Multi-Partition Feature Alignment Network for Unsupervised Domain Adaptation

Sanatan Sukhija[*‡], Srenivas Varadarajan [†], Narayanan C. Krishnan [*§], Sujit Rai [*¶]
[*] *Indian Institute of Technology Ropar, India*
[†] *Intel, India*
[‡]sanatan@iitrpr.ac.in, [†]emailsreni24@gmail.com, [§]ckn@iitrpr.ac.in, [¶]2017csm1006@iitrpr.ac.in

Abstract—**In this paper, we present a novel unsupervised domain adaptation framework, Multi-Partition Feature Alignment Network, that learns a deep neural model for the target domain without the need for any supervision. Recent leading approaches for unsupervised domain adaptation are based on adversarial alignment. Aligning the global distribution of the domain representations via adversarial training does not guarantee the class-wise distribution alignment. The proposed approach is built on adversarial learning with the focus on carefully aligning class-wise domain representations. Our algorithm utilizes the pseudo-labels (the predicted labels) of the target features to stimulate class-wise alignment. As the pseudo-labels of individual target features can be erroneous, instead of iteratively aligning individual target samples, the proposed framework introduces a generic class-specific multi-partition alignment procedure that enables superior class-discriminative alignment of domain representations. The competitive performance of the proposed framework against state-of-the-art approaches over a wide variety of visual recognition tasks, namely, the digits classification task and the object recognition task, validates its effectiveness for unsupervised domain adaptation.**

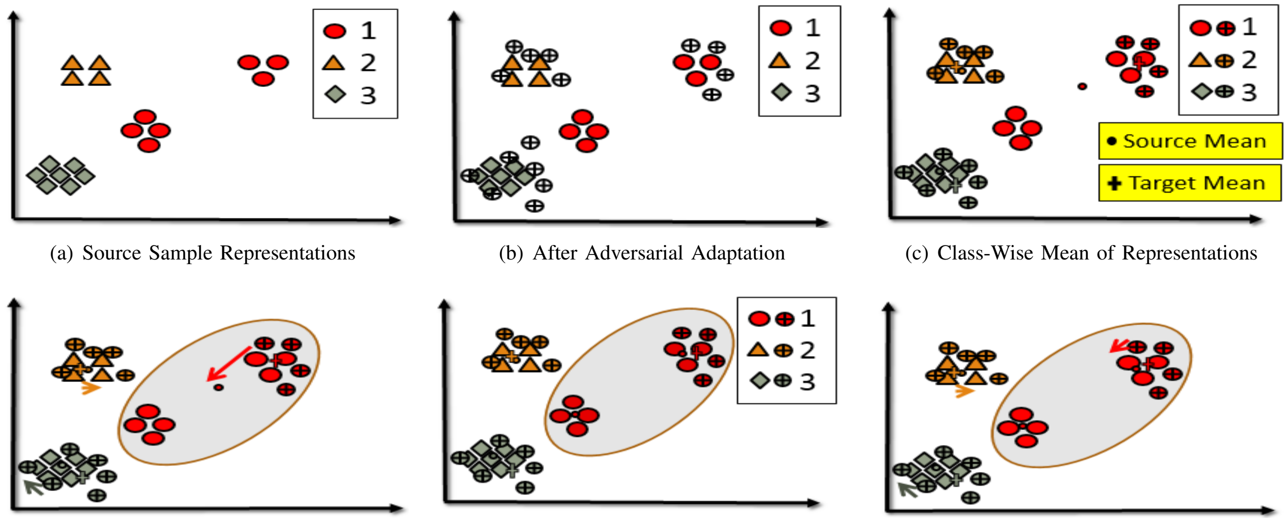
I. Introduction

With deep neural networks, the key to success for supervised visual recognition tasks is the availability of plentiful labeled examples. Often, for many real-world problems, the quantity of available labeled data is scarce. Even if unlabeled data is available, manually labeling the training data demands domain expertise and is a laborious task. Leveraging labeled samples from existing auxiliary domains (often termed as the source domains) can help to learn the desired task for the domain of interest (often referred to as the target domain). However, when the well-trained source model is evaluated on target examples, the performance suffers due to the distribution differences between the domains. The performance of a well-trained classifier in a new domain depends on two factors [1], 1) the performance in its domain and 2) the discrepancy between the two domains. The discrepancy between the domains, popularly known as domain shift or dataset shift, calls for domain adaptation. In this paper, we propose a deep adaptation framework that learns an efficient deep neural model for the target domain where no labeled data is available. This setting is commonly known as Unsupervised Domain Adaptation (UDA).

A common theme among recent promising deep UDA approaches is the use of adversarial training to align the feature distributions of the source and target domain. Based on the Generative Adversarial Network (GAN) loss, the adversarial adaptation procedure trains two competing networks - a feature generator network (one network each for the source and the target domain) and a domain classifier network. Here, the domain classifier (also known as the discriminator network) distinguishes the representations of the source and the target domain. In general, the source feature generator learns the class-discriminative features from the source labeled data whereas the target feature generator network is adversarially trained to confuse the domain classifier by generating source-like representations from the target samples. Learning domain invariant features with unsupervised adversarial adaptation only aligns the marginals distributions of the domains [21] and does not guarantee the class-specific alignment of the domain representations. Matching the class-wise distribution involves bringing the similarly labeled source and target representations closer to each other. As the target labels are not available in the unsupervised setting, some of the recent deep UDA approaches [15], [18], [26], [24], [3] utilize the pseudo-labels (the predicted labels for the target samples) as the categorical information for the target domain. However, the predicted labels are not guaranteed to be correct (even if confidently predicted). Consequently, progressive class-wise alignment of individual pseudo-labeled target samples leads to error accumulation and thereby, limits the transfer performance. Therefore, instead of aligning individual target samples, two recent approaches [24], [3] align the class-wise mean of the source and the target domain representations. Aligning the means limits the effect of noise as it alleviates the bias from incorrect pseudo-labeled target samples and has shown significant improvement over the adversarial adaptation baseline.

One explicit shortcoming with these approaches is that during the class-wise alignment of representations, there may exist several partitions for every label in the feature space that are distant from each other. Hence, only aligning the individual class-wise means would be ill-suited to match the conditional distribution of the domains. Figure 1 depicts a visual comparison of the learned representations with class-wise mean-alignment [24], [3] approaches versus the proposed approach. Consider a simple scenario, where for the red class (refer Figure 1 (d)), there are two distant compact partitions of the source representations and one dense partition formed by the target representations. Here, bringing the target mean closer to the source mean will lack in the good congruence

978-1-7281-6926-2/20/$31.00 ©2020 IEEE



(a) Source Sample Representations (b) After Adversarial Adaptation (c) Class-Wise Mean of Representations

(d) Class-Wise Mean Alignment (e) Class-Wise Partitioning of Representations (f) Multi-Partition Centroid Alignment

Fig. 1: (Best viewed in color) The given example contains 3 classes where each class is denoted with a different color. Figure (a) shows the 2D representation of the source samples. Figure (b) depicts the adversarially aligned target sample representations (denoted with ⊕) with source sample representations. In figure (c), the color of a target sample representation corresponds to the class-label obtained from the classifier (the predicted label, also known as the pseudo-label). The colored bullet (●) and plus (✚) symbols represent the individual class-mean of the source samples and pseudo-labeled target samples respectively. Figure (d) depicts the class-wise mean alignment of target representations with fixed source representations. The direction of the arrow indicates the course of the pseudo-labeled target representations (the target mean) towards the source mean having the same label whereas the length of the arrow represents the magnitude of movement. In contrast to aligning the mean for each class, the proposed approach aligns a target partition to its nearest source partition of the same class (refer Figure (f)). Figure (e) depicts the individual centroids of the obtained partitions (clusters) for each class in the source and target domain.

of the conditional alignment of the domain representations. So, considering that the domain representations can form multiple compact partitions (for every label), we propose a simple generic solution for effectively aligning the representations. The proposed adversarial adaptation framework, Multi-Partition Feature Alignment Network (MPFAN), overcomes this weakness by considering per-label partitions to align the class-wise distributions (refer Figure 1 (f)). We introduce the "centroid loss" that brings a refined pseudo-labeled target partition (cluster) closer to the nearest source partition with the same label. Moreover, inspired by the contrastive loss [5], we keep the partitions with dissimilar class-labels distant from each other. Besides, as a final step, we use an ensemble-based strategy for correcting the predicted labels of the target samples.

sary notations. Section **??** briefly describes the existing approaches that are related to the proposed framework. The subsequent section describes the proposed framework in detail. A brief description about the datasets and the transfer approaches for performance comparison is given in Section **??**. Section **??** presents the experimental results along with key insights from ablation studies.

A. Problem Statement

We are given a source domain D_s with n_s labeled examples $\{x_s^i, y_s^i\}_{i=1}^{n_s}$ derived from a joint probability distribution $P_1(X_s, Y_s)$ and a target domain D_t with n_t unlabeled target examples $\{x_t^j\}_{j=1}^{n_t}$ derived from a joint probability distribution $P_2(X_t, Y_t)$. Here, X_s and X_t denote the source and target feature distributions respectively. The two domains share the same categories i.e. $Y_s = Y_t$. Let the total number of categories be K numbered from $\{1, 2 \cdots K\}$. With $P_1 \neq P_2$, the goal is to learn a model that can correctly predict the labels of unlabeled target examples $\{y_t^j\}_{j=1}^{n_t}$ by leveraging the labeled samples from D_s.

II. Related Work

In this section, we briefly describe the most relevant methods to our proposed framework which comes under the category of deep UDA methods for the image classification task. Most of the prior work can be summarized as *Latent Feature Transformation (LFT) approaches* where the goal is to minimize the domain discrepancy in a shared feature space. This is achieved by learning shared hidden representations while matching the distributions between the source and target data via some distance metric. Prior work has used Maximum Mean Discrepancy [8], [23] and correlation distance [20] to reduce the domain differences in the common feature space. Recent state-of-the-art UDA frameworks [7], [22], [9], [24], [17], [3] are based on adversarial adaptation. Motivated by parameter sharing frameworks [19], some adversarial adaptation frameworks [4], [16] share the parameters of the hidden layers to learn a joint distribution from the source and target images.

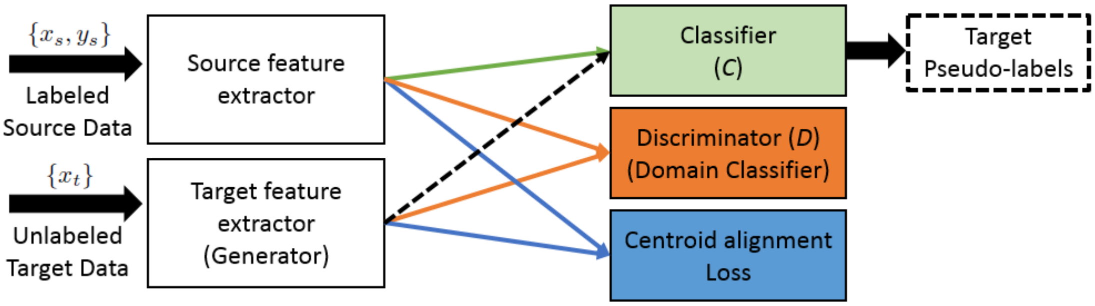


Fig. 2: Depicts the proposed adversarial alignment framework. The pseudo-labels for the target images are obtained from the classifier. There are three alternating steps in the training procedure. In the first step, we train the classifier with a randomly generated mini-batch of labeled source examples. Thereafter, in the second step, the domain representations are adversarially aligned by fooling the discriminator. The second step involves updates to the generator and the discriminator. The third step aligns the centroids of the target partitions with the source partitions. These three steps are repeated until convergence.

The aforementioned adaptation approaches ignore the conditional distribution differences between the source and target data while aligning the representations. Two recent adversarial adaptation frameworks, namely, 1) Moving Semantic Transfer Network (MSTN) [24] and Progressive Feature Alignment Network (PFAN) [3] utilize the pseudo-labels of the target samples to improve the class-wise alignment of the source and target representations. Instead of bringing individual pseudo-labeled target samples closer to similar source samples, both MSTN and PFAN align the class-wise means of the source and the target representations to match the conditional distributions. As most target samples are expected to be correctly predicted, the contribution of the false pseudo-labeled target samples towards the mean is unlikely to have a radical impact on overall alignment. To further reduce the impact of incorrectly predicted target samples, PFAN introduced a pruning strategy for the pseudo-labeled target samples. A pseudo-labeled target sample contributes to its class-wise mean when its cosine similarity score concerning the source mean (computed from the source samples having the same label) lies above a certain threshold. Nevertheless, during the class-wise alignment, both MSTN and PFAN do not consider that the representations of the samples having the same label can be scattered in the feature space forming multiple partitions that can be distant from each other.

Overall, the proposed framework of multi-partition centroid alignment can be considered as a generic framework for unsupervised domain adaptation where both MSTN and PFAN can be viewed as a particular case of ours with the number of partitions per label being set to 1. However, there are three key differences, namely, 1) In contrast to grouping the representations of all the samples belonging to the same class to compute the mean, the proposed adversarial adaptation framework takes into account that samples from the same class can form multiple distant partitions. 2) Moreover, the dissimilar partitions are kept distant from each other. 3) Additionally, the proposed framework incorporates an ensemble-based correction strategy for correcting final predictions for the target samples.

III. Proposed Framework

We present a deep UDA framework that learns a deep neural model to effectively label the given unlabeled samples with the help of labeled samples from a related domain. Figure 2 depicts an overview of the architecture of the proposed framework. The source feature extractor is responsible for learning class discriminative features from the labeled source data whereas the target feature extractor (the generator) is expected to generate domain-invariant representations from the target samples. The output image representations from the feature extractors are then labeled by the classifier. The domain classifier (the discriminator) distinguishes between the source and target representations and the centroid alignment component aligns the conditional distributions of the domains. Overall, the proposed framework can be summarized as a training procedure that comprises of three alternating steps. The three key steps are described in detail in the subsequent sections.

A. Step 1: Training the source classifier

The first step of the proposed framework is training the source classifier. With the standard supervised classification loss $\mathcal{L}_{cls}$ (refer Equation 1) on labeled source data (X_s, Y_s), the source feature extractor network learns class-discriminative features and the classifier C to categorize each source sample into one of K given classes.

$$\min_{M_s, C} \mathcal{L}_{cls}(X_s, Y_s) = \\ - E_{(x_s, y_s)\sim(X_S, Y_S)} \sum_{k=1}^{K} 1_{[k=y_s]} \log C(M_s(x_s)) \tag{1}$$

Here, M_s indicates the source feature extractor. The classifier is trained with a randomly generated minibatch with the same number of samples from every class. After training the classifier, the weights of the source feature extractor network are kept fixed for the second step.

B. Step 2: Adversarial Adaptation

In the second step, similar to Adversarial Discriminative Domain Adaptation (ADDA) [22], the target representations are adversarially mapped to the source representations. The target feature generator is trained to mimic source-like representations from the unlabeled target images X_t which is achieved by fooling the domain classifier. Overall, the adversarial adaptation step involves two phases. In the first phase, while keeping the target feature extractor (generator) fixed, we update the domain classifier (discriminator) with a randomly generated mini-batch comprising of samples from the source and target domain. Thereafter, in the second phase, while keeping the domain classifier fixed, we train the target generator with inverted labels. Keeping M_s fixed, Equation 2 depicts the alternate training of the discriminator and the target generator. The discriminator loss $\mathcal{L}_{adv_D}$ is a standard classification loss where the labels indicate the domain of origin whereas for the generator loss $\mathcal{L}_{adv_M}$, the generator features are learned by simply inverting the labels. Here, M_t indicates the target feature extractor.

$$\begin{aligned}&\min_D \mathcal{L}_{adv_D}(X_s, X_t, M_s, M_t) = \\ &- E_{x_s \sim X_S}[\log D(M_s(x_s))] - E_{x_t \sim X_t}[\log(1 - D(M_t(x_t))] \\ \\ &\min_{M_t} \mathcal{L}_{adv_M}(X_s, X_t, D) = -E_{x_t \sim X_t}[\log D(M_t(x_t))]\end{aligned} \tag{2}$$

With adversarial alignment, the sample representations belonging to the same class from the source and the target domain are expected to be mapped in the vicinity of each other. However, while matching the distributions with adversarial adaptation, the semantic relationships within the target domain data and the class-boundaries learned from the source data are ignored. This can lead the target representations to get mapped near the classification boundaries which in turn can lead to incorrect predictions. Hence, there is a need to match the conditional distributions of the domains for improving the alignment. Aligning the conditional distributions requires labeled samples in the target domain. As labeled target samples are not available under the UDA setting, the proposed framework makes use of the pseudo-labels (predicted labels) of the target samples. Aligning the conditional distributions requires bringing the target representations closer to source representations of the same class. However, the predicted pseudo-labels of individual target samples can be noisy. Consequently, aligning individual pseudo-labeled target samples can affect the transfer performance. Similar to MSTN and PFAN, we propose to align the class-wise means of the source and target representations. As explained earlier in Section **??**, aligning the class-wise means limits the impact of noise on the alignment as it reduces the bias from individual pseudo-labeled target samples. However, directly aligning the class-wise means is only adequate when the domain representations (for every label) are densely grouped i.e. the representations form a single cluster for every label. So, the problem arises how to effectively align when the domain representations (in either of the domains) can form multiple compact partitions that can be distant from each other while keeping the noise in check. This issue is addressed in Step 3 of the proposed framework.

C. Step 3: Centroid Alignment

We bring a pseudo-labeled target partition closer to the nearest source partition having the same label. This is achieved by introducing a loss that minimizes the difference between the centroids of a target partition with the nearest source partition having the same label. We will refer to this loss as the "centroid loss" in the remainder of the draft.

Alignment with the centroid loss requires identifying the number of partitions/clusters formed by the similar labeled samples in the source and target domain. We use Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [2] algorithm for identifying the per-label partitions. There are several reasons to why we selected HDBSCAN over other clustering algorithms. First of all, it does not assume anything about the underlying distribution of the clusters and performs better than most algorithms over a wide range of clustering tasks [12]. It can identify partitions with varying density and is relatively faster than most clustering algorithms [11]. Moreover, it lets us define the cluster importance based on size and can effectively identify the outliers that can be ignored while computing the centroids of the partitions.

Let $C_n^{s_k}$ and $C_m^{t_k}$ denote the centroids of the source and target clusters respectively for the k^{th} label where $n \in [1, 2 \cdots n_k]$ and $m \in [1, 2 \cdots m_k]$. Here, n_k and m_k denote the number of partitions obtained for label k in the source and target domain respectively. A centroid of a source partition is computed as the mean vector of the sample representations obtained from the feature extractor M_s. Equation 3 depicts the centroid computation for the i^{th} source partition $P_i^{s_k}$ having the label k.

$$C_i^{s_k} = \frac{1}{N_i^s} \sum_{(x_s^i, y_s^i) \in P_i^{s_k}} M_s(x_s^i) \tag{3}$$

After identifying the per-label partitions in both the domains, we apply the centroid loss that brings a target partition closer to the nearest source partition having the same label. The Centroid Alignment (CA) loss $\mathcal{L}_{CA}$ (refer Equation 4) for aligning the i^{th} target partition having the label j is the euclidean distance to the nearest source partition having the same label. Here, ϕ is the euclidean distance between a pair of the centroids $C \in \mathbb{R}^d$.

$$\mathcal{L}_{CA}(C_i^{t_j}) = arg \min_{\forall n} \ \phi(C_n^{s_j}, C_i^{t_j}), n \in [1, 2 \cdots n_k] \tag{4}$$

The three alternating steps are the focal part of the proposed framework. While minimizing the centroid loss in practice, it is not necessary that samples from all labels are picked in the target minibatch as it is randomly selected. Hence, the categorical information within a minibatch might not be sufficient for computing the per-label partitions. So, after every few iterations, we use all the domain representations (from both source and target) to perform the centroid alignment step. The overall centroid alignment procedure is summarized in Algorithm 1.

Algorithm 1 Centroid Alignment Framework

```
Input: Labeled Source samples: $(X_s, Y_s)$ and unlabeled Target samples: $X_t$. The number of labels is $K$ and $n_k$ denotes the number of partitions for label $k$.
Output: $\mathcal{L}_{CA}$, the Centroid Alignment (CA) loss between the source partitions $C_n^{s_k}$ and the target partitions $C_n^{t_k}$ where $n \in [1, 2 \cdots n_k]$ and $k \in [1, 2 \cdots K]$.
  1. for every label $j \in Y_s$ do
         $n_j^s, C_n^{s_j} \leftarrow$ HDBSCAN($M_s(X_s, Y_s == j)$)
     end for
  2. Randomly sample a batch from $X_t$: $\widehat{X_t}$ and get the pseudo-labels $\widehat{Y_t}$ for this batch.
  3. for every label $j$ do
         $n_j^t, C_n^{t_j} \leftarrow$ HDBSCAN($M_t(\widehat{X_t}, \widehat{Y_t} == j)$)
     end for
  4. for every label $p \in Y_s$ do
         for every cluster $i \in [1, 2 \cdots n_n^{t_p}]$ do
             for every cluster $j \in [1, 2 \cdots n_m^{s_p}]$ do
                 $\mathcal{L}_{CA} \leftarrow \phi(C_i^{s_p}, C_j^{t_p})$
             end for
         end for
     end for
  5. return $\mathcal{L}_{CA}$
```

1) Contrastive Loss Term: The contrastive loss term $\mathcal{L}_{CL}$ (refer Equation 5) enforces the pseudo-labeled target representations to be distant from the source representations with a dissimilar label by a distance of atleast m (the margin). Here, $n_i^{s_p}$ and $n_j^{t_q}$ denote the i^{th} labeled source partition for the p^{th} class-label and j^{th} pseudo-labeled target partition for the q^{th} class-label respectively. Similarly, $C_i^{s_p}$ and $C_j^{t_q}$ denote the centroids of those partitions. The label dissimilarity term is denoted with W_{ij}.

$$\mathcal{L}_{CL} \leftarrow \forall_{p \neq q} \sum_{i=1}^{n_i^{s_p}} \sum_{j=1}^{n_j^{t_q}} \max(0, m- \mid\mid C_i^{s_p} - C_j^{t_q} \mid\mid^2) W_{ij} \quad (5)$$

where $W_{ij} = 1, \text{if} \quad p \neq q \quad \text{else } 0.$

2) Iterative Pruning Strategy: Similar to PFAN [3], we introduced a progressive pruning strategy that mitigates the transfer detriment from incorrect pseudo-labeled examples. The strategy differentiates between "easy" pseudo-labeled target samples from "hard" samples. An "easy" target sample is more likely to get correctly classified than a "hard" sample. After adversarial adaptation, the target sample representations are expected to get mapped near the centroids of the source partitions in the representational space. A target sample x_t^j is considered as a easy sample when its Cosine Similarity (CS) score $\psi(x_t^j)$ (refer Equation 6) with respect to its nearest source partition lies above the threshold τ (refer Equation 7) else it is designated as a hard sample.

$$\psi(x_t^j) = arg \min \; CS(M_t(x_t^j), C_n^{s_j}), n \in [1, 2 \cdots n_k] \quad (6)$$

$$\tau = \frac{1}{1 + e^{-\mu(m+1)}} - 0.01 \quad (7)$$

Here, μ is a constant and m denotes the training steps. With progressive alignment, the similarity of samples increases as target samples get closer to the source samples which in turn leads to hard samples being regarded as easy samples in later iterations. During the centroid alignment procedure, the estimated easy samples for a partition contribute to its centroid whereas the hard samples are ignored.

D. Noisy labels correction

After performing the centroid alignment procedure, the predicted labels of the target samples can still be noisy. So, we employ an iterative label noise correction strategy known as Iterative Cross Learning (ICL) [25]. ICL combines two key ideas, namely, majority voting and co-training to train multiple independent networks over multiple stages where each network is trained from a partition of the given data at every stage. These independent networks work alongside each other to correct each others data for the subsequent stages. For an unseen data point, if the independent networks agree, the prediction is retained else a random label is assigned from the set of given labels. The random flipping of labels on disagreement ensures that the induced noise becomes less structured and uniformly random, that in turn helps to reduce the bias for the networks.

This correction strategy is applied post the centroid alignment step on the pseudo-labeled target samples $(X_t, \widehat{Y_t})$. Here, $\widehat{Y_t}$ indicates the pseudo-labels of the target samples. We train two independent convolutional neural networks with the same network architecture like that of the feature extractors but initialized with different weights. We also use standard data augmentation methods (random cropping and flipping, varying brightness and contrast) while training these networks.

IV. Experiments

We chose three diverse digits datasets: two grey handwritten digits datasets, 1) MNIST[1] [6] and 2) USPS[2], and a real-world digits dataset obtained from natural scene images, 3) SVHN [3] [13]. We followed the same experimental setting given in [22] to evaluate the performance of the proposed algorithm on three transfer scenarios: 1) SVHN → MNIST, 2) MNIST → USPS and 3) USPS → MNIST. We follow the same experimental setting as given in [22] for the three digit-classification transfer tasks. We also evaluated the transfer performance of our algorithm on the Office dataset [14]. It contains images from 3

[1] http://yann.lecun.com/exdb/mnist/

[2] https://www.kaggle.com/bistaumanga/usps-dataset

[3] http://ufldl.stanford.edu/housenumbers/

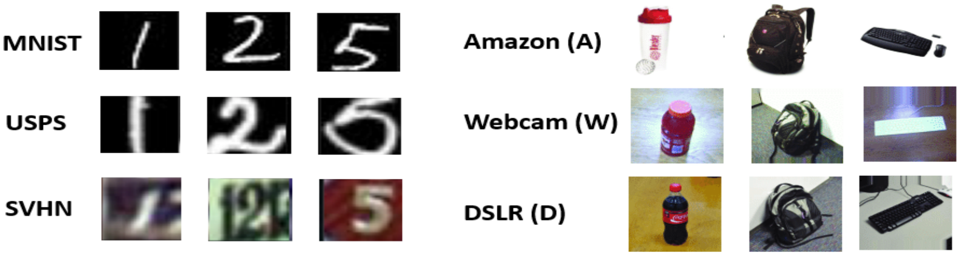


Fig. 3: Depicts a few sample images from the digits datasets and the Office dataset. We compare the transfer performance of the proposed framework against state-of-the-art UDA approaches on cross-dataset digit classification tasks and cross-domain object recognition tasks.

Source→Target	SVHN→MNIST	MNIST→SVHN	MNIST→USPS
Source Only	60.0±1.1	33.0±1.2	75.2±1.6
RevGrad	73.8 ±1.7	35.7 ±2.0	77.1 ±1.8
ADDA	76.0 ±1.8	-	89.4 ±0.2
MSTN	91.7±1.5	did not converge	92.9±1.1
PFAN	93.9±0.8	57.6 ±1.8	95.0±1.3
MPFAN	93.6±1.0 (+0.5)	57.7±1.9 (+1.2)	94.6±1.4 (+0.6)
MPFAN + ICL	**94.1±1.1** (+0.5)	**58.5±1.3** (+1.4)	**95.5±1.0** (+0.4)

TABLE I: Transfer results on digits classification tasks are depicted in terms of mean accuracy and standard deviation. The results of the other transfer approaches are taken from [3]. - indicates that the results are not available. The best results have been highlighted in bold. Here, the value (+x) shows the improvement in the mean accuracy due to the contrastive loss term.

Approach	A →W	D→W	W→D	A→D	D→A	W→A
Source only	61.5±0.5	95.1±0.3	99.0±0.2	64.4±0.5	48.8±0.3	47.0±0.4
RevGrad	73.0±0.5	96.4±0.3	99.2±0.3	72.3±0.3	53.4±0.4	51.2±0.5
ADDA	75.1	97.0	99.6	-	-	-
MSTN	80.5±0.4	96.9±0.1	99.9±0.1	74.5±0.4	62.5±0.4	60.0±0.6
PFAN	83.0±0.3	99.0±0.2	99.9±0.1	76.3±0.3	63.3±0.3	60.8±0.5
MPFAN	82.7±0.5 (+0.9)	98.8±0.2 (+0.1)	99.9±0.1 (+0.0)	76.1±0.4 (+1.2)	63.0±0.3 (+1.7)	60.5±0.5 (+1.2)
MPFAN + ICL	**83.8±0.5** (+1.2)	**99.2±0.3** (+0.1)	**100.0±0.0** (+0.0)	**77.2±0.4** (+0.9)	**63.7±0.3** (+1.5)	**62.4±0.3** (+1.3)

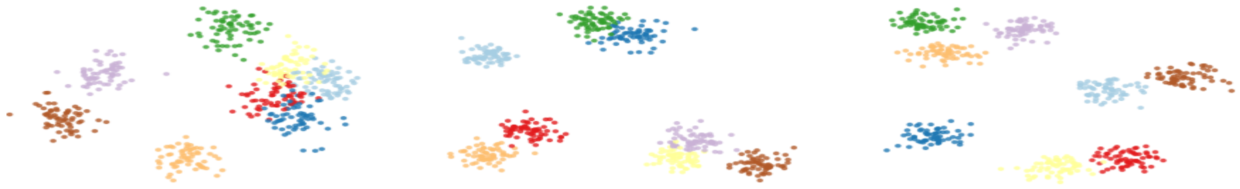
TABLE II: Transfer results on the object recognition tasks (Office dataset) are depicted in terms of mean accuracy and standard deviation. Apart from ADDA [22], the results of the other transfer approaches are taken from [3]. - indicates that the results are not available. The best results have been highlighted in bold. Here, the value (+x) shows the improvement in the mean accuracy due to the contrastive loss term.

different domains, namely, Amazon (A) (from amazon.com), Webcam (W), and DSLR (D) where each domain contains 31 categories. The images for the Webcam and DSLR domain are taken by a webcam and a DSLR camera respectively with varying lighting and pose changes in an office environment. For a fair comparison, we follow the same experimental setting as given in [3] for the six transfer tasks. Figure 3 depicts example images from the chosen datasets.

We utilize all the available labeled samples in the source domain to train the classifier and all the unlabeled samples in the target domain are used for adaptation. Similar to MSTN [24], we repeated each transfer experiment five times and reported the average accuracy and standard deviation.

We compared the performance of the proposed framework against the following baselines and related UDA frameworks.

- **Source only**: Here, we report the accuracy of the trained source classifier on the target dataset.
- **Adversarial Discriminative Domain Adaptation** (ADDA) [22]: ADDA is a generic adversarial adaptation framework that uses GAN loss for aligning the marginal distributions of the source and the target domain.
- **Gradient Reversal** (RevGrad) [4]: In contrast to ADDA, the weights of the source and the target feature generator are tied. Here, the common feature extractor learns domain-invariant features by maximizing the domain classifier loss.
- **Moving Semantic Transfer Network** (MSTN) [24]: MSTN is an iterative adversarial framework that utilizes the pseudo-labels of the target samples to further induce class-wise alignment in the shared feature space. After the adversarial step, the mean of the pseudo-labeled target samples and the mean of the labeled source samples (having the same label) is brought closer in every iteration to optimize class-wise alignment.
- **Progressive Feature Alignment Network** (PFAN) [3]: Similar to MSTN, PFAN also progressively aligns the distributions of the source and the target domain. However, while doing the class-wise mean alignment, PFAN computes the mean from only those target samples that are similar (computed with cosine similarity) to the source-mean having the same label. In every iteration, a gradually increasing similarity threshold is used to identify the relevant subset of target samples that contribute to the class-wise target mean.



(a) MSTN [24]

(b) PFAN [3]

(c) MPFAN

Fig. 4: (Best viewed in color) Depicts the 2D t-SNE visualization of the target representations for the A → W (8 classes were randomly selected) adaptation task.

A. Implementation Details

Network Architecture: For a fair comparison, we use the same architecture for the feature extractors and the discriminator as mentioned in [22] for the three transfer scenarios with the digit classification task. Similarly, for the six transfer scenarios constructed from the Office dataset, we use the same network architecture as followed in [4] for the object recognition tasks.

B. Hyper-parameter Tuning

We follow the same values of the hyper-parameters as mentioned in [22] for learning the source classifier. For a fair comparison of the transfer tasks, the batch size is set to 128 and the same annealing strategy is followed for the learning rate as mentioned in [4]. The moving average coefficient is set as $\theta = 0.7$ and Stochastic Gradient Descent with momentum $= 0.9$ is used as the optimizer for all transfer experiments. In all our experiments, the centroid alignment procedure is applied after every 100 iterations. We observed that in the early iterations of adversarial adaptation, the predictions on the target samples are inconsistent on an average which deters the transfer performance. Consequently, for the first few epochs (in our experiments, it is set to 5), the centroid alignment procedure was not performed. The parameter μ is set to 0.8 [3]. For the final label correction, we used Adam optimizer with a learning rate of 10^{-4} for training the independent networks. The final results are reported with the network training terminated at Stage 2 for the digit classification tasks and Stage 3 for the image classification tasks.

V. Results and Discussion

Digits datasets: The experimental results for the three cross-dataset transfer scenarios on the digits classification task are presented in Table I. It can be observed that the proposed framework (MPFAN + ICL) outperforms all the other methods on all the transfer tasks. It achieves a significant transfer improvement of $+27\%$ on the source-only baseline on an average over the three transfer scenarios. In comparison to standard adversarial adaptation frameworks RevGrad and ADDA, it shows superior performance with a margin of $+19\%$ and $+11\%$ respectively (on average). Both ADDA and RevGrad only match the marginal distributions of the source and target domain but ignore the class-wise alignment in the shared representational space. The significant improvement of MSTN, PFAN, and MPFAN over these two frameworks suggests that matching the class-wise distributions with the help of pseudo-labels enriches the alignment of representations and yields more discriminative features. The proposed framework (MPFAN + ICL) also significantly outperforms MSTN by 2% and marginally outperforms PFAN by 0.5% on an average. The transfer improvement of the proposed framework (MPFAN) over MSTN validates that aligning the per-label partitions is comparatively better than aligning the unpruned class-wise means for matching the class-wise distributions.

Office dataset: The results for the six cross-domain object recognition tasks are presented in Table II. Apart from PFAN, it can be observed that the proposed framework (MPFAN + ICL) still outperforms all the other approaches. However, its performance is slightly inferior to PFAN. This can be attributed to the smaller size of the training datasets for the cross-domain transfer scenarios. The clustering capability of HDBSCAN slightly suffers for the sparser datasets in the high-dimensional representational space. Another key observation is that the ICL strategy for label correction improves the transfer performance of the proposed framework by 0.7% on an average over the three cross-dataset digits classification tasks and by 1.2% on the cross-domain object recognition tasks. This suggests that there is scope for noise correction in the final predictions for the target samples.

Moreover, from Table I and II, it can be observed that there is a significant improvement in the transfer performance with the contrastive loss term. This validates the hypothesis that keeping the dissimilar domain partitions at bay helps to learn better feature representations for the target task. For all the experiments, the value of the margin (m) was set to 0.5. To sum up, the overall framework (MPFAN + ICL + contrastive loss) significantly outperforms all the other approaches over all the transfer tasks.

Figure 4 shows the 2D t-SNE [10] visualization of the target representations for the A → W adaptation task from the Office-31 dataset. It can be observed that the mean-alignment approaches, MSTN and PFAN, including the proposed approach, MPFAN, learn good class-discriminative representations. However, the inter-class spread is much more for MPFAN in comparison to the other approaches. Secondly, for some categories, the representations learned with MSTN and PFAN appear to sparsely coalesce with each other, which is not the case for MPFAN. These observations suggest that MPFAN learns better target representations than the mean-alignment approaches.

VI. Conclusion

In this paper, we present a simple and effective UDA framework that carefully aligns the target representations to the source representations by matching the feature distributions between the domains, even at the class-level. The transfer improvement over the extensive experiments on cross-dataset digit classification tasks and cross-domain object recognition tasks suggests that the multi-partition centroid alignment approach is significantly better than aligning the class-wise means of the domain representations.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[2] R. J. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[3] C. Chen, W. Xie, T. Xu, W. Huang, Y. Rong, X. Ding, Y. Huang, and J. Huang. Progressive feature alignment for unsupervised domain adaptation. *CoRR*, abs/1811.08585, 2018.

[4] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

[5] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 1735–1742. IEEE, 2006.

[6] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[7] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[8] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105, 2015.

[9] Z. Luo, Y. Zou, J. Hoffman, and L. F. Fei-Fei. Label efficient learning of transferable representations acrosss domains and tasks. In *Advances in Neural Information Processing Systems*, pages 165–177, 2017.

[10] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[11] L. McInnes and J. Healy. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 33–42. IEEE, 2017.

[12] L. McInnes, J. Healy, and S. Astels. Comparing python clustering algorithms. *Hdbscan Docs*, 2016.

[13] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[14] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[15] K. Saito, Y. Ushiku, and T. Harada. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2988–2997. JMLR. org, 2017.

[16] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018.

[17] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018.

[18] O. Sener, H. O. Song, A. Saxena, and S. Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 2110–2118, 2016.

[19] X. Shu, G.-J. Qi, J. Tang, and J. Wang. Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 35–44. ACM, 2015.

[20] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pages 443–450. Springer, 2016.

[21] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015.

[22] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[23] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[24] S. Xie, Z. Zheng, L. Chen, and C. Chen. Learning semantic representations for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 5419–5428, 2018.

[25] B. Yuan, J. Chen, W. Zhang, H.-S. Tai, and S. McMains. Iterative cross learning on noisy labels. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 757–765. IEEE, 2018.

[26] W. Zhang, W. Ouyang, W. Li, and D. Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3801–3809, 2018.