of n_s labeled examples and a target domain $D_t = \{x_j^t\}_{j=1}^{n_t}$ of unlabeled examples. The source domain and target domain are sampled from joint distributions $P(X^s, Y^s)$ and $Q(X^t, Y^t)$ respectively, where $P \neq Q$. The goal of our method is to develop a deep network architecture, that contains transfer features $f = G_f(x)$ and adaptive classifier $y = G_c(f)$. This model will minimize the shift in joint distribution across relevant subdomains and learns transferable representations simultaneously.



(a) Illustrative example of Subdomain Adaptation

(b) Architecture of Adaptive feature norm on unsupervised subdomain adapatation

Fig. 1: Domain Adaptation vs Subdomain Adaptation and Architecture of our proposed method.

The formal representation for unsupervised domain adaptation is as follows.

$$\min_f \frac{1}{n_s} \sum_{i=1}^{n_s} J(f(X_i^s), y_i^s) + \lambda \hat{d}(p, q) \tag{1}$$

where $J()$ is the cross-entropy loss function (classification loss) and $\hat{d}()$ is domain adaptation loss. $\lambda > 0$ is the trade-off parameter of the domain adaptation loss and the classification loss.

This representation covers the global source and target domain without taking into account the relevant information between subdomains within the same category between the source and target domains. Nevertheless, the global alignment may not capture the nuances among subdomains. This may lead to domain shift issue as well. The subdomain information can exploit the relationship between different domains. The formal representation of the loss of subdomain adaptation can be

$$\min_f \frac{1}{n_s} \sum_{i=1}^{n_s} J(f(X_i^s), y_i^s) + \lambda \mathbf{E}_c[\hat{d}(p^{(c)}, q^{(c)}])\tag{2}$$

where $\mathbf{E}_c[.]$ is a mathematical expectation of the class.

3.1 Local Maximum Mean Discrepancy

We have used local MMD as the baseline architecture. It was proposed by [31] to align distributions of the relevant subdomains.

$$d_H(p, q) = \mathbf{E}_c||\mathbf{E}_{p^{(c)}}[\phi(x^s)] - \mathbf{E}_{q^{(c)}}[\phi(x^t)]||_H^2 \tag{3}$$

where x^s and x^t are the instances in D_s and D_t, and $p^{(c)}$ and $q^{(c)}$ are the distributions of $D_s^{(c)}$ and $D_t^{(c)}$, respectively. The equation (3) can measure class by class difference of the relevant subdomains. Additionally, this can be used to align the subdomains within the target domain with those in the source domain. Since we have an assumption that each sample belongs to each class according to weight w^c, we use an unbiased estimator of equation (3) as

$$d_H(p,q) = \frac{1}{C}\sum_{c=1}^{C} || \sum_{x_i^s \in D_s} w_i^{sc}\phi(x_i^s) - \sum_{x_j^t \in D_t} w_j^{tc}\phi(x_j^t)||_H^2 \tag{4}$$

where w_i^{sc} and w_j^{tc} represent the weight of x_i^s and x_j^t belonging to class c, respectively. The sum of weights are both equal to one. We can formulate w_i^c for the sample x_i as

$$w_i^c = \frac{y_{ic}}{\sum_{(x_j,y_j \in D} y_{jc}} \tag{5}$$

where y_{ic} is the cth entry of vector y_i. For source domain, we use the ground truth y_i^s as a one-hot vector to calculate w_i^c for each sample. But, for target domain, we use the probability of assigning x_i^t to each of the classes. we can not use the formula of equation (4) directly. The output of the deep neural network is a probability distribution. We use that probability distribution which characterizes the probability of assigning samples to the classes for each target sample. Then, we can calculate w_j^{tc}. Finally, we can calculate equation (4).

3.2 Architecture

Standard domain adaptation considers two domains to share a similar label space. In our framework, the input consists of subdomains from the source and target domains. We have a backbone network G_f, which denotes the feature extraction module. Classifier G_c is the task-specific classifier. We apply the feature norm adaptation along with the local MMD based method to optimize the source classification loss during each iteration.

In each iteration, each individual sample's feature norm is getting added a small but progressive step size of r. This way, if any target samples are far way from the small norm region, after the domain adaptation step, it could be classified correctly in an automatic manner. Figure 1(b) demonstrates the architecture of our approach.

3.3 Adaptive Feature Norm Loss

One of the major bottlenecks that we observe is smaller feature norm of the source and target samples that can lead to poor transfer gains[26]. Inspired from them, we extend the idea into subdomain spaces. We keep a parameter r, which progressively modifies the mean feature norm in each iteration. Instead of having a fixed feature norm, we consider a moving parameter which changes the mean feature norm. This method has been unexplored for the subdomain adaptation case. This loss value impacts the target samples to be correctly classified without additional supervision. This variant impacts positively towards learning task-specific features in a continuous manner. We propose

$$\begin{aligned}\hat{d}_H(p,q) = \mathbf{E}_c||\mathbf{E}_{p(c)}[\phi(x^s)] - \mathbf{E}_{q(c)}[\phi(x^t)]||^2_H \\ +\mathbf{E}_c||\mathbf{E}_{p(c)}[(h(x_i;\theta_0) + \Delta r), h(x_i;\theta)]||\end{aligned} \tag{6}$$

where $h(x) = ||.||_2 \cdot G_f \cdot G_c(x)$, where θ_0 and θ are model parameters of last and current iterations. The effectiveness of this model parameter enables the optimization process fetching more informative features with larger norms.

4 Experiment

We evaluate our technique on three popular object recognition datasets, including Office-31, Office-Home, and ImageCLEF-DA. The code will be published in future.

4.1 Dataset

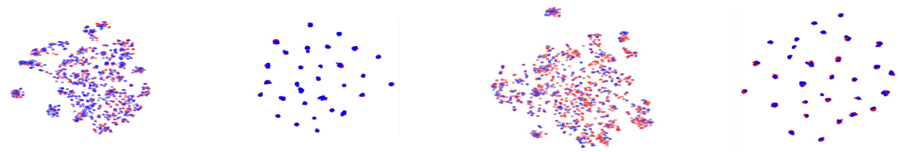
We present a detailed overview of the datasets that we use for our experiments.

Office-31 [17] is a very popular dataset for benchmarking domain adaptation. This dataset contains more than 4000 images in 31 classes collected from three different domains: Amazon (A), which consists of images downloaded from amazon.com, and Webcam(W), and DSLR(D), which comprises of images taken by web camera and digital SLR camera with various photographic settings, respectively. Table 1 reports the performance of our method compared with other works on Office-31 dataset.

Office-Home [24] is another challenging dataset for unsupervised domain adaptation. This dataset contains four domains: Art(Ar), Clipart(CI), Product(Pr), and Real-World(Rw). Each domain has common 65 categories. The Art domain contains the artistic description of objects including painting, sketches etc. The Clipart are the collection of clipart images. In the Product, domain images have no background. The Real-Work domain contains an object taken from a regular camera. In Table 2, we compare our result with previous methods on Office-Home dataset.

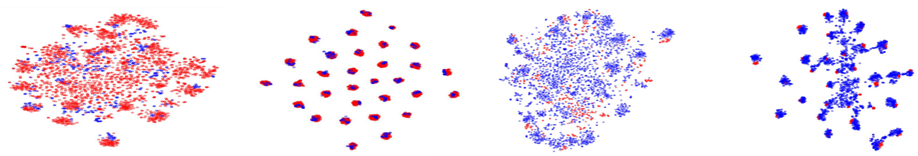
ImageCLEF-DA[1] contains three domains: Caltech-256(C), ILSVRC 2012 (I), and Pascal-VOC 2012 (P). Each domain has 12 common classes, and each class has 50 samples. In total, there are 600 images in each domain. Table 3 reports the performance of our method with previous methods on ImageCLEF dataset.

[1] http://imageclef.org/2014/adaptation



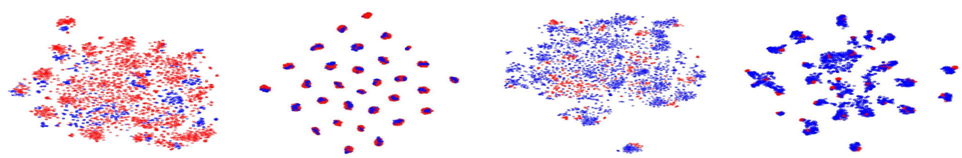
(a) $D \to W$ (Before Adaptation) (b) $D \to W$ (After Adaptation) (c) $W \to D$ (Before Adaptation) (d) $W \to D$ (After Adaptation)

Fig. 2: t-SNE feature visualization from DSLR (Red) to Webcam (Blue) ((a) & (b)) and from Webcam (Red) to DSLR (Blue) ((c) & (d)) on Office-31 dataset.



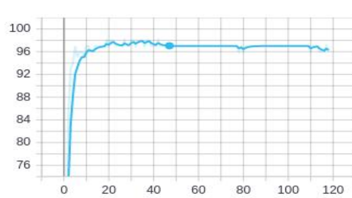
(a) $A \to D$ (Before Adaptation) (b) $A \to D$ (After Adaptation) (c) $D \to A$ (Before Adaptation) (d) $D \to A$ (After Adaptation)

Fig. 3: t-SNE feature visualization from Amazon (Red) to DSLR (Blue) ((a) & (b)) and from DSLR (Red) to Amazon (Blue) ((c) & (d)) on Office-31 dataset.

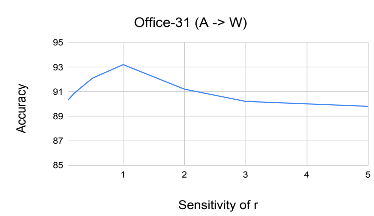


(a) $A \to W$ (Before Adaptation) (b) $A \to W$ (After Adaptation) (c) $W \to A$ (Before Adaptation) (d) $W \to A$ (After Adaptation)

Fig. 4: t-SNE feature visualization from Amazon(Red) to Webcam (Blue) ((a) & (b)) and from Webcam(Red) to Amazon (Blue) ((c) & (d)) on Office-31 dataset.



(a) Convergence Test on Webcam to DSLR on Office-31 dataset.



(b) parameter sensitivity of Δr on Office-31 (Amazon→Webcam) dataset.

Fig. 5: Convergence test on task Webcam to DSLR and parameter sensitivity test on Webcam to Amazon on Office-31 dataset.

4.2 Setup

In our experiment, we used the open-source implementation of a popular deep learning framework, Pytorch [15], to train the models on multiple Nvidia Geforce GTX 1080Ti GPUs. The machine has Intel Core-i7-5930k CPU@ 3.50GHz x 12 processors with 64GB of memory. For the visual classification task, we applied ResNet50 [7] as the backbone network. For comparison, all the baseline models use identical architecture. We fine-tune all the layers except classifier layers from ImageNet[4] pre-trained models and train the fully connected layers for classification through backward-propagation. We set the learning rate to 0.01, batch size to 32, we use stochastic gradient descent (SGD) with a momentum of 0.9, the learning rate is getting changed during SGD using the formula: $lr_{new} = lr_{old}/(1 + \alpha(epoch - 1)/epoch)^{\beta}$, where $\alpha = 10$, and $\beta = 0.75$. For the adaptation feature norm loss, we observe the embedding size of task-specific features played a major role in norm computation. We found $r = 1$ and $\lambda = 0.05$ provide the best result. the highest value of r is to $R = 5$, so it progresses each step r incrementally. The average classification accuracy and error are reported over three random repeats.

5 Results and Discussion

We use our proposed approach for unsupervised subdomain adaptation. We use the protocol to utilize source data with labels and target data without labels. The visual classification results of Office-31, Office-Home, and ImageCLEF-DA are promising. Our method outperforms previous methods on these datasets. Some of the observations from our experiments are:

- Comparing our proposed approach with the global domain adaptation methods and several adversarial subdomain adaptation methods [11, 16, 25], these methods are more complex compared to our approach. The reason is most of the methods use the adversarial loss function, and don't consider the kernel mean embeddings between source and target subdomains, and has more number of parameters. Moreover, our method achieves better accuracy compared with other methods in all three datasets.
- The t-SNE feature visualization on the transfer task between DSLR and Webcam, Amazon and DSLR, Webcam and Amazon is presented in Figure 2, Figure 3, Figure 4 respectively. Source samples are colored as red and target samples are colored as blue in each figure from 2 to 4. This visualization shows the effectiveness of subdomain adaptation.
- We conducted convergence test (Figure 5(a)) on task webcam to dslr and further case studies to investigate the sensitivity (on task Amazon to Webcam) of parameter Δr in Figure 5(b). The accuracy increases upto $\Delta r = 1$ than gradually decreases.

Most of these methods do not consider the subdomain relationship, which effectively captures nuances for each class. Additionally, we incorporate adaptive feature norm loss inside of subdomain distributions. It contributes to apprehend more fine-grained information. The results validate the efficacy of our approach.

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet[7]	68.4 ± 0.5	96.7 ± 0.5	99.3 ± 0.1	68.9 ± 0.2	62.5 ± 0.3	60.7 ± 0.3	76.1
DDC[23]	75.8 ± 0.2	95.0 ± 0.2	98.2 ± 0.1	77.5 ± 0.3	67.4 ± 0.3	64.0 ± 0.5	79.7
D-CORAL[20]	77.7 ± 0.3	97.6 ± 0.2	99.5 ± 0.1	81.1 ± 0.4	64.6 ± 0.3	64.0 ± 0.4	80.8
DAN[10]	83.8 ± 0.4	96.8 ± 0.2	99.5 ± 0.1	78.4 ± 0.2	66.7 ± 0.3	62.7 ± 0.2	81.3
DANN[6]	82.0 ± 0.4	96.8 ± 0.2	99.1 ± 0.1	79.7 ± 0.4	68.2 ± 0.4	67.4 ± 0.5	82.2
ADDA[22]	86.2 ± 0.5	96.2 ± 0.3	98.4 ± 0.3	77.8 ± 0.3	69.5 ± 0.4	68.9 ± 0.5	82.9
JAN[13]	85.4 ± 0.3	97.4 ± 0.2	99.8 ± 0.2	84.7 ± 0.3	68.6 ± 0.3	70.0 ± 0.4	84.3
MADA[16]	90.0 ± 0.1	97.4 ± 0.1	99.6 ± 0.1	87.8 ± 0.2	70.3 ± 0.3	66.4 ± 0.3	85.2
CDAN[11]	93.1 ± 0.2	98.2 ± 0.2	100 ± 0	89.8 ± 0.3	70.1 ± 0.4	68.0 ± 0.4	86.6
iCAN[29]	92.5 ± 0.2	98.8 ± 0.1	100 ± 0	90.1 ± 0.1	72.1 ± 0.2	69.9 ± 0.1	87.2
CDAN + E[11]	94.1 ± 0.1	98.6 ± 0.1	100 ± 0	92.9 ± 0.2	73.5 ± 0.5	69.3 ± 0.3	87.7
DSAN[31]	93.4 ± 0.2	98.3 ± 0.1	100 ± 0	90.2 ± 0.7	73.5 ± 0.5	74.8 ± 0.4	88.2
Ours	93.2 ± 0.2	98.7 ± 0.2	100 ± 0	90.1 ± 0.2	75.1 ± 0.3	72.8 ± 0.4	**88.5**

Table 1: Accuracy Comparison of Unsupervised Domain Adaptation on Office-31 Dataset

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
ResNet[7]	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[10]	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[6]	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[13]	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]	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
CDAN + E[11]	50.7	70.8	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
DSAN[31]	54.4	70.8	75.4	60.4	67.8	68.0	62.6	55.9	78.5	73.8	60.6	83.1	67.5
Ours	55.0	71.0	75.3	61.1	69.4	68.0	61.4	55	78	72.9	60.0	83.6	**67.7**

Table 2: Accuracy Comparison of Unsupervised Domain Adaptation on Office-Home Dataset

Method	I→P	P→I	I→C	C→I	C→P	P→C	Avg
ResNet[7]	74.8 ± 0.3	83.9 ± 0.1	91.5 ± 0.3	78.0 ± 0.2	65.5 ± 0.3	91.2 ± 0.3	80.7
DDC[23]	74.6 ± 0.3	85.7 ± 0.8	91.1 ± 0.3	82.3 ± 0.7	68.3 ± 0.4	88.8 ± 0.2	81.8
DAN[10]	75.0 ± 0.4	86.2 ± 0.2	93.3 ± 0.2	84.1 ± 0.4	69.8 ± 0.4	91.3 ± 0.4	83.3
DANN[6]	75.0 ± 0.4	86.0 ± 0.3	96.2 ± 0.4	87.0 ± 0.5	74.3 ± 0.5	91.5 ± 0.6	85.0
D-CORAL[20]	76.9 ± 0.2	88.5 ± 0.3	93.6 ± 0.3	86.8 ± 0.6	74.0 ± 0.3	91.6 ± 0.3	85.2
JAN[13]	76.8 ± 0.4	88.0 ± 0.2	94.7 ± 0.2	89.5 ± 0.3	74.2 ± 0.3	91.7 ± 0.3	85.8
MADA[16]	75.0 ± 0.3	87.9 ± 0.2	96.0 ± 0.3	88.8 ± 0.3	75.2 ± 0.2	92.2 ± 0.3	85.8
CDAN[11]	76.7 ± 0.3	90.6 ± 0.3	97.0 ± 0.4	90.5.8 ± 0.4	74.5 ± 0.3	93.5 ± 0.4	87.1
iCAN[29]	79.5 ± 0.1	89.7 ± 0.1	94.6 ± 0.2	89.9 ± 0.4	78.5 ± 0.1	92.0 ± 0.1	87.4
DSAN[31]	80.2 ± 0.2	93.3 ± 0.4	97.2 ± 0.3	93.8 ± 0.2	80.8 ± 0.4	95.9 ± 0.4	90.1
Ours	79.8 ± 0.2	93.5 ± 0.2	98.1 ± 0.2	94.4 ± 0.2	79.8 ± 0.1	96.3 ± 0.2	**90.4**

Table 3: Accuracy Comparison of Unsupervised Domain Adaptation on Image-CLEF Dataset

6 Conclusion

In this work, we have proposed an innovative UDA approach, which incorporates local mean distributed discrepancy measure(LMMD) with adaptive feature norm on subdomain adaptation. Our method can boost the transfer gains more and precisely align the distributions of related subdomains within the source and target domains' relevant category. Extensive experiments are performed on three of the most popular datasets for domain adaptation. Our results show the method's effectiveness, implying that task-specific features with larger norms are more transferable on subdomain adaptation.

References

1. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine learning **79**(1-2), 151–175 (2010)
2. Borgwardt, K.M., Gretton, A., Rasch, M.J., Kriegel, H.P., Schölkopf, B., Smola, A.J.: Integrating structured biological data by kernel maximum mean discrepancy. Bioinformatics **22**(14), e49–e57 (2006)
3. Candela, J.Q., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset shift in machine learning. The MIT Press **1**, 5 (2009)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
5. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International conference on machine learning. pp. 1180–1189. PMLR (2015)
6. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The Journal of Machine Learning Research **17**(1), 2096–2030 (2016)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
8. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: International conference on machine learning. pp. 1989–1998. PMLR (2018)
9. Kumar, A., Sattigeri, P., Wadhawan, K., Karlinsky, L., Feris, R., Freeman, W.T., Wornell, G.: Co-regularized alignment for unsupervised domain adaptation. arXiv preprint arXiv:1811.05443 (2018)
10. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: International conference on machine learning. pp. 97–105. PMLR (2015)
11. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems. pp. 1640–1650 (2018)
12. Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S.: Transfer feature learning with joint distribution adaptation. In: Proceedings of the IEEE international conference on computer vision. pp. 2200–2207 (2013)
13. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: International conference on machine learning. pp. 2208–2217. PMLR (2017)

14. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks **22**(2), 199–210 (2010)
15. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
16. Pei, Z., Cao, Z., Long, M., Wang, J.: Multi-adversarial domain adaptation. In: Proceedings of the IEEE conference on AAAI. pp. 3934–3941 (2018)
17. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: European conference on computer vision. pp. 213–226. Springer (2010)
18. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3723–3732 (2018)
19. Sankaranarayanan, S., Balaji, Y., Castillo, C.D., Chellappa, R.: Generate to adapt: Aligning domains using generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8503–8512 (2018)
20. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: European conference on computer vision. pp. 443–450. Springer (2016)
21. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR 2011. pp. 1521–1528. IEEE (2011)
22. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7167–7176 (2017)
23. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014)
24. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5018–5027 (2017)
25. Xie, S., Zheng, Z., Chen, L., Chen, C.: Learning semantic representations for unsupervised domain adaptation. In: International Conference on Machine Learning. pp. 5423–5432 (2018)
26. Xu, R., Li, G., Yang, J., Lin, L.: Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1426–1435 (2019)
27. Ye, J., Lu, X., Lin, Z., Wang, J.Z.: Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers. arXiv preprint arXiv:1802.00124 (2018)
28. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in neural information processing systems. pp. 3320–3328 (2014)
29. Zhang, W., Ouyang, W., Li, W., Xu, D.: Collaborative and adversarial network for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3801–3809 (2018)
30. Zhu, Y., Zhuang, F., Wang, D.: Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 5989–5996 (2019)
31. Zhu, Y., Zhuang, F., Wang, J., Ke, G., Chen, J., Bian, J., Xiong, H., He, Q.: Deep subdomain adaptation network for image classification. IEEE Transactions on Neural Networks and Learning Systems (2020)