Unsupervised Domain Adaptation using Feature-Whitening and Consensus Loss

Subhankar Roy[1,2], Aliaksandr Siarohin[1], Enver Sangineto[1], Samuel Rota Bulò[3],
Nicu Sebe[1] and Elisa Ricci[1,2]

[1]DISI, University of Trento, Italy, [2]Fondazione Bruno Kessler, Trento, Italy, [3]Mapillary Research

{subhankar.roy, aliaksandr.siarohin, enver.sangineto, niculae.sebe, e.ricci}@unitn.it
samuel@mapillary.com

Abstract

A classifier trained on a dataset seldom works on other datasets obtained under different conditions due to domain shift. This problem is commonly addressed by domain adaptation methods. In this work we introduce a novel deep learning framework which unifies different paradigms in unsupervised domain adaptation. Specifically, we propose domain alignment layers which implement feature whitening for the purpose of matching source and target feature distributions. Additionally, we leverage the unlabeled target data by proposing the Min-Entropy Consensus loss, which regularizes training while avoiding the adoption of many user-defined hyper-parameters. We report results on publicly available datasets, considering both digit classification and object recognition tasks. We show that, in most of our experiments, our approach improves upon previous methods, setting new state-of-the-art performances.

1. Introduction

Deep learning methods have been successfully applied to different visual recognition tasks, demonstrating an excellent generalization ability. However, analogously to other statistical machine learning techniques, deep neural networks also suffer from the problem of *domain shift* [47], which is observed when predictors trained on a dataset do not perform well when applied to novel domains.

Since collecting annotated training data from every possible domain is expensive and sometimes even impossible, over the years several Domain Adaptation (DA) methods [34, 5] have been proposed. DA approaches leverage labeled data in a source domain in order to learn an accurate prediction model for a target domain. Specifically, in the special case of Unsupervised Domain Adaptation (UDA), no annotated target data are available at training time. Note that, even if target-sample labels are not available, unlabeled data can and usually are exploited at training time.

Most UDA methods attempt to reduce the domain shift

by directly aligning the source and target marginal distributions. Notably, approaches based on the *Correlation Alignment* paradigm model domain data distributions in terms of their second-order statistics. Specifically, they match distributions by minimizing a loss function which corresponds to the difference between the source and the target covariance matrices obtained using the network's last-layer activations [43, 44, 32]. Another recent and successful UDA paradigm exploits *domain-specific alignment layers*, derived from Batch Normalization (BN) [18], which are directly embedded within the deep network [3, 24, 31]. Other prominent research directions in UDA correspond to those methods which also exploit the target data posterior distribution. For instance, the *entropy minimization* paradigm adopted in [3, 37, 13], enforces the network's prediction probability distribution on each target sample to be peaked with respect to some (unknown) class, thus penalizing high-entropy target predictions. On the other hand, the *consistency-enforcing* paradigm [38, 7, 46] is based on specific loss functions which penalize inconsistent predictions over perturbed copies of the same target samples.

In this paper we propose to unify the above paradigms by introducing two main novelties. First, we align the source and the target data distributions using covariance matrices similarly to [43, 44, 32]. However, instead of using a loss function computed on the last-layer activations, we use domain-specific alignment layers which compute domain-specific covariance matrices of intermediate features. These layers "whiten" the source and the target features and project them into a common spherical distribution (see Fig. 1 (a), blue box). We call this alignment strategy *Domain-specific Whitening Transform* (DWT). Notably, our approach generalizes previous BN-based DA methods [3, 24, 30] which do not consider inter-feature correlations and rely only on feature standardization.

The second novelty we introduce is a novel loss function, the Min-Entropy Consensus (MEC) loss, which merges both the entropy [3, 37, 13] and the consistency [7] loss function. The motivation behind our proposal is to avoid the
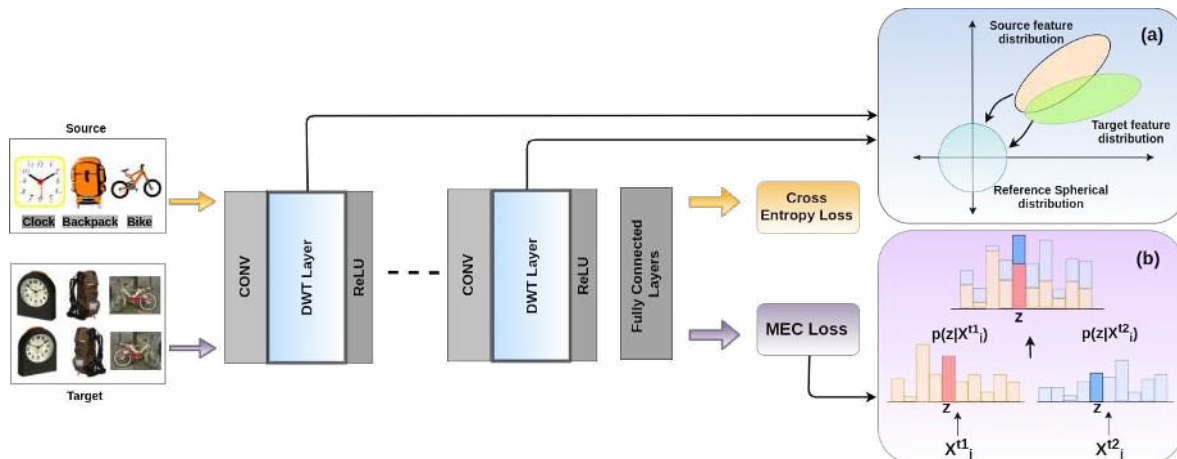
Figure 1. Overview of the proposed deep architecture embedding our DWT layers and trained with the proposed MEC loss. (a) Due to domain shift the source and the target data have different marginal feature distributions. Our DWT estimates these distributions using dedicated sample batches and then "whitens" them projecting them into a common, spherical distribution. (b) The proposed MEC loss univocally selects a pseudo-label z that maximizes the agreement between two perturbed versions $\mathbf{x}_i^{t_1}$ and $\mathbf{x}_i^{t_2}$ of the same target sample.

tuning of the many hyper-parameters which are typically required when considering several loss terms and, specifically, the confidence-threshold hyper-parameters [7]. Indeed, due to the mismatch between the source and the target domain, and because of the unlabeled target-data assumption, hyper-parameters are hard to be tuned in UDA [32]. The proposed MEC loss simultaneously encourages coherent predictions between two perturbed versions of the same target sample and exploits these predictions as pseudo-labels for training. (Fig. 1 (b), purple box).

We plug our proposed DWT and the MEC loss into different network architectures and we empirically show a significant boost in performance. In particular, we achieve state-of-the-art results in different UDA benchmarks: MNIST [22], USPS [8], SVHN [33], CIFAR-10, STL10 [4] and Office-Home [50]. Our code[1] is publicly available.

2. Related Work

Unsupervised Domain Adaptation. Several previous works have addressed the problem of DA, considering both shallow models and deep architectures. In this section we focus on only deep learning methods for UDA, as these are the closest to our proposal.

UDA methods mostly differ in the strategy used to reduce the discrepancy between the source and the target feature distributions and can be grouped in different categories. The first category includes methods modeling the domain distributions in terms of their first and second order statistics. For instance, some works aim at reducing the domain shift by minimizing the Maximum Mean Discrepancy

[27, 28, 50] and describe distributions in terms of their first order statistics. Other works consider also second-order statistics using the *correlation alignment* paradigm (Sec. 1) [44, 32]. Instead of introducing additional loss functions, more recent works deal with the domain-shift problem by directly embedding into a deep network *domain alignment layers* which exploit BN [24, 3, 31, 29].

A second category of methods include approaches which learn domain-invariant deep representations. For instance, in [9] a gradient reversal layer learns discriminative domain-agnostic representations. Similarly, in [48] a domain-confusion loss is introduced, encouraging the network to learn features robust to the domain shift. Haeusser *et al.* [14] present Associative Domain Adaptation, an approach which also learns domain-invariant embeddings.

A third category includes methods based on Generative Adversarial Networks (GANs) [35, 1, 45, 40, 39]. The main idea behind these approaches is to directly transform images from the target domain to the source domain. While GAN-based methods are especially successful in adaptation from synthetic to real images and in case of non-complex datasets, they have limited capabilities for complex images.

Entropy minimization, first introduced in [12], is a common strategy in semi-supervised learning [51]. In a nutshell, it consists in exploiting the high-confidence predictions of unlabeled samples as pseudo-labels. Due to its effectiveness, several popular UDA methods [35, 3, 37, 28] have adopted the entropy-loss for training deep networks.

Another popular paradigm in UDA, which we refer to as the *consistency-enforcing* paradigm, is realized by perturbing the target samples and then imposing some consistency between the predictions of two perturbed versions of the same target input. Consistency is imposed by defining

---

[1]Code available at https://github.com/roysubhankar/dwt-domain-adaptation

appropriate loss functions, as shown in [37, 7, 38]. The consistency loss paradigm is effective but it becomes uninformative if the network produces uniform probability distributions for corresponding target samples. Thus, previous methods also integrate a Confidence Thresholding (CT) technique [7], in order to discard unreliable predictions. Unfortunately, CT introduces additional user-defined and dataset-specific hyper-parameters which are difficult to tune in an UDA scenario [32]. Differently, as demonstrated in our experiments, our MEC loss eliminates the need of CT and the corresponding hyper-parameters.

Feature Decorrelation. Recently, Huang *et al.* [17] and Siarohin *et al.* [42] proposed to replace BN with feature whitening in a discriminative and generative setting, respectively. However, none of these works consider a DA problem. We show in this paper that feature whitening can be used to align the source and the target marginal distributions using layer-specific covariance matrices without the need of a dedicated loss function as in previous correlation alignment methods.

3. Method

In this section we present the proposed UDA approach. Specifically, after introducing some preliminaries, we describe our Domain-Specific Whitening Transform and, finally, the proposed Min-Entropy Consensus loss.

3.1. Preliminaries

Let $\mathcal{S} = \{(I_j^s, y_j^s)\}_{j=1}^{n_s}$ be the labeled source dataset, where I_j^s is an image and $y_j^s \in \mathcal{Y} = \{1, 2 \ldots, C\}$ its associated label, and $\mathcal{T} = \{I_i^t\}_{i=1}^{n_t}$ be the unlabeled target dataset. The goal of UDA is to learn a predictor for the target domain by using samples from both $\mathcal{S}$ and $\mathcal{T}$. Learning a predictor for the target domain is not trivial because of the issues discussed in Sec. 1.

A common technique to reduce domain shift is to use BN-based layers inside a network, such as to project the source and target feature distributions to a reference distribution through feature standarization. As mentioned in Sec. 1, in this work we propose to replace feature standarization with whitening, where the whitening operation is domain-specific. Before introducing the proposed whitening-based distribution alignment, we recap below BN. Let $B = \{\mathbf{x}_1, ..., \mathbf{x}_m\}$ be a mini-batch of m input samples to a given network layer, where each element $\mathbf{x}_i \in B$ is a d-dimensional feature vector, *i.e.* $\mathbf{x}_i \in \mathbb{R}^d$. Given B, in BN each $\mathbf{x}_i \in B$ is transformed as follows:

$$BN(x_{i,k}) = \gamma_k \frac{x_{i,k} - \mu_{B,k}}{\sqrt{\sigma_{B,k}^2 + \epsilon}} + \beta_k, \qquad (1)$$

where k ($1 \leq k \leq d$) indicates the k-th dimension of the data, $\mu_{B,k}$ and $\sigma_{B,k}$ are, respectively, the mean and the stan-

dard deviation computed with respect to the k-th dimension of the samples in B and ϵ is a constant used to prevent numerical instability. Finally, γ_k and β_k are scaling and shifting learnable parameters.

In the next section we present our DWT, while in Sec. 3.3 we present the proposed MEC loss. It is worth noting that each proposed component can be plugged independently in a network without having to rely on each other.

3.2. Domain-specific Whitening Transform

As stated above, BN is based on a per-dimension *standardization* of each sample $\mathbf{x}_i \in B$. Hence, once normalized, the batch samples may still have correlated feature values. Since our goal is to use feature normalization in order to alleviate the domain-shift problem (see below), we argue that plain standardization is not enough to align the source and the target marginal distributions. For this reason we propose to use Batch Whitening (BW) instead of BN, which is defined as:

$$BW(x_{i,k}; \Omega) = \gamma_k \hat{x}_{i,k} + \beta_k, \qquad (2)$$

$$\hat{\mathbf{x}}_i = W_B(\mathbf{x}_i - \boldsymbol{\mu}_B). \qquad (3)$$

In Eq. (3), the vector $\boldsymbol{\mu}_B$ is the mean of the elements in B (being $\mu_{B,k}$ its k-th component) while the matrix W_B is such that: $W_B^\top W_B = \Sigma_B^{-1}$, where Σ_B is the covariance matrix computed using B. $\Omega = (\boldsymbol{\mu}_B, \Sigma_B)$ are the batch-dependent first and second-order statistics. Eq. (3) performs the *whitening* of $\mathbf{x}_i$ and the resulting set of vectors $\hat{B} = \{\hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_m\}$ lie in a spherical distribution (i.e., with a covariance matrix equal to the identity matrix).

Our network takes as input two different batches of data, randomly extracted from $\mathcal{S}$ and $\mathcal{T}$, respectively. Specifically, given any arbitrary layer l in the network, let $B^s = \{\mathbf{x}_1^s, ..., \mathbf{x}_m^s\}$ and $B^t = \{\mathbf{x}_1^t, ..., \mathbf{x}_m^t\}$ denote the batch of intermediate output activations, from layer l, for the source and target domain, respectively. Using Eq. (2)-(3) we can now define our Domain-specific Whitening Transform (DWT). Let x^s and x^t denote the inputs to the DWT layer from the source and the target domain, respectively. Our DWT is defined as follows (we drop the sample index i and dimension index k for the sake of clarity):

$$DWT(x^s; \Omega^s) = BW(x^s, \Omega^s), \qquad (4)$$

$$DWT(x^t; \Omega^t) = BW(x^t, \Omega^t). \qquad (5)$$

We estimate separate statistics ($\Omega^s = (\boldsymbol{\mu}_B^s, \Sigma_B^s)$ and $\Omega^t = (\boldsymbol{\mu}_B^t, \Sigma_B^t)$) for B^s and B^t and use them for whitening the corresponding activations, projecting the two batches into a common spherical distribution (Fig. 1 (a)).

W_B^s and W_B^t are computed following the approach described in [42], which is based on the Cholesky decomposition [6]. The latter is faster [42] than the ZCA-based whitening [19] adopted in [17]. In the Supplementary Material we provide more details on how W_B^s and W_B^t are

computed. Differently from [42] we replace the "coloring" step after whitening with simple scale and shift operations, thereby preventing the introduction of extra parameters in the network. Also, differently from [42] we use *feature grouping* [17] (Sec. 3.2.1) in order to make the batch-statistics estimate more robust when m is small and d is large. During training, the DWT layers accumulate the statistics for the target domain using a moving average of the batch statistics (Ω_{avg}^t).

In summary, the proposed DWT layers replace the correlation alignment of the last-layer feature activations with the intermediate-layer feature whitening, performed at different levels of abstraction. In Sec. 3.2.1 we show that BN-based domain alignment layers [24, 3] can be seen as a special case of DWT layers.

3.2.1 Implementation Details

Given a typical block (Conv layer → BN → ReLU) of a CNN, we replace the BN layer with our proposed DWT layer (see in Fig. 1), obtaining: (Conv layer → DWT → ReLU). Ideally, in order to project the source and target feature distributions to a reference one, the DWT layers should perform full-feature whitening using a $d \times d$ whitening matrix, where d is the number of features. However, the computed covariance matrix Σ_B can be ill-conditioned if d is large and m is small. For this reason, unlike [42] and similar to [17] we use *feature grouping*, where the features are grouped into subsets of size g. This results in better-conditioned covariance matrices but into partially whitened features. In this way we reach a compromise between full-feature whitening and numerical stability. Interestingly, when $g = 1$, the whitening matrices reduce to diagonal matrices, thus realizing feature standardization as in [3, 24].

3.3. Min-Entropy Consensus Loss

The impossibility of using the cross-entropy loss on the unlabeled target samples is commonly circumvented using some common unsupervised loss, such as the entropy [3, 37] or the consistency loss [7, 38]. While minimizing the entropy loss ensures that the predictor maximally separates the target data, minimization of the consistency loss forces the predictor to deliver consistent predictions for target samples coming from identical (yet unknown) category. Therefore, given the importance of exploiting better the unlabeled target data and the limitations of the above two losses (see Sec. 1), we propose a novel Min-Entropy Consensus (MEC) loss within the framework of UDA. We explain below how MEC loss merges both the entropy and the consistency loss into a single unified function.

Similar to the consistency loss, the proposed MEC loss requires input data perturbations. Unless otherwise explicitly specified, we apply common data-perturbation techniques on both $\mathcal{S}$ and $\mathcal{T}$ using affine transformations and Gaussian blurring operations. When we use the MEC loss, the network is fed with three batches instead of two. Specifically, apart from B^s, we use two different target batches (B_1^t and B_2^t), which contain duplicate pairs of images differing only with respect to the adopted image perturbation.

Conceptually, we can think of this pipeline as three different networks with three separate domain-specific statistics Ω^s, Ω_1^t and Ω_2^t but with shared network weights. However, since both B_1^t and B_2^t are drawn from the same distribution, we estimate a single Ω^t using both the target batches ($B_1^t \bigcup B_2^t$). As an additional advantage, this makes it possible to use $2m$ samples for computing Σ_B^t.

Let $B^s = \{\mathbf{x}_1^s, ..., \mathbf{x}_m^s\}$, $B_1^t = \{\mathbf{x}_1^{t_1}, ..., \mathbf{x}_m^{t_1}\}$ and $B_2^t = \{\mathbf{x}_1^{t_2}, ..., \mathbf{x}_m^{t_2}\}$ be three batches of the last-layer activations. Since the source samples are labeled, the cross-entropy loss (L^s) can be used in case of B^s:

$$L^s(B^s) = -\frac{1}{m} \sum_{i=1}^{m} \log p(y_i^s | \mathbf{x}_i^s), \qquad (6)$$

where $p(y_i^s | \mathbf{x}_i^s)$ is the (soft-max-based) probability prediction assigned by the network to a sample $\mathbf{x}_i^s \in B^s$ with respect to its ground-truth label y_i^s. However, ground-truth labels are not available for target samples. For this reason, we propose the following MEC loss (L^t):

$$L^t(B_1^t, B_2^t) = \frac{1}{m} \sum_{i=1}^{m} \ell^t(\mathbf{x}_i^{t_1}, \mathbf{x}_i^{t_2}), \qquad (7)$$

$$\ell^t(\mathbf{x}_i^{t_1}, \mathbf{x}_i^{t_2}) = -\frac{1}{2} \max_{y \in \mathcal{Y}} \Big(\log p(y | \mathbf{x}_i^{t_1}) + \log p(y | \mathbf{x}_i^{t_2}) \Big). \qquad (8)$$

In Eq. (8), $\mathbf{x}_i^{t_1} \in B_1^t$ and $\mathbf{x}_i^{t_2} \in B_2^t$ are activations of two corresponding perturbed target samples.

The intuitive idea behind our proposal is that, similarly to consistency-based losses [7, 38], since $\mathbf{x}_i^{t_1}$ and $\mathbf{x}_i^{t_2}$ correspond to the same image, the network should provide similar predictions. However, unlike the aforementioned methods which compute the L2-norm or the binary cross-entropy between these predictions, the proposed MEC loss finds the class z such that $z = \operatorname{argmin}_{y \in \mathcal{Y}} \Big(\log p(y | \mathbf{x}_i^{t_1}) + \log p(y | \mathbf{x}_i^{t_2}) \Big)$. z is the class in which the posteriors corresponding to $\mathbf{x}_i^{t_1}$ and $\mathbf{x}_i^{t_2}$ maximally agree. We then use z as the pseudo-label, which can be selected without ad-hoc confidence thresholds. In other words, instead of using high-confidence thresholds to discard unreliable target samples [7], we use all the samples but we backpropagate the error with respect to only z.

The dynamics of MEC loss is the following. First, similarly to the consistency losses, it forces the network to provide coherent predictions. Second, differently from consistency losses, which are prone to attain a near zero value with uniform posterior distributions, it enforces peaked predictions. See the Supplementary Material for a more formal

relation between the MEC loss and both entropy and consistency loss.

The final loss L is a weighted sum of L^s and L^t:
$$L(B^s, B_1^t, B_2^t) = L^s(B^s) + \lambda L^t(B_1^t, B_2^t).$$

3.4. Discussion

The proposed DWT generalizes the BN-based DA approaches by decorrelating the batch features. Besides the analogy with the correlation-alignment methods mentioned in Sec. 1, in which covariance matrices are used to estimate and align the source and the target distributions, a second reason for which we believe that full-whitening is important is due to the relation between feature normalization and the smoothness of the loss [41, 21, 17, 23, 36]. For instance, previous works [23, 36] showed that better conditioning of the input-feature covariance matrix leads to better conditioning of the Hessian of the loss function, making the gradient descent weight updates closer to Newton updates. However, BN only performs standardization, which barely improves the conditioning of the covariance matrix when the features are correlated [17]. Conversely, feature whitening completely decorrelates the batch samples, thus potentially improving the smoothness of the landscape of the loss function.

The importance of a smoothed loss function is even higher when entropy-like losses on unlabeled data are used. For instance, Shu *et al.* [41] showed that minimizing the entropy forces the classifier to be confident on the unlabeled target data, thus potentially driving the classifiers decision boundaries away from the target data. However, without a locally-Lipschitz constraint on the loss function (*i.e.* with a non smoothed loss landscape), the decision boundaries can be placed close to the training samples even when the entropy is minimized [41]. Since our MEC loss is related with both the entropy and the consistency loss, we employ DWT also to improve the smoothness of our loss function in order to alleviate overfitting phenomena related to the use of unlabeled data.

4. Experiments

In this section we provide details about our implementation and training protocols and we report our experimental evaluation. We conduct experiments on both small and large-scale datasets and we compare our method with state-of-the-art approaches. We also present an ablation study to analyze the impact of each of our contributions on the classification accuracy.

4.1. Datasets

We conduct experiments on the following datasets:

MNIST $\leftrightarrow$ USPS. The **MNIST** dataset [22] contains grayscale images (28×28 pixels) depicting handwritten

(a) MNIST $\leftrightarrow$ USPS

(b) SVHN $\leftrightarrow$ MNIST

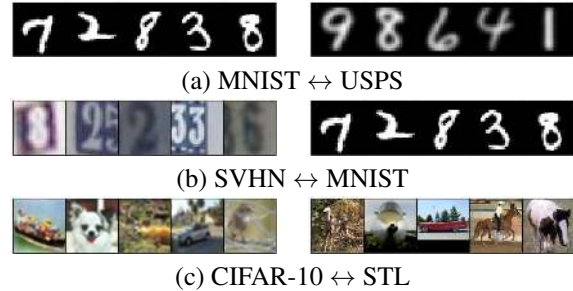(c) CIFAR-10 $\leftrightarrow$ STL

Figure 2. Small image datasets used in our experiments.

Figure 3. Sample images from the Office-Home dataset.

digits ranging from 0 to 9. The **USPS** [8] dataset is similar to MNIST, but images have smaller resolution (16×16 pixels). See Fig. 2(a) for sample images.

MNIST $\leftrightarrow$ SVHN. Street View House Number (SVHN) [33] images are 32×32 pixels RGB images. Similarly to the MNIST dataset digits range from 0 to 9. However, in SVHN images have variable colour intensities and depict non-centered digits. Thus, there is a significant domain shift with respect to MNIST (Fig. 2(b))

CIFAR-10 $\leftrightarrow$ STL: CIFAR-10 is a 10 class dataset of RGB images depicting generic objects and with resolution 32×32 pixels. STL [4] is similar to the CIFAR-10, except it has fewer labelled training images per class and has images of resolution 96×96 pixels. The non-overlapping classes - "frog" and "monkey" are removed from CIFAR-10 and STL, respectively. Samples are shown in Fig. 2.(c).

Office-Home: The Office-Home [50] dataset comprises 4 distinct domains, each corresponding to 65 different categories (Fig. 3). There are 15,500 images in the dataset, thus this represents large-scale benchmark for testing domain adaptation methods. The domains are: Art(**Ar**), Clipart (**Cl**), Product (**Pr**) and Real World (**Rw**).

4.2. Experimental Setup

To fairly compare our method with other UDA approaches, in the digits experiments we adopt the same base networks proposed in [10]. For the CIFAR-10$\leftrightarrow$STL experiments we use the network described in [7]. We train the networks using the Adam optimizer [20] with a mini-batch of cardinality $m = 64$ samples, an initial learning rate of 0.001 and weight decay of 5×10^{-4}. The networks are trained for a total of 120 epochs with learning rate being decreased by a factor of 10 after 50 and 90 epochs. We use the SVHN $\rightarrow$ MNIST setting to fix the value of the hyperparameter λ to 0.1 and to set group size (g) equal to 4. These hyperparameters values are used for all the datasets.

In the Office-Home dataset experiments we use a ResNet-50 [15] following [26]. In our experiments we modify ResNet-50 by replacing the first BN layer and the BN layers in the first residual block (with 64 features) with DWT layers. The network is initialized with weights taken from a pre-trained model trained on the ILSVRC-2012 dataset. We discard the final fully-connected layer and we replace it with a randomly initialized fully-connected layer with 65 output logits. During training, each domain-specific batch is limited to $m = 20$ samples (due to GPU memory constraints). The SGD optimizer is used with an initial learning rate of 10^{-2} for the randomly initialized final layer and 10^{-3} for the rest of the trainable parameters of the network. The network is trained for a total of 60 epochs where one "epoch" is the pass through the entire data set having the lower number of training samples. The learning rates are then decayed by a factor of 10 after 54 epochs. Differently from the small-scale datasets experiments, where target samples have predefined train and test splits, in the Office-Home experiments, all the target samples (without labels) are used during training and evaluation.

To demonstrate the effect our contributions, we consider three different variants for the proposed method. In the first variant (denoted as **DWT** in Sec. 3.2), we only consider DWT layers *without* the proposed MEC loss. In practice, in the considered network architectures we replace the BN layers which follows the convolutional layers with DWT layers. Supervised cross-entropy loss is used for the labeled source samples and the entropy-loss as in [3] is used for the unlabeled target samples. No data-augmentation is used here. In the second variant, denoted as **DWT-MEC**, we also exploit the proposed MEC loss (this corresponds to our full method). In this case we need perturbations of the input data, which are obtained by following some basic data-perturbation schemes like image translation by a factor of [0.05, 0.05], Gaussian blur ($\sigma = 0.1$) and random affine transformation as proposed in [7]. In the third variant (**DWT-MEC (MT)**) we plug our proposed DWT layers and the MEC loss in the Mean-Teacher (MT) training paradigm [46].

4.3. Results

In this section we present an extensive experimental analysis of our approach, showing both the results of an ablation study and a comparison with state-of-the-art methods.

4.3.1 Ablation Study

We first conduct a thorough analysis of our method assessing, in isolation, the impact of our two main contributions: (i) aligning source and target distributions by embedded DWT layers; and (ii) leveraging target data through our threshold-free MEC loss.

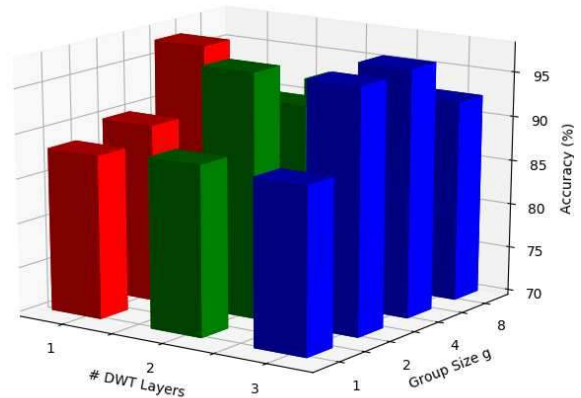First, we consider the SVHN→MNIST setting and we

Figure 4. SVHN → MNIST experiment: accuracy at varying number of DWT layers and group size. Different colors are used to improve readability.

show the benefit of feature whitening over BN. We vary the number of whitening layers from 1 to 3 and simultaneously change the group size (g) from 1 to 8 (see Sec. 3.2.1). With group size equal to 1, DWT layers reduces to DA layers as proposed in [3, 24]. Our results are shown in Fig. 4 and from the figure it is clear that when $g = 1$ the accuracy stays consistently below 90%. This behaviour can be ascribed to the sub-optimal alignment of source and target data distributions achieved with previous BN-based DA layers. When the group size increases, the feature decorrelation performed by the DWT layers comes into play and results into a significant improvement in terms of performance. The accuracy increases monotonically as the group size grows until the value of $g = 4$, then it start to decrease. This final drop is probably due to ill-conditioned covariance matrices. Indeed, a covariance matrix with size 8×8 is perhaps poorly estimated due to the lack of samples in a batch (Sec. 3.2.1). Importantly, Fig. 4 also shows that increasing the number of DWT layers has a positive impact on the accuracy. This is in contrast with [17], where feature decorrelation is used only in the first layer of the network.

In Tab. 2 we evaluate the effectiveness of the proposed MEC loss and we compare our approach with the consistency based loss adopted by French *et al.* [7]. We use Self-Ensembling (SE) [7] with and without confidence thresholding (CT) on the network predictions of the teacher network. To fairly compare our approach with SE we also consider a mean-teacher scheme in our framework. We observe that SE have excellent performance when the CT is set to a very high value (0.936 as in [7]) but it performance drops when CT is set equal to 0, especially in the SVHN→MNIST setting. This shows that the consistency loss in [7] may be harmful when the network is not confident on the target samples. Conversey, the proposed MEC loss leads to re-

Methods	Source Target	MNIST USPS	USPS MNIST	SVHN MNIST	MNIST SVHN
Source Only		78.9	57.1±1.7	60.1±1.1	20.23±1.8
w/o augmentation					
CORAL [43]		81.7	-	63.1	-
MMD [48]		81.1	-	71.1	-
DANN [10]		85.1	73.0±2.0	73.9	35.7
DSN [2]		91.3	-	82.7	-
CoGAN [25]		91.2	89.1±0.8	-	-
ADDA [49]		89.4±0.2	90.1±0.8	76.0±1.8	-
DRCN [11]		91.8±0.1	73.7±0.1	82.0±0.2	40.1±0.1
ATT [37]		-	-	86.20	52.8
ADA [13]		-	-	97.6	-
AutoDIAL [3]		97.96	97.51	89.12	10.78
SBADA-GAN [35]		97.6	95.0	76.1	**61.1**
GAM [16]		95.7±0.5	98.0±0.5	74.6±1.1	-
MECA [32]		-	-	95.2	-
DWT		**99.09**±0.09	**98.79**±0.05	**97.75**±0.10	28.92 ±1.9
Target Only		96.5	99.2	99.5	96.7
w/ augmentation					
SE [a] [7]		88.14±0.34	92.35±8.61	93.33±5.88	33.87±4.02
SE [b] [7]		98.23±0.13	**99.54**±0.04	**99.26**±0.05	**37.49**±2.44
SE [† b] [7]		99.29±0.16	99.26±0.04	97.88±0.03	24.09±0.33
DWT-MEC[b]		99.01±0.06	99.02±0.05	97.80±0.07	30.20±0.92
DWT-MEC (MT)[b]		**99.30**±0.19	99.15±0.05	99.14±0.02	31.58±2.34

Table 1. Accuracy (%) on the digits datasets: comparison with state of the art. [a] indicates minimal usage of data augmentation and [b] considers augmented source and target data. [†] indicates our implementation of SE [7].

Method	Source Target	MNIST USPS	USPS MNIST	SVHN MNIST
SE (w/ CT) [7]		99.29	99.26	97.88
SE (w/o CT) [7]		98.71	97.63	26.80
DWT-MEC (MT)		**99.30**	**99.15**	**99.14**

Table 2. Accuracy (%) on the digits datasets. Comparison between the consistency loss in SE method [7] (with and without CT) and our threshold-free MEC loss.

sults which are on par to SE in the MNIST↔USPS settings and to higher accuracy in the SVHN→MNIST setting. This clearly demonstrates that our proposed loss avoids the need of introducing the CT hyper-parameter and, at the same time, yields to better performance. It is important to remark that, in the case of UDA, tuning hyper-parameters is hard as target samples are unlabeled and cross-validation on source data is unreliable because of the domain shift problem [32].

4.3.2 Comparison with State-of-the-Art Methods

In this section we present our results and compare with previous UDA methods. Tab. 1 reports the results obtained on the digits datasets. We compare with several baselines: Correlation Alignment (CORAL) [43], Simultaneous Deep Transfer (MMD) [48], Domain-Adversarial Training of Neural Networks (DANN) [10], Domain separation networks [2], Coupled generative adversarial net-works (Co-GAN) [25], Adversarial discriminative domain adaptation (ADDA) [49], Deep reconstruction-classification networks (DRCN), [11], Asymmetric tri-training [37], Associative domain adaptation (ADA) [13], AutoDIAL [3], SBADA-GAN [35], Domain transferthrough deep activation matching (GAM) [16], Minimal-entropy correlation alignment (MECA) [32] and SE [7]. Note that the Virtual Adversarial Domain Adaptation (VADA) [41] use a different network, thus cannot be compared with the other methods (including ours) which are based on a different capacity network. For this reason, [41] is not reported in Tab. 1. Results associated with each method are taken from the corresponding papers. We re-implemented SE as the numbers reported in the original paper [7] refer to different network architectures.

Tab. 1 is split in two sections, separating those methods that exploit data augmentation from those which use only the original training data. Compared with no-data augmentation methods, our DWT performs better than previous UDA methods in the three settings. Our method is less effective in the MNIST→SVHN due to the strong domain shift between the two domains. In this setting, GAN-based

Method	Source Target	Ar Cl	Ar Pr	Ar Rw	Cl Ar	Cl Pr	Cl Rw	Pr Ar	Pr Cl	Pr Rw	Rw Ar	Rw Cl	Rw Pr	Avg
ResNet-50 [15]		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 [27]		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 [10]		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 [28]		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
SE [7]		48.8	61.8	72.8	54.1	63.2	65.1	50.6	49.2	72.3	66.1	55.9	78.7	61.5
CDAN-RM [26]		49.2	64.8	72.9	53.8	63.9	62.9	49.8	48.8	71.5	65.8	56.4	79.2	61.6
CDAN-M [26]		**50.6**	65.9	73.4	55.7	62.7	64.2	51.8	**49.1**	74.5	68.2	**56.9**	80.7	62.8
DWT-MEC		50.3	**72.1**	**77.0**	**59.6**	**69.3**	**70.2**	**58.3**	48.1	**77.3**	**69.3**	53.6	**82.0**	**65.6**

Table 3. Accuracy(%) on Office-Home dataset with Resnet-50 as base network and comparison with the state-of-the-art methods.

	Source Target	CIFAR-10 STL	STL CIFAR-10
Source Only		60.35	51.88
w/o augmentation			
DANN [10]		66.12	56.91
DRCN [11]		66.37	58.65
AutoDIAL [3]		79.10	70.15
DWT		**79.75**±0.25	**71.18**±0.56
Target Only		67.75	88.86
w/ augmentation			
SE [a] [7]		77.53±0.11	71.65±0.67
SE [b] [7]		80.09±0.31	69.86±1.97
DWT-MEC[b]		80.39±0.31	**72.52**±0.94
DWT-MEC (MT)[b]		**81.83**±0.14	71.31±0.22

Table 4. Accuracy (%) on the CIFAR-10↔STL: comparison with state of the art. [a] indicates minimal data augmentation and [b] considers augmented source and target data.

methods [35] are more effective. Looking at methods which consider data augmentation, we compare our approach with SE [7]. To be consistent with other methods, we plug the architectures described in [9] in SE. Comparing the proposed approach with our re-implementation of SE (SE[†b]) we observe that DWT-MEC (MT) is almost on par with SE in the MNIST↔USPS setting and better than SE in the SVHN→MNIST. For the sake of completeness, we also report the performance of SE taken from the original paper [7], considering SE with minimal augmentation (only gaussian blur) and SE with full augmentation.

With the rapid progress of deep DA methods, the results in the digits datasets have saturated. This makes it difficult to gauge the merit of the proposed contributions. Therefore, we also consider the CIFAR10 ↔ STL setting. Our results are reported in Tab. 4. Similarly to the experiments in Tab. 1, we separate those methods exploiting data augmentation from those not using target-sample perturbations. Tab. 4 shows that our method (DWT), outperforms all previous baselines which also do not consider augmentation. Furthermore, by exploiting data perturbation and the

proposed MEC loss our approach (with and without Mean-Teacher) reaches higher accuracy than SE.[2]

Finally, we also perform experiments on the large-scale Office-Home dataset and we compare with the baselines methods as reported by Long *et al.* [26]. The results reported in Tab. 3 show that our approach outperforms all the other methods. On average, the proposed approach improves over Conditional Domain Adversarial Networks (CDAN) by 2.8% and it is also more accurate than SE.

5. Conclusions

In this work we address UDA by proposing domain-specific feature whitening with DWT layers and the MEC loss. On the one hand, whitening of intermediate features enables the alignment of the source and the target distributions at intermediate feature levels and increases the smoothness of the loss landscape. On the other hand, our MEC loss better exploits the target data. Both these components can be easily integrated in any standard CNN. Our experiments on standard benchmarks show state-of-the-art performance on digits categorization and object recognition tasks. As future work, we plan to extend our method to handle multiple source and target domains.

---

[2]In this case the accuracy values reported for SE are taken directly from the original paper as the underlying network architecture is the same.

References

[1] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with gans. In *CVPR*, 2017.

[2] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *NIPS*, 2016.

[3] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò. Autodial: Automatic domain alignment layers. In *ICCV*, 2017.

[4] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011.

[5] G. Csurka, editor. *Domain Adaptation in Computer Vision Applications*. Advances in Computer Vision and Pattern Recognition. Springer, 2017.

[6] D. Dereniowski and K. Marek. Cholesky factorization of matrices in parallel and ranking of graphs. In *5th Int. Conference on Parallel Processing and Applied Mathematics*, 2004.

[7] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for visual domain adaptation. *ICLR*, 2018.

[8] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. 2001.

[9] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *ICML*, 2015.

[10] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

[11] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*, 2016.

[12] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, 2004.

[13] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers. Associative domain adaptation. In *ICCV*, 2017.

[14] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers. Associative domain adaptation. In *ICCV*, 2017.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[16] H. Huang, Q. Huang, and P. Krahenbuhl. Domain transfer through deep activation matching. In *ECCV*, 2018.

[17] L. Huang, D. Yang, B. Lang, and J. Deng. Decorrelated batch normalization. In *CVPR*, 2018.

[18] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[19] A. Kessy, A. Lewin, and K. Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 2017.

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv :1412.6980*, 2014.

[21] J. Kohler, H. Daneshmand, A. Lucchi, M. Zhou, K. Neymeyr, and T. Hofmann. Towards a Theoretical Understanding of Batch Normalization. *arXiv:1805.10694*, 2018.

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

[23] Y. LeCun, L. Bottou, G. B. Orr, and K. Müller. Efficient backprop. In *Neural Networks: Tricks of the Trade - Second Edition*, pages 9–48. 2012.

[24] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou. Revisiting batch normalization for practical domain adaptation. *arXiv:1603.04779*, 2016.

[25] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *NIPS*, 2016.

[26] M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. *arXiv:1705.10667v2*, 2018.

[27] M. Long and J. Wang. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

[28] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. *ICML*, 2017.

[29] M. Mancini, S. R. Bulò, B. Caputo, and E. Ricci. Adagraph: Unifying predictive and continuous domain adaptation through graphs. In *CVPR*, 2019.

[30] M. Mancini, H. Karaoguz, E. Ricci, P. Jensfelt, and B. Caputo. Kitting in the wild through online domain adaptation. *IROS*, 2018.

[31] M. Mancini, L. Porzi, S. R. Bulò, B. Caputo, and E. Ricci. Boosting domain adaptation by discovering latent domains. *CVPR*, 2018.

[32] P. Morerio, J. Cavazza, and V. Murino. Minimal-entropy correlation alignment for unsupervised deep domain adaptation. *ICLR*, 2018.

[33] 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.

[34] S. J. Pan, Q. Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[35] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo. From source to target and back: symmetric bi-directional adaptive gan. In *CVPR*, 2018.

[36] H. N. S. Wiesler. A convergence analysis of log-linear training. In *NIPS*, 2011.

[37] K. Saito, Y. Ushiku, and T. Harada. Asymmetric tri-training for unsupervised domain adaptation. *arXiv:1702.08400*, 2017.

[38] M. Sajjadi, M. Javanmardi, and T. Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *NIPS*, 2016.

[39] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, 2018.

[40] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. *arXiv:1612.07828*, 2016.

[41] R. Shu, H. H. Bui, H. Narui, and S. Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.

[42] A. Siarohin, E. Sangineto, and N. Sebe. Whitening and Coloring transform for GANs. *arXiv:1806.00420*, 2018.

[43] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, 2016.

[44] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. *ECCV*, 2016.

[45] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. *ICLR*, 2017.

[46] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*, 2017.

[47] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

[48] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015.

[49] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

[50] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

[51] X. Zhu. Semi-supervised learning literature survey. 2005.