Improving Transferability of Domain Adaptation Networks Through Domain Alignment Layers

Lucas F. A. Silva[1], Daniel C. G. Pedronette[2], Fabio A. Faria[1], João P. Papa[3], and Jurandy Almeida[1]

[1]*Instituto de Ciência e Tecnologia, Universidade Federal de São Paulo - UNIFESP, São José dos Campos, SP – Brazil*
Email: {e.lucas, ffaria, jurandy.almeida}@unifesp.br

[2]*Dept. of Statistics, Applied Mathematics and Computing, São Paulo State University - UNESP, Rio Claro, SP – Brazil*
Email: daniel.pedronette@unesp.br

[3]*Dept. of Computing, São Paulo State University - UNESP, Bauru, SP – Brazil*
Email: joao.papa@unesp.br

***Abstract*—Deep learning (DL) has been the primary approach used in various computer vision tasks due to its relevant results achieved on many tasks. However, on real-world scenarios with partially or no labeled data, DL methods are also prone to the well-known domain shift problem. Multi-source unsupervised domain adaptation (MSDA) aims at learning a predictor for an unlabeled domain by assigning weak knowledge from a bag of source models. However, most works conduct domain adaptation leveraging only the extracted features and reducing their domain shift from the perspective of loss function designs. In this paper, we argue that it is not sufficient to handle domain shift only based on domain-level features, but it is also essential to align such information on the feature space. Unlike previous works, we focus on the network design and propose to embed Multi-Source version of DomaIn Alignment Layers (MS-DIAL) at different levels of the predictor. These layers are designed to match the feature distributions between different domains and can be easily applied to various MSDA methods. To show the robustness of our approach, we conducted an extensive experimental evaluation considering two challenging scenarios: digit recognition and object classification. The experimental results indicated that our approach can improve state-of-the-art MSDA methods, yielding relative gains of up to +30.64% on their classification accuracies.**

I. Introduction

Machine learning algorithms, mainly the ones based on deep learning (DL), have been the first approach to tackle a wide variety of problems, for they have achieved outstanding results and demonstrated a high generalization capacity on well-known benchmarks [1]–[6]. However, when applied to non-controlled environments, usually faced with real-world problems, DL methods can have their performance degraded to some extent. The first issue is that many real-world applications have no or few labeled data available in real-time [1], [7], [8]. The most logical but naive approach concerns using similar sets of annotated data as a source to train the predictor for further inferring the unlabeled target data. However, even similar data can be drawn from different underlying distributions, *i.e.*, they are obtained under different circumstances, exhibiting a phenomenon called *domain shift* [1], [7], [9].

The aforementioned shortcomings have been firstly tackled by the *unsupervised domain adaptation* (UDA) paradigm to adapt a single source domain to a single target domain on shallow machine learning methods, later extended to DL methods [2], [9]. UDA approaches, in general, can be divided into two main groups: (*i*) methods that try to learn domain invariant features and (*ii*) methods that minimize domain discrepancy, in a way that regularization terms are added to the loss function [3]. However, often it is possible to find more than one correlated dataset to the target task (*e.g.*, digit recognition) [2], [9]. Additionally, it is necessary to consider the fact that data are not usually extracted from a single source of information, and can present as many as possible domains even within a same dataset, the so-called *latent domains* [10]. A common strategy to deal with more than one domain is to group them into a single source domain [2], [11], which does not always hold effective results, once distinct domains may contribute in different ways to the transfer of knowledge.

The *multi-source unsupervised domain adaptation* (MSDA) has been proposed to mitigate the weakness of UDA methods by adapting a finite number of source domains to a single target domain. Indeed, this goal is more challenging, since we must deal with a shift between the source distributions. Each source domain can contribute with complementary information to fashion the knowledge about the target domain. Typically, they do not share the same label space [2]. To reduce domain shift, most MSDA methods try to learn domain-invariant representations by introducing appropriate loss terms to penalize the discrepancy of features across domains [2], [8], [9], [11].

Although such methods are quite effective, we argue that it is not enough to deal with domain shift only based on the loss function and feature alignment at different levels of the network is also an indispensable part of domain adaptation. Different from previous works based on designing loss functions solely, we focus on improving the transferability of the network by redesigning its architectural components. For this, we propose to embed Multi-Source version of DomaIn Alignment Layers (MS-DIAL) [12] at different levels of any given DL model. These layers are designed to match the feature distributions between different domains and can be easily applied to various MSDA methods.

Experiments were conducted on two challenging tasks using six popular benchmarks widely-used to evaluate MSDA methods: (*i*) digit recognition using MNIST [13], MNIST-M [14], SVHN [15], and *Synthetic Digits* [16] datasets; and (*ii*) object classification on Office-31 [17] and Office-Home [18] datasets. We evaluated five different state-of-the-art MSDA methods and compared their results with and without using MS-DIAL. The obtained results indicated that our approach yields significant effectiveness gains, reaching up to +30.64% of relative gains on classification accuracies of the state-of-the-art MSDA methods. In addition, a visual analysis was conducted in order to highlight the benefits of our approach.

The remainder of this paper is organized as follows. Section II discusses related work. Section III describes our approach and shows how it can be used for improving the transferability of any MSDA model. Section IV presents the experimental setup and reports our results. Finally, we offer our conclusions and directions for future work in Section V.

II. Related Work

UDA paradigm aims to transfer knowledge from a labeled source domain to an unlabeled target domain in order to learn a model well-performing on the target distribution. Early UDA works have focused on shallow machine learning (ML) methods but later they have shifted to DL methods [2]. On shallow methods, the main approaches rely on minimizing the discrepancy between the target domain and the source domain in order to obtain domain-invariant features, like the Transfer Component Analysis (TCA) [19] and Distribution-Matching Embeddings (DME) [20].

A common approach for DL methods consists in exploiting adversarial training to either transform source samples towards target samples or to overpower the feature extractor to fool the classifier by outputting features that are closer to the target domain, like Domain-Adversarial Neural Networks (DANN) [16] and Weighted Maximum Mean Discrepancy (WMMD) [21]. Other methods try to align feature distributions by embedding domain-specific normalization layers into a neural network, such as DIAL [3] and AutoDIAL [22].

The aforementioned UDA methods try to reduce the domain shift from a single-source to a single-target. MSDA methods, on the other hand, face a challenging scenario of adopting more than one source domain to predict a single target domain. In the last years, many different MSDA approaches have been proposed in the literature, such as Moment Matching for Multi-Source Domain Adaptation (M3SDA) [9], Deep CockTail Network (DCTN) [2], Multiple Domain Matching Network (MDMN) [23], Multi-Source Domain Adversarial Networks (MDAN) [24] and Domain Aggregation Networks (DARN) [11], Adversarial Domain Aggregation Networks (MADAN) [1], MultiDIAL [7], and MS-DIAL [12].

Except for these last two approaches, all the other MSDA works employ a multi-stream network, usually with one stream for each domain, some with different parameters for each stream and, in this case, each domain has an independent feature extractor and classifier; and others with parameters shared among the streams, usually by feature extractors from all domains, but each having its own classifier. Usually, one stream is used to represent the task model and the others are used to align the target and source domains. In this way, a traditional task loss based on the labeled data and another alignment loss to tackle the domain shift problem are jointly optimized during the training phase.

In this paper, we demonstrate that most of those MSDA methods can be improved by not just optimizing additional loss terms but also aligning feature distributions at different levels of the network. For this, we use MS-DIAL [12], an extended version of DIAL [3] for MSDA.

III. DIAL: DomaIn Alignment Layers

Let $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_M\}$ be a finite set of labeled source domains sharing the same set $\mathcal{Y}$ of categories with an unlabeled target domain $\mathcal{T}$. Each source domain $\mathcal{S}_i = \{(\mathbf{x}_i^j, \mathbf{y}_i^j)\}_{j=1}^{N_i}$ refers to a set of tuples composed of N_i samples $\mathbf{x}_i^j$ and their respective labels $\mathbf{y}_i^j$. Since we do not know the labels of the target domain beforehand, the set $\mathcal{T} = \{\mathbf{x}_T^j\}_{j=1}^{N_T}$ comprises the target samples only.

The source and target samples are drawn from distinct distributions. The distribution of the source domains can be estimated from the set of source samples and their labels. On the other hand, such information is unknown concerning the target domain. The final goal is to learn a function $f(\mathbf{x}_T; \theta)$ defined by a set of parameters θ, whose input are the target samples $\mathbf{x}_T \in \mathcal{T}$. Such a function is expected to approximate the unknown target distribution and better classify samples in $\mathcal{T}$. This definition also holds to the single-source to single-target UDA problem by limiting the number of source domains, i.e., $M = 1$.

To solve this problem for a single source domain (*i.e.*, $M = 1$), Carlucci *et al.* [3] proposed to align feature distributions at different levels of a neural network through DomaIn Alignment Layers (DIAL). DIAL require one model only, shared across all domains, where each layer tries to bring all domain distributions to a canonical superposed one and, subsequently, to make few minor adjustments by performing a jointly linear transformation on all distributions. In this work, we adopt an extended version of DIAL that generalizes it for multiple source domains (*i.e.*, $M > 1$), called MS-DIAL [12].

MS-DIAL are based on Batch Normalization (BN) [25] layers, which normalizes batches based on the estimation of their first and second order statistics (*i.e.*, mean and variance). BN layers also use additional parameters to linearly transform the normalized features and let the model restore the batch representation. MS-DIAL aligns all domain distributions to a canonical distribution, forwarding samples through a respective BN layer that is individual to each domain without performing the affine transformation. Finally, after the normalization process, MS-DIAL performs an affine transformation.

Note that MS-DIAL act in different ways on the training step and in the inference mode. We need to ensure that batches will comprise samples from all domains organized in the same order for training purposes. Concerning the inference mode,

as we are interested in classifying samples from the target domain only, they are forwarded to their respective BN layer, driving MS-DIAL to use BN simply.

During training, source (and labeled) domain samples jointly with target (unlabeled) domain samples are fed to the neural network. The loss function of our model is a weighted sum of a classification term (Equation 1) and a distribution alignment term (Equation 2). The classification term refers to the well-known cross-entropy loss function between the source predictions and their labels, as follows:

$$\mathcal{L}_{\mathcal{S}}(\theta) = -\sum_{i=1}^{M}\sum_{k=1}^{N_i} \mathbf{y}_i^k \log f_i(\mathbf{x}_i^k;\theta), \tag{1}$$

where $f_i(\mathbf{x}_i^k;\theta)$ denotes the prediction function concerning source domain $\mathcal{S}_i$.

The distribution alignment term minimizes the entropy of target samples in order to force the model to decide more confidently and is given by:

$$\mathcal{L}_{\mathcal{T}}(\theta) = -\sum_{k=1}^{N_T} f_T(\mathbf{x}_T^k;\theta) \log f_T(\mathbf{x}_T^k;\theta), \tag{2}$$

in which $f_T(\mathbf{x}_T^k;\theta)$ denotes the prediction function for the target domain. The final loss function is computed as follows:

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathcal{S}}(\theta) + \lambda\mathcal{L}_{\mathcal{T}}(\theta), \tag{3}$$

where λ weights the distribution alignment term.

We follow different strategies to include MS-DIAL on off-the-shelf methods: if the model has BN layers, we replace them by MS-DIAL and transfer their parameters to the new BN layers inside MS-DIAL; otherwise, if the model has no BN layers, we embed MS-DIAL after each convolutional and fully-connected layer to perform the feature alignment on each new representation, except on its final classification layer. This process is better described in Algorithm 1.

Algorithm 1: Automatic MS-DIAL Insertion

Input: DL model without MS-DIAL
Output: DL model with MS-DIAL

```
/* The loop below iterates through
   all layers to verify if they
   contain BN layers.                  */
if backbone model has BN layers then
    foreach layer l of the backbone model do
        if l is BN Layer then
            Replace it by MS-DIAL;
        end
    end
else
    foreach layer l of the backbone model do
        if l is a convolutional layer then
            Replace it by a building block formed by
              the same convolutional layer but now
              followed by MS-DIAL;
        end
        else if l is a fully-connected layer then
            Replace it by a building block formed by
              the same fully-connected layer but now
              followed by MS-DIAL;
        end
    end
end
/* The affine parameters of the
   original BN layers, if present,
   are copied to MS-DIAL.              */
```

IV. Experiments and Results

In this section, we provide details about the experimental setup adopted in order to evaluate the proposed method as well as we report the obtained results. A rigorous and extensive experimental evaluation was conducted on both small and large-scale datasets and the performance of state-of-the-art approaches was compared with and without using our approach.

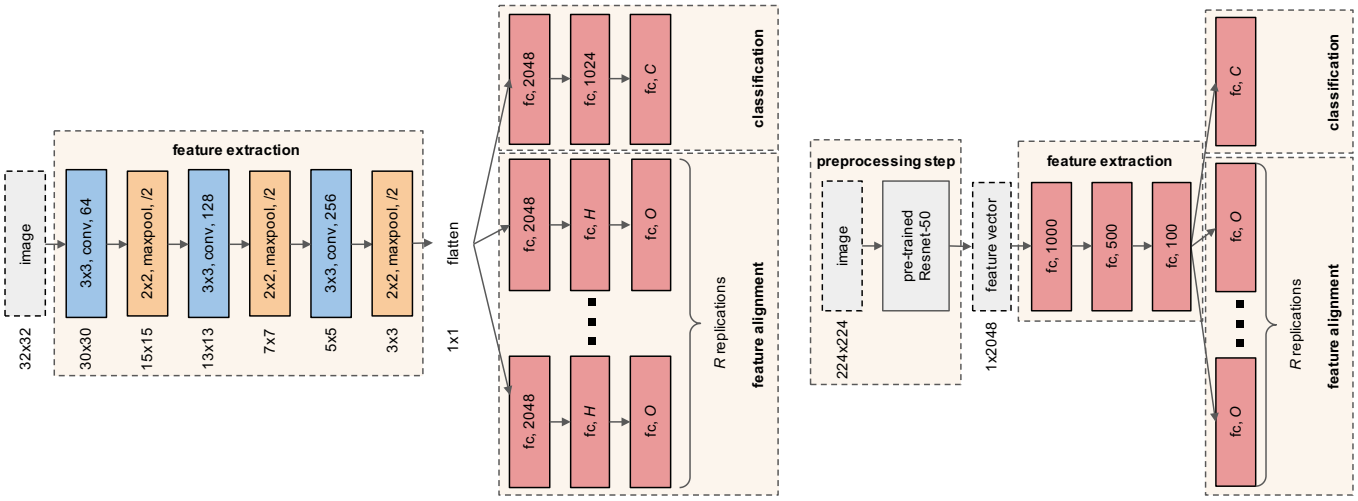
A. Datasets

The digit recognition was carried out on four different datasets, each of them related to a distinct domain, namely MNIST [13], MNIST-M [14], SVHN [15], and Synth [16]. MNIST contains 70,000 grayscale images of handwritten digits with a resolution of 28x28 pixels, of which 60,000 are for training and 10,000 for testing. MNIST-M [14] is composed by 59,001 training and 9,001 testing images with a resolution of 32x32 pixels obtained by modifying MNIST images with colored patches randomly extracted from the BSDS500 dataset [28]. The Street View House Number (SVHN) [15] dataset contains 73,257 training and 26,032 testing images, in RGB color and size 32x32, of house numbers collected from Google Street View. Synthetic Digits (Synth) [16] is a collection of 479,400 training and 9,553 testing images with size 32x32 generated from Windows fonts by varying position, orientation and background. All these datasets contain 10 categories related to digits from '0' to '9'.

The object classification was conducted on the Office-31 [17] and Office-Home [18] datasets. Office-31 [17] is a standard benchmark in the MSDA literature. This dataset is composed of 4,652 images collected from Amazon.com or taken from an office environment using a Webcam or a DSLR camera and with varying lighting and pose changes. Those images comprise 31 classes from three different domains: Amazon, DSRL, and Webcam. Office-Home [18] is a large-scale benchmark widely-used for testing MSDA methods. This dataset is composed of 15,500 images collected from several search engines and online image directories. They are distributed among 65 object categories and divided into 4 distinct domains: Art, Clipart, Product, and Real World.

B. Experimental Protocol

An essential point concerning DL works is their reproducibility, since most methods are highly affected by non-



(a) Digit recognition (b) Object Classification

Fig. 1. Deep learning models used for each task, that are: (a) digit recognition, which, rvexcept for the M3SDA, is composed by a feature extractor with three convolutional layers and a classifier with three fully-connected (FC) layers; and (b) object classification, where pre-computed features are extracted by a ResNet-50 [26] model pre-trained on the ImageNet [27] dataset and passed through four FC layers, in which the first three are for feature learning and the last is a classifier. After feature extraction, the network is splitted into $R + 1$ branches, where one performs classification and the others are for feature alignment. The number C of neurons in the output layer of the branch used for classification denotes the number of classes, which is 10 for digit recognition, 65 for object classification on the Office-Home dataset or 31 on the Office-31 dataset. The number R of branches used for feature alignment is the number of source domains (M) for DARN and MDAN; or 1 for DANN, MDMN, and M3SDA. The number H of neurons in the hidden layer of such branches is 2048 for DARN, MDAN, DANN, and MDMN; or 1024 for M3SDA. The number O of neurons in the output layer of such branches is the number of domains ($M + 1$) for MDMN; the number classes (C) for M3SDA; or 1 for DARN, MDAN, and DANN. Except for the output layers of all the branches, dropout is applied before each convolutional or FC layer, which are followed by ReLU activation.

deterministic factors [4]. A good practice is to re-run the source code with the parameters presented on the paper and compare its results with those obtained by the authors.

For reproducibility purposes, our approach was implemented in PyTorch (version 1.4.0) upon the DARN [11] implementation[1], whose source code is publicly available along with re-implementations of four state-of-the-art MSDA methods: DANN [16], MDAN [24], M3SDA [9], and MDMN [23]. Since our intent is to evaluate whether or not aligning domains on feature space may benefit existing MSDA methods, we first ran the original code and performed all its experiments, as suggested by Bucci *et al.* [4]. Next, we modified the original code with as minimum changes as possible in order to embed MS-DIAL into all the aforesaid methods[2]. Last but not least, we evaluated the use of MS-DIAL for MSDA on two different tasks: (*i*) digit recognition and (*ii*) object classification.

For a fair comparison, we adopt the same experimental protocol used by DARN [11]. In each experiment, one domain was chosen as the target and the rest was used as source domains. This process was repeated several times, each time with a different domain as the target. Twenty replications were performed for each experiment in order to ensure statistically sound results. The reported results refer to the mean and standard error of the classification accuracies measured at the end of each experiment for all the replications.

We evaluated five different state-of-the-art MSDA methods, namely: **DANN** [16], **M3SDA** [9], **MDAN** [24], **MDMN** [23], and **DARN** [11]. The purpose of our experiments is to compare their results with and without using MS-DIAL. For reference, we report the results obtained using only **MS-DIAL** [12]. Also, two additional baselines were considered: **SRC**, where a model was trained on a large set formed by merging all the source domains; and **TAR**, where a model was trained only on the target domain but having access to its true labels. They can be seen as lower (SRC) and upper (TAR) bounds for the results that can be achieved by MSDA methods.

For digit recognition, we randomly chose 20000 training samples as the training set and 9000 testing samples as the testing set for each domain. The network architecture used in such experiments is shown in Figure 1a. Roughly speaking, it is a Convolutional Neural Network (CNN) with three convolutional layers as a feature extractor on top of which is stacked a classifier with three fully-connected (FC) layers, whose the output layer has 10 neurons, each corresponding to a particular digit. In addition to the classifier, three streams are used for feature alignment by DARN and MDAN; or only one stream in the case of DANN, MDMN, and M3SDA. These streams have a different amount of neurons depending on the MSDA method used, with 2048 neurons in the hidden layer and 1 in the output layer in the case of DANN, MDAN, and DARN; 2048 neurons in the hidden layer and 4 in the output layer for MDMN; and 1024 neurons in the hidden layer and 10 in the output layer for M3SDA.

For object classification, pre-computed features were first extracted by a ResNet-50 [26] model pre-trained on the Ima-

[1]https://github.com/junfengwen/DARN (As of July, 2021)

[2]Our code is available at https://github.com/LucasFernando-aes/MS-DIAL

geNet [27] dataset. For this, we follow the same preprocessing step performed by He *et al.* [26] to feed ResNet-50 models. During training, each image was resized to 256 pixels on its shortest side, and then a 224x224 crop was randomly sampled from the resulting image or its horizontal flip, normalizing it to the mean and standard deviation of the ImageNet dataset. In the inference mode, we first rescaled the image such that its shorter side was of length 256, then cropped out the central 224x224 patch from the resulting image and normalized it to the mean and standard deviation of the ImageNet dataset. The preprocessed images were passed through the network and 2048-dimensional feature vectors were extracted from its average pooling layer. These feature vectors were passed through four FC layers, as shown in Figure 1b. The first three FC layers are for feature learning and have 1000, 500, and 100 neurons, respectively. The last layer is a classifier and has 31 output logits in the case of the Office-31 dataset and 65 for the Office-Home dataset. For feature alignment, three streams were used by DARN and MDAN; or only one stream in the case of DANN, MDMN, and M3SDA. Each stream corresponds to single FC layer whose size is 1 for DARN, MDAN, and DANN; 4 for MDMN; and the number of classes for M3SDA (*i.e.*, 31 for Office-31 or 65 for Office-Home).

Except for the output layers, dropout is applied before each convolutional or FC layer, which are followed by ReLU activation. All the models were trained from scratch for 50 epochs using the Adadelta [29] optimizer with a learning rate of 1.0. We used a mini-batch size of 128 for digit recognition and 32 for object classification. The hyper-parameters adopted for each MSDA method are the same as described in [11].

C. Ablation Study

The key advantage of MS-DIAL is to align feature distributions. To improve the model's confidence on unsupervised target samples, an entropy penalty is included in the loss function. If the entropy penalty is significant, the model may learn to make wrong predictions in order to minimize errors.

A critical part of MS-DIAL is to set the λ hyper-parameter in order to balance entropy penalties with classification errors. The former leads to more confident predictions for target samples whereas the latter takes knowledge from source domains.

To assess the impact of the λ hyper-parameter, we tested DARN with MS-DIAL on digit recognition task using five different values for the λ hyper-parameter (i.e., $\lambda = \{0.001, 0.005, 0.01, 0.05, 0.1\}$) and compared its average accuracy for all the target domains, as shown in Figure 2. Notice that small values for the λ hyper-parameter are better and, for this reason, we used $\lambda = 0.001$ in the next experiments.

D. Quantitative Results

Table I presents the results obtained for digit recognition. When MNIST and Synth are used as target domain, probably the former by its simplicity (gray-scale images) and the latter because it is synthetically generated, we can observe a slight gain of performance between the original and MS-DIAL version, showing that the evaluated methods already can achieve

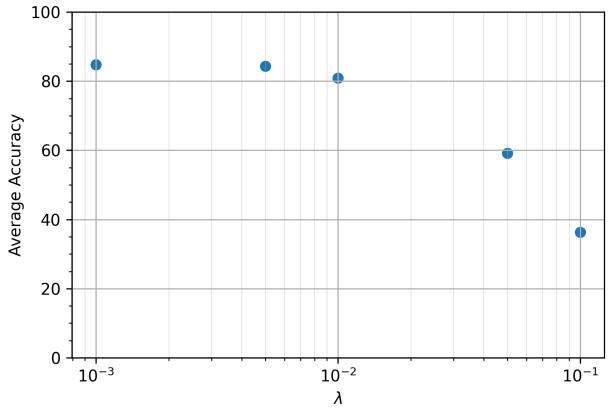


Fig. 2. Average accuracy results for different values of the λ hyper-parameter.

good results for these adaptation tasks. On the other hand, it does not hold for more difficult domains, presenting a bigger domain shift between them, like MNIST-M and SVHN, which have random patterns and colors and digit images coming from real world data, respectively. When MS-DIAL are used, we can clearly see a gain of performance, even for DANN, which is a single-source to single-target UDA method.

Tables II and III present the results obtained for object classification on the Office-31 and Office-Home datasets, respectively. For Office-31, the results obtained for the DSLR and Webcam domains were better than those for the Amazon domain. However, in all cases, the use of MS-DIAL improved the performance of all the MSDA methods by a consistent margin, reaching, on average, +3% of relative gains on the classification accuracy. For Office-Home, as expected, the Clipart domain is the most difficult to be adapted from the other source domains, because it usually is a mix of digital artistic concepts with simplistic elements, making its results the worst among all the domains. However, we can also see improvements of almost +13% for DANN and 11% for M3SDA with Clipart as target domain, and of almost +16% for DANN and +15% for M3SDA with Product as target domain, just by embedding MS-DIAL. As we can observe, the use of MS-DIAL for aligning feature distributions can improve all the MSDA methods by a large margin, specially single-source to single-target UDA methods, like DANN, for which was obtained a relative gain of, on average, +30.64% on its classification accuracy.

Finally, we can clearly notice that MS-DIAL is complementary to all the other MSDA methods, since for all the evaluated tasks its combination yielded superior results to those obtained by each method in isolation. For digit recognition, for instance, we can see improvements of, on average, +3% in the results of M3SDA w/ MS-DIAL in relation to those of only MS-DIAL.

E. Visualization Analysis

In order to enrich the discussion about the proposed approach, we employed dimensionality reduction methods to represent the impact of the proposed method on a 2-D projection

TABLE I
CLASSIFICATION ACCURACY (%) FOR DIGIT RECOGNITION.

Methods		Domains				Average	Relative Gain
		MNIST	MNIST-M	SVHN	Synth		
SRC		96.48 ± 0.12	60.44 ± 0.41	68.32 ± 1.20	83.73 ± 0.31	77.24 ± 0.29	-
DANN [16]	w/o MS-DIAL	96.70 ± 0.11	61.36 ± 0.37	67.42 ± 1.80	84.08 ± 0.32	77.39 ± 0.52	+5.71%
	w/ MS-DIAL	97.65 ± 0.08	64.85 ± 0.14	80.57 ± 0.37	84.15 ± 0.15	81.81 ± 0.11	
M3SDA [9]	w/o MS-DIAL	96.44 ± 0.09	65.13 ± 0.29	76.96 ± 0.62	81.78 ± 0.26	80.08 ± 0.02	+7.07%
	w/ MS-DIAL	98.43 ± 0.03	72.58 ± 0.15	84.71 ± 0.29	88.98 ± 0.13	86.17 ± 0.08	
MDAN [24]	w/o MS-DIAL	97.27 ± 0.09	64.83 ± 0.30	76.51 ± 0.73	85.72 ± 0.14	81.03 ± 0.21	+3.04%
	w/ MS-DIAL	98.06 ± 0.07	69.03 ± 0.12	80.82 ± 0.40	86.04 ± 0.13	83.49 ± 0.10	
MDMN [23]	w/o MS-DIAL	97.21 ± 0.09	63.14 ± 0.33	76.74 ± 0.67	85.78 ± 0.17	80.72 ± 0.21	+4.14%
	w/ MS-DIAL	98.47 ± 0.04	68.48 ± 0.19	82.01 ± 0.70	87.29 ± 0.12	84.06 ± 0.20	
DARN [11]	w/o MS-DIAL	97.96 ± 0.03	67.97 ± 0.20	78.37 ± 0.34	86.61 ± 0.19	82.73 ± 0.12	+0.58%
	w/ MS-DIAL	97.94 ± 0.05	69.26 ± 0.33	79.21 ± 0.79	86.41 ± 0.12	83.21 ± 0.23	
MS-DIAL [12]		98.45 ± 0.03	68.32 ± 0.19	81.82 ± 0.54	87.13 ± 0.08	83.93 ± 0.13	-
TAR		99.04 ± 0.02	94.83 ± 0.08	87.48 ± 0.20	97.01 ± 0.06	94.59 ± 0.05	-

TABLE II
CLASSIFICATION ACCURACY (%) FOR OBJECT CLASSIFICATION ON THE OFFICE-31 DATASET.

Methods		Domains			Average	Relative Gain
		Amazon	DSLR	Webcam		
SRC		65.50 ± 0.23	93.92 ± 0.55	92.22 ± 0.28	83.88 ± 0.21	-
DANN [16]	w/o MS-DIAL	66.02 ± 0.22	92.27 ± 0.62	93.06 ± 0.21	83.79 ± 0.21	+2.37%
	w/ MS-DIAL	67.00 ± 0.21	95.85 ± 0.45	94.48 ± 0.20	85.78 ± 0.19	
M3SDA [9]	w/o MS-DIAL	65.34 ± 0.23	91.02 ± 0.69	91.06 ± 0.34	82.48 ± 0.28	+4.54%
	w/ MS-DIAL	67.01 ± 0.19	96.93 ± 0.43	94.74 ± 0.26	86.23 ± 0.18	
MDAN [24]	w/o MS-DIAL	66.53 ± 0.23	92.84 ± 0.55	92.17 ± 0.37	83.85 ± 0.25	+2.39%
	w/ MS-DIAL	67.37 ± 0.20	95.57 ± 0.49	94.62 ± 0.23	85.85 ± 0.22	
MDMN [23]	w/o MS-DIAL	63.03 ± 0.22	93.64 ± 0.52	92.84 ± 0.33	84.17 ± 0.24	+2.69%
	w/ MS-DIAL	67.94 ± 0.14	96.76 ± 0.50	94.60 ± 0.19	86.43 ± 0.18	
DARN [11]	w/o MS-DIAL	65.82 ± 0.33	93.69 ± 0.54	93.26 ± 0.30	84.26 ± 0.21	+2.59%
	w/ MS-DIAL	67.72 ± 0.18	96.93 ± 0.39	94.66 ± 0.26	86.44 ± 0.16	
MS-DIAL [12]		67.64 ± 0.18	96.08 ± 0.49	94.64 ± 0.25	86.12 ± 0.18	-
TAR		73.64 ± 0.32	95.34 ± 0.46	91.73 ± 0.46	86.90 ± 0.22	-

of feature space. The analysis was performed on all the digits datasets, using the well-known UMAP [30] algorithm. Due to space limitations, we chose to report the results only for MDMN, comparing the projections produced with and without using MS-DIAL. However, similar results were observed for all the MSDA methods.

Figure 3 presents the visualizations of the application of UMAP on the digits datasets. In most situations, an evident increase in the separability among classes can be observed. We can highlight the results for MNIST, presented in the first row, where the class representations are clearly apart when using the proposed approach.

V. CONCLUSIONS

In this paper, we focused on improving the transferability of DL models with simple and efficient network layers easily pluggable into the network backbones of existing MSDA methods. More specifically, we proposed to embed MS-DIAL [12] at different levels of any given DL model. The main contribution of MS-DIAL is to perform the feature alignment on each new representation generated along the network.

Our approach was validated on digit recognition and object classification tasks. Six popular benchmarks widely-used to evaluate MSDA methods and five different state-of-the-art MSDA methods were considered in our experiments. Extensive experimental results demonstrated that MS-DIAL significantly boosts existing MSDA methods, leading to relative gains of up to +30.64% on their classification accuracies.

As future work, we intend to evaluate the use of MS-DIAL with other MSDA methods. Also, we want to augment MS-DIAL by exploring latent domain discovery strategies. In addition, we also plan to extend our ideas to other types of domain adaptation problems, like domain generalization.

TABLE III
CLASSIFICATION ACCURACY (%) FOR OBJECT CLASSIFICATION ON THE OFFICE-HOME DATASET.

Methods		Domains				Average	Relative Gain
		Art	Clipart	Product	Real World		
SRC		42.53 ± 0.53	29.53 ± 0.21	55.85 ± 0.36	63.43 ± 0.23	47.84 ± 0.15	-
DANN [16]	w/o MS-DIAL	42.68 ± 0.48	30.08 ± 0.38	55.68 ± 0.44	63.33 ± 0.33	47.94 ± 0.14	+30.64%
	w/ MS-DIAL	60.68 ± 0.47	43.83 ± 0.22	71.46 ± 0.21	74.54 ± 0.26	62.63 ± 0.14	
M3SDA [9]	w/o MS-DIAL	48.22 ± 0.48	33.90 ± 0.17	57.14 ± 0.43	63.27 ± 0.28	50.63 ± 0.16	+25.81%
	w/ MS-DIAL	62.11 ± 0.79	44.42 ± 0.19	72.65 ± 0.16	75.60 ± 0.15	63.70 ± 0.20	
MDAN [24]	w/o MS-DIAL	55.92 ± 0.56	38.65 ± 0.22	66.48 ± 0.19	71.28 ± 0.21	58.08 ± 0.17	+7.49%
	w/ MS-DIAL	60.75 ± 0.44	44.28 ± 0.22	70.66 ± 0.18	74.03 ± 0.16	62.43 ± 0.12	
MDMN [23]	w/o MS-DIAL	57.49 ± 0.43	38.66 ± 0.25	68.34 ± 0.28	72.52 ± 0.22	59.25 ± 0.17	+7.20%
	w/ MS-DIAL	62.76 ± 0.55	44.46 ± 0.20	71.91 ± 0.18	74.91 ± 0.14	63.51 ± 0.16	
DARN [11]	w/o MS-DIAL	57.59 ± 0.55	40.21 ± 0.23	69.40 ± 0.22	73.56 ± 0.24	60.19 ± 0.16	+4.84%
	w/ MS-DIAL	61.17 ± 0.38	42.84 ± 0.14	73.02 ± 0.16	75.38 ± 0.16	63.10 ± 0.12	
MS-DIAL [12]		62.25 ± 0.47	44.28 ± 0.15	72.27 ± 0.18	74.90 ± 0.21	63.42 ± 0.15	-
TAR		57.68 ± 0.54	42.39 ± 0.31	79.54 ± 0.26	73.58 ± 0.22	63.29 ± 0.23	-

ACKNOWLEDGMENTS

This research was supported by the São Paulo Research Foundation - FAPESP (grants 2013/07375-0, 2014/12236-1 2017/25908-6, 2018/15597-6, 2019/07665-4, and 2020/08770-3) and the Brazilian National Council for Scientific and Technological Development - CNPq (grants 307066/2017-7, 427968/2018-6, 309439/2020-5, and 314868/2020-8).

REFERENCES

[1] S. Zhao, B. Li, P. Xu, X. Yue, G. Ding, and K. Keutzer, "MADAN: multi-source adversarial domain aggregation network for domain adaptation," *IJCV*, vol. 129, no. 8, pp. 2399–2424, 2021.

[2] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin, "Deep cocktail network: Multi-source unsupervised domain adaptation with category shift," in *CVPR*, 2018, pp. 3964–3973.

[3] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò, "Just DIAL: domain alignment layers for unsupervised domain adaptation," in *Int. Conf. Image Analysis and Processing*, 2017, pp. 357–369.

[4] S. Bucci, M. R. Loghmani, and T. Tommasi, "On the effectiveness of image rotation for open set domain adaptation," in *ECCV*, 2020, pp. 422–438.

[5] S. F. Santos, N. Sebe, and J. Almeida, "CV-C3D: action recognition on compressed videos with convolutional 3d networks," in *SIBGRAPI – Conference on Graphics, Patterns and Images*, 2019, pp. 24–30.

[6] S. F. Santos and J. Almeida, "Faster and accurate compressed video action recognition straight from the frequency domain," in *SIBGRAPI – Conference on Graphics, Patterns and Images*, 2020, pp. 62–68.

[7] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulo, "Multidial: Domain alignment layers for (multisource) unsupervised domain adaptation," *TPAMI*, 2020.

[8] S. Zhao, G. Wang, S. Zhang, Y. Gu, Y. Li, Z. Song, P. Xu, R. Hu, H. Chai, and K. Keutzer, "Multi-source distilling domain adaptation," in *AAAI Conf. Artificial Intelligence*, 2020, pp. 12 975–12 983.

[9] X. Peng, Q. Bai, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *ICCV*, 2019, pp. 1406–1415.

[10] M. Mancini, L. Porzi, S. R. Buló, B. Caputo, and E. Ricci, "Inferring latent domains for unsupervised deep domain adaptation," *TPAMI*, vol. 43, no. 2, pp. 485–498, 2021.

[11] J. Wen, R. Greiner, and D. Schuurmans, "Domain aggregation networks for multi-source domain adaptation," in *ICML*, 2020, pp. 10 927–10 937.

[12] L. F. A. Silva and J. Almeida, "Ms-dial: Multi-source domain alignment layers for unsupervised domain adaptation," in *WVC*, 2020, pp. 111–116.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] Y. Ganin and V. S. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *ICML*, 2015, pp. 1180–1189.

[15] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NeurIPS*, 2011.

[16] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky, "Domain-adversarial training of neural networks," *JMLR*, vol. 17, pp. 59:1–59:35, 2016.

[17] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *ECCV*, 2010, pp. 213–226.

[18] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *CVPR*, 2017, pp. 5385–5394.

[19] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[20] M. Baktashmotlagh, M. T. Harandi, and M. Salzmann, "Distribution-matching embedding for visual domain adaptation," *JMLR*, vol. 17, pp. 108:1–108:30, 2016.

[21] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *CVPR*, 2017, pp. 945–954.

[22] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò, "Autodial: Automatic domain alignment layers," in *ICCV*, 2017, pp. 5077–5085.

[23] Y. Li, M. Murias, G. Dawson, and D. E. Carlson, "Extracting relationships by multi-domain matching," in *NeurIPS*, 2018, pp. 6799–6810.

[24] H. Zhao, S. Zhang, G. Wu, J. M. F. Moura, J. P. Costeira, and G. J. Gordon, "Adversarial multiple source domain adaptation," in *NeurIPS*, 2018, pp. 8568–8579.

[25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F.-F. Li, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[28] P. Arbelaez, M. Maire, C. C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *TPAMI*, vol. 33, no. 5, pp. 898–916, 2011.

[29] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012.

[30] L. McInnes and J. Healy, "UMAP: uniform manifold approximation and projection for dimension reduction," *CoRR*, vol. abs/1802.03426, 2018.

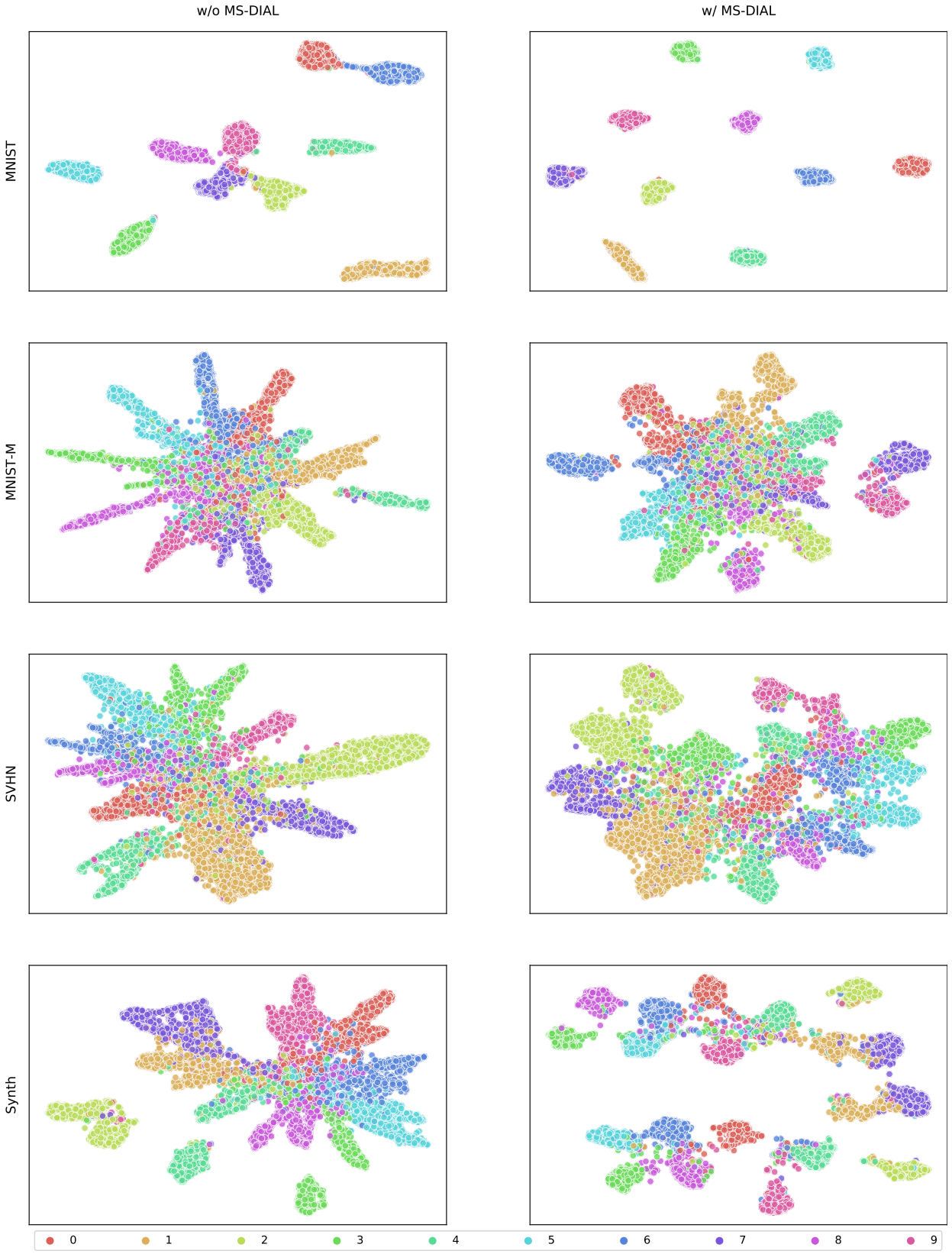


Fig. 3. UMAP [30] dimensional reduction of input features in final classification layers (100-dimensional vectors) to 2-dimensional vectors. Columns refer to, respectively, MDMN original method and MDMN adapted with MS-DIAL, on digit recognition task, with each row indicating the target domain.