This ECCV 2018 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ECCV 2018
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/eccv

DeepJDOT: Deep Joint Distribution Optimal Transport for Unsupervised Domain Adaptation

Bharath Bhushan Damodaran[1⋆], Benjamin Kellenberger[2⋆], Rémi Flamary[3], Devis Tuia[2], Nicolas Courty[1]

[1] Université de Bretagne Sud, IRISA, UMR 6074, CNRS, France
[2] Wageningen University, the Netherlands
[3] Université Côte d'Azur, OCA, UMR 7293, CNRS, Laboratoire Lagrange, France
{bharath-bhushan.damodaran@irisa.fr, benjamin.kellenberger@wur.nl}

Abstract. In computer vision, one is often confronted with problems of domain shifts, which occur when one applies a classifier trained on a source dataset to target data sharing similar characteristics (e.g. same classes), but also different latent data structures (e.g. different acquisition conditions). In such a situation, the model will perform poorly on the new data, since the classifier is specialized to recognize visual cues specific to the source domain. In this work we explore a solution, named DeepJDOT, to tackle this problem: through a measure of discrepancy on joint deep representations/labels based on optimal transport, we not only learn new data representations aligned between the source and target domain, but also simultaneously preserve the discriminative information used by the classifier. We applied DeepJDOT to a series of visual recognition tasks, where it compares favorably against state-of-the-art deep domain adaptation methods.

Keywords: Deep Domain Adaptation, Optimal Transport

1 Introduction

The ability to generalize across datasets is one of the holy grails of computer vision. Designing models that can perform well on datasets sharing similar characteristics such as classes, but also presenting different underlying data structures (for instance different backgrounds, colorspaces, or acquired with different devices) is key in applications where labels are scarce or expensive to obtain. However, traditional learning machines struggle in performing well out of the datasets (or *domains*) they have been trained with. This is because models generally assume that both training (or *source*) and test (or *target*) data are issued from the same generating process. In vision problems, factors such as objects position, illumination, number of channels or seasonality break this assumption and call for adaptation strategies able to compensate for such shifts, or *domain adaptation* strategies [1].

⋆ authors contributed equally

In a first rough subdivision, domain adaptation strategies can be separated into *unsupervised* and *semi-supervised* domain adaptation: the former assumes that no labels are available in the target domain, while the latter assumes the presence of a few labeled instances in the target domain that can be used as reference points for the adaptation. In this paper, we propose a contribution for the former, more challenging case. Let $\mathbf{x}^s \in \mathbb{X}^S$ be the source domain examples with the associated labels $y^s \in \mathbb{Y}^S$. Similarly, let $\mathbf{x}^t \in \mathbb{X}^T$ be the target domain images, but with unknown labels. The goal of the unsupervised domain adaptation is to learn the classifier f in the target domain by leveraging the information from the source domain. To this end, we have access to a source domain dataset $\{\mathbf{x}_i^s, y^s\}_{i=1,\dots,n^s}$ and a target domain dataset $\{\mathbf{x}_i^t\}_{i=1,\dots,n^t}$ with only observations and no labels.

Early unsupervised domain adaptation research tackled the problem as the one of finding a common representation between the domains, or a latent space, where a single classifier can be used independently from the datapoint's origin [2, 3]. In [4], the authors propose to use discrete optimal transport to match the shifted marginal distributions of the two domains under constraints of class regularity in the source. In [5] a similar logic is used, but the joint distributions are aligned directly using a coupling accounting for the marginals and the class-conditional distributions shift *jointly*. However, the method has two drawbacks, for which we propose solutions in this paper: 1) first, the JDOT method in [5] scales poorly, as it must solve a $n_1 \times n_2$ coupling, where n_1 and n_2 are the samples to be aligned; 2) secondly, the optimal transport coupling γ is computed between the input spaces (and using a ℓ_2 distance), which is a poor representation to be aligned, since we are interested in matching more semantic representations supposed to ease the work of the classifier using them to take decisions.

We solve the two problems above by a strategy based on deep learning. On the one hand, using deep learning algorithms for domain adaptation has found an increasing interest and has shown impressive results in recent computer vision literature [6–9]. On the other hand (and more importantly), a Convolutional Neural Network (CNN) offers the characteristics needed to solve our two problems: 1) by gradually adapting the optimal transport coupling along the CNN training, we obtain a scalable solution, an approximated and stochastic version of JDOT; 2) by learning the coupling in a deep layer of the CNN, we align the representation the classifier is using to take its decision, which is a more semantic representation of the classes. In summary, we learn jointly the embedding between the two domains and the classifier in a single CNN framework. We use a domain adaptation-tailored loss function based on optimal transport and therefore call our proposition *Deep Joint Distribution Optimal Transportation (DeepJDOT)*.

We test DeepJDOT on a series of visual domain adaptation tasks and compare favorably against several recent state of the art competitors.

2 Related works

Unsupervised domain adaptation. Unsupervised domain adaptation studies the situation where the source domain carries labeled instances, while the target domain is unlabeled, yet accessible during training [10]. Earlier approaches consider projections aligning data spaces to each other [2, 11, 12], thus trying to exploit shift-invariant information to match the domains in their original (input) space. Later works extended such logic to deep learning, typically by weight sharing [6]/reconstruction [13], by adding Maximum Mean Discrepancy (MMD) and association-based losses between source and target layers [14–16]. Other major developments focus on the inclusion of adversarial loss functions pushing the CNN to be unable to discriminate whether a sample comes from the source or the target domain [7, 8, 17]. Finally, the most recent works extend this adversarial logic to the use of GANs [18, 19], for example using two GAN modules with shared weights [9], forcing image to image architectures to have similar activation distributions [20] or simply fooling a GAN's discriminator discerning between domains [21]. These adversarial image generation based methods [18–20] use a class-conditioning or cycle consistency term to learn the discriminative embedding, such that semantically similar images in both domains are projected closeby in the embedding space. Our proposed DeepJDOT uses the concept of a shared embedding for both domains [17] and is built on a similar logic as the MMD-based methods, yet adding a clear discriminative component to the alignment: the proposed DeepJDOT associates representation and discriminative learning, since the optimal transport coupling ensures that distributions are matched, while *i)* the JDOT class loss performs source label propagation to the target samples and *ii)* the fact of learning the coupling in deep layers of the CNN ensures discrimination power.

Optimal transport in domain adaptation. Optimal transport [22–24] has been used in domain adaptation to learn the transformation between domains [4, 25, 26], with associated theoretical guarantees [27]. In those works, the coupling γ is used to transport (i.e. transform) the source data samples through an estimated mapping called barycentric mapping. Then, a new classifier is trained on the transported source data representation. But those different methods can only address problems of small to medium sizes because they rely on the exact solution of the OT problem on all samples. Very recently, Shen *et al.* [28] used the Wasserstein distance as a loss in a deep learning setting to promote similarities between embedded representations using the dual formulation of the problem exposed in [29]. However, none of those approaches considers an adaptation w.r.t. the discriminative content of the representation, as we propose in this paper.

3 Optimal transport for domain adaptation

Our proposal is based on optimal transport. After recalling the associated basic notions and its relation with domain adaptation, we detail the JDOT method [5], which is the starting point of our proposition.

3.1 Optimal Transport

Optimal transport [24] (OT) is a theory that allows to compare probability distributions in a geometrically sound manner. It permits to work on empirical distributions and to exploit the geometry of the data embedding space. Formally, OT searches a probabilistic coupling $\gamma \in \Pi(\mu_1, \mu_2)$ between two distributions μ_1 and μ_2 which yields a minimal displacement cost

$$OT_c(\mu_1, \mu_2) = \inf_{\gamma \in \Pi(\mu_1, \mu_2)} \int_{\mathcal{R}^2} c(\mathbf{x}_1, \mathbf{x}_2) d\gamma(\mathbf{x}_1, \mathbf{x}_2) \tag{1}$$

w.r.t. a given cost function $c(\mathbf{x}_1, \mathbf{x}_2)$ measuring the dissimilarity between samples $\mathbf{x}_1$ and $\mathbf{x}_2$. Here, $\Pi(\mu_1, \mu_2)$ describes the space of joint probability distributions with marginals μ_1 and μ_2. In a discrete setting (both distributions are empirical) this becomes:

$$OT_c(\mu_1, \mu_2) = \min_{\gamma \in \Pi(\mu_1, \mu_2)} < \gamma, \mathbf{C} >_F, \tag{2}$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius dot product, $\mathbf{C} \geq 0$ is a cost matrix $\in \mathbb{R}^{n_1 \times n_2}$ representing the pairwise costs $c(\mathbf{x}_i, \mathbf{x}_j)$, and γ is a matrix of size $n_1 \times n_2$ with prescribed marginals. The minimum of this optimization problem can be used as a distance between distributions, and, whenever the cost c is a norm, it is referred to as the Wasserstein distance. Solving equation (2) is a simple linear programming problem with equality constraints, but scales super-quadratically with the size of the sample. Efficient computational schemes were proposed with entropic regularization [30] and/or stochastic versions using the dual formulation of the problem [31, 29, 32], allowing to tackle small to middle sized problems.

3.2 Joint Distribution Optimal Transport

Courty et al. [5] proposed the joint distribution optimal transport (JDOT) method to prevent the two-steps adaptation (i.e. first adapt the representation and then learn the classifier on the adapted features) by directly learning a classifier embedded in the cost function c. The underlying idea is to align the joint features/labels distribution instead of only considering the features distribution. Consequently, μ_s and μ_t are measures of the product space $\mathcal{X} \times \mathcal{Y}$. The generalized cost associated to this space is expressed as a weighted combination of costs in the feature and label spaces, reading

$$d\left(\mathbf{x}_i^s, \mathbf{y}_i^s; \mathbf{x}_j^t, \mathbf{y}_j^t\right) = \alpha c(\mathbf{x}_i^s, \mathbf{x}_j^t) + \lambda_t L(\mathbf{y}_i^s, \mathbf{y}_j^t) \tag{3}$$

for the i-th source and j-th target element, and where $c(\cdot, \cdot)$ is chosen as a ℓ_2^2 distance and $L(\cdot, \cdot)$ is a classification loss (e.g. hinge or cross-entropy). Parameters α and λ_t are two scalar values weighing the contributions of distance terms. Since target labels $\mathbf{y}_j^t$ are unknown, they are replaced by a surrogate version $f(\mathbf{x}_j^t)$, which depends on a classifier $f : \mathcal{X} \to \mathcal{Y}$. Accounting for the classification loss leads to the following minimization problem:

$$\min_{f, \gamma \in \Pi(\mu_s, \mu_t)} < \gamma, \mathbf{D}_f >_F, \tag{4}$$

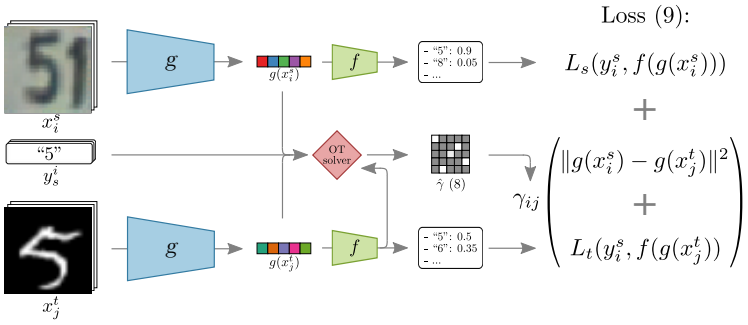

Fig. 1. Overview of the proposed DeepJDOT method. While the structure of the feature extractor g and the classifier f are shared by both domains, they are represented twice to distinguish between the two domains. Both the latent representations and labels are used to compute per batch a coupling matrix γ that is used in the global loss function.

where $\mathbf{D}_f$ depends on f and gathers all the pairwise costs $d(\cdot, \cdot)$. As a by-product of this optimization problem, samples that share a common representation and a common label (through classification) are matched, yielding better discrimination. Interestingly, it is proven in [5] that minimizing this quantity is equivalent to minimizing a learning bound on the domain adaptation problem. However, JDOT has two major drawbacks: *i)* on large datasets, solving for γ becomes intractable because γ scales quadratically in size to the number of samples; *ii)* the cost $c(\mathbf{x}_i^s, \mathbf{x}_j^t)$ is taken in the input space as the squared Euclidean norm on images and can be uninformative of the dissimilarity between two samples. Our proposed DeepJDOT solves those two issues by introducing a stochastic version computing only small couplings along the iterations of a CNN, and by the fact that the optimal transport is learned between the semantic representations in the deeper layers of the CNN, rather than in the image space.

4 Proposed method

4.1 Deep Joint Distribution Optimal Transport(DeepJDOT)

The DeepJDOT model, illustrated in Fig. 1, is composed of two parts: an embedding function $g : \mathbf{x} \to \mathbf{z}$, where the input is mapped into the latent space Z, and the classifier $f : \mathbf{z} \to \mathbf{y}$, which maps the latent space to the label space on the target domain. The latent space can be any feature layer provided by a model, as in our case the penultimate fully connected layer of a CNN. DeepJDOT optimizes jointly this feature space and the classifier to provide a method that performs well on the target domain. The solution to this problem can be achieved by minimizing the following objective function:

$$\min_{\gamma \in \Pi(\mu_s, \mu_t), f, g} \quad \sum_i \sum_j \gamma_{ij} d\left(g(\mathbf{x}_i^s), \mathbf{y}_i^s; g(\mathbf{x}_j^t), f(g(\mathbf{x}_j^t))\right), \tag{5}$$

where $d\left(g(\mathbf{x}_i^s), \mathbf{y}_i^s; g(\mathbf{x}_j^t), f(g(\mathbf{x}_j^t))\right) = \alpha\|g(x_i^s) - g(x_j^t)\|^2 + \lambda_t L\left(y_i^s, f(g(x_j^t))\right)$, and α and λ_t are the parameters controlling the tradeoff between the two terms, as in equation (3). Similarly to JDOT, the first term in the loss compares the compatibility of the embeddings for the source and target domain, while the second term considers the classifier f learned in the target domain and its regularity with respect to the labels available in the source. Despite similarities with the formulation of JDOT [5], our proposition comes with the notable difference that, in DeepJDOT, the Wasserstein distance is minimized between the joint (embedded space/label) distributions within the CNN, rather than between the original input spaces. As the deeper layers of a CNN encode both spatial and semantic information, we believe them to be more apt to describe the image content for both domains, rather than the original features that are affected by a number of factors such as illumination, pose or relative position of objects.

One can note that the formulation reported in equation (5) only depends on the classifier learned in the target domain. By doing so, one puts the emphasis on learning a good classifier for the target domain, and disregards the performance of the classifier when considering source samples. In recent literature, such a degradation in the source domain has been named as '*catastrophic forgetting*' [33, 34]. To avoid such forgetting, one can easily re-incorporate the loss on the source domain in (5), leading to the final DeepJDOT objective:

$$\min_{\gamma, f, g} \frac{1}{n^s} \sum_i L_s\left(y_i^s, f(g(x_i^s))\right) + \sum_{i,j} \gamma_{ij} \left(\alpha\|g(x_i^s) - g(x_j^t)\|^2 + \lambda_t L_t\left(y_i^s, f(g(x_j^t))\right)\right). \tag{6}$$

This last formulation is the optimization problem solved by DeepJDOT. However, for large sample sizes the constraint of computing a full γ yields a computationally infeasible problem, both in terms of memory and time complexity. In the next section, we propose an approximation method based on stochastic optimization.

4.2 Solving the optimization problem with stochastic gradients

In this section, we describe the approximate optimization procedure for solving DeepJDOT. Equation (6) involves two groups of variables to be optimized: the OT matrix γ and the models f and g. This suggest the use of an alternative minimization approach (as proposed in the original JDOT). Indeed, when $\hat{g}$ and $\hat{f}$ are fixed, solving equation (6) boils down to a standard OT problem with associated cost matrix $C_{ij} = \alpha\|\hat{g}(x_i^s) - \hat{g}(x_j^t)\|^2 + \lambda_t L_t\left(y_i^s, \hat{f}(\hat{g}(x_j^t))\right)$. When fixing $\hat{\gamma}$, optimizing g and f is a classical deep learning problem. However, computing the optimal coupling with the classical OT solvers is not scalable to large-scale datasets. Despite some recent development for large scale OT with general ground loss [31, 32], the model does not scale sufficiently to meet requirements of recent computer vision tasks.

Therefore, in this work we propose to solve the problem with a stochastic approximation using minibatches from both the source and target domains [35].

This approach has two major advantages: it is scalable to large datasets and can be easily integrated in modern deep learning frameworks. More specifically, the objective function (6) is approximated by sampling a mini-batch of size m, leading to the following optimization problem:

$$\min_{f,g} \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} L_s\left(y_i^s, f(g(x_i^s))\right) + \min_{\gamma\in\Delta}\sum_{i,j}^{m}\gamma_{ij}\left(\alpha\|g(x_i^s)-g(x_j^t)\|^2 + \lambda_t L_t\left(y_i^s, f(g(x_j^t))\right)\right)\right] \tag{7}$$

where $\mathbb{E}$ is the expected value with respect to the randomly sampled mini-batches drawn from both source and target domains. The classification loss functions for the source (L_s) and target (L_t) domains can be any general class of loss functions that are twice differentiable. We opted for a traditional cross-entropy loss in both cases. Note that, as discussed in [35], the expected value over the minibtaches does not converge to the true OT coupling between every pair of samples, which might then lead to the appearance of connections between samples that would not have been connected in the full coupling. However, this can also be seen as a regularization that will promote sharing of the mass between neighboring samples. Finally note that we did not use the regularized version of OT as in [35], since it introduces an additional regularization parameter that should be cross-validated, which can make the model calibration even more complex. Still, the extension of DeepJDOT to regularized OT is straightforward and could be beneficial for high-dimensional embeddings g.

Consequently, we propose to obtain the stochastic update for Eq.(7) as follows (and summarized in Algorithm 4):

1. With fixed CNN parameters $(\hat{g}, \hat{f})$ and for every randomly drawn minibatch (of m samples), obtain the coupling

$$\min_{\gamma\in\Pi(\mu_s,\mu_t)} \quad \sum_{i,j=1}^{m}\gamma_{ij}\left(\alpha\|\hat{g}(x_i^s)-\hat{g}(x_j^t)\|^2 + \lambda_t L_t\left(y_i^s, \hat{f}(g(x_j^t))\right)\right) \tag{8}$$

   using the network simplex flow algorithm.
2. With fixed coupling $\hat{\gamma}$ obtained at the previous step, update the embedding function (g) and classifier (f) with stochastic gradient update for the following loss on the minibatch:

$$\frac{1}{m}\sum_{i=1}^{m} L_s\left(y_i^s, f(g(x_i^s))\right) + \sum_{i,j=1}^{m}\hat{\gamma}_{ij}\left(\alpha\|g(x_i^s)-g(x_j^t)\|^2 + \lambda_t L_t\left(y_i^s, f(g(x_j^t))\right)\right). \tag{9}$$

   The domain alignment term aligns only the source and target samples with similar activation/labels and the sparse matrix $\hat{\gamma}$ will automatically perform label propagation between source and target samples. The classifier f is simultaneously learnt in both source and target domain.

Algorithm 1 DeepJDOT stochastic optimization

Require: $\mathbf{x}^s$: source domain samples, $\mathbf{x}^t$: target domain samples, $\mathbf{y}^s$: source domain labels

1: **for** each batch of source $(\mathbf{x_b}^s, \mathbf{y_b}^s)$ and target samples $(\mathbf{x_b}^t)$ **do**
2: fix $\hat{g}$ and $\hat{f}$, solve for γ as in equation (8)
3: fix $\hat{\gamma}$, and update for g and f according to equation (9)
4: **end for**

5 Experiments and Results

We evaluate DeepJDOT on three adaptation tasks: digits classification (Section 5.1), the OfficeHome dataset (Section 5.2), and the Visual Domain Adaptation challenge (visDA; Section 5.3). For each dataset, we first present the data, then detail the implementation and finally present and discuss the results.

5.1 Digit classification

Datasets We consider four data sources (domains) from the digits classification field: MNIST [36], USPS [37], MNIST-M, and the Street View House Numbers (SVHN) [38] dataset. Each dataset involves a 10-class classification problem (retrieving numbers 0-9):

- *USPS*. The USPS datasets contains 7‘291 training and 2‘007 test grayscale images of handwritten images, each one of size 16×16 pixels.
- *MNIST*. The MNIST dataset contains 60‘000 training and 10‘000 testing grayscale images of size 28×28.
- *MNIST M*. We generated the MNIST-M images by following the protocol in [8]. MNIST-M is a variation on MNIST, where the (black) background is replaced by random patches extracted from the Berkeley Segmentation Data Set (BSDS500) [39]. The number of training and testing samples are the same as the MNIST dataset discussed above.
- *SVHN*. The SVHN dataset contains house numbers extracted from Google Street View images. We used the *Format2* version of SVHN, where the images are cropped into 32×32 pixels. Multiple digits can appear in a single image, the objective is to detect the digit in the center of the image. This dataset contains 73‘212 training images, and 26‘032 testing images of size $32\times 32\times 3$. The respective examples of the each dataset is shown in Figure 2.

The three following experiments were run (the arrow direction corresponds to the sense of the domain adaptation):

- *USPS*$\leftrightarrow$*MNIST*. The USPS images are zero-padded to reach the same size as MNIST dataset. The adaptation is considered in both directions: USPS $\rightarrow$ MNIST, and MNIST $\rightarrow$ USPS.

Fig. 2. Examples from the MNIST, USPS, SVHN and MNIST-M datasets.

- *SVHN→MNIST*. The single-channel MNIST images are replicated three times to form a gray 3 channels image, and resized to match the resolution of the SVHN images. Here, the adaptation is considered in only one direction: SVHN→MNIST. Adapting SVHN images to MNIST is challenging due to the variations in the SVHN images [8]
- *MNIST→MNIST-M*. MNIST is considered as the source domain and MNIST-M as the target domain. The color MNIST-M images can be easily identified by a human, however it is challenging for the CNN trained on MNIST, which is only grayscale. Again, the gray scale MNIST images are replicated three times to match the color resolution of the MNIST-M images.

Model For all digits adaptation experiments, our embedding function g is trained from scratch with six 3×3 convolutional layers containing 32, 32, 64, 64, 128 and 128 filters, and one fully-connected layer of 128 hidden units followed by a sigmoid nonlinearity respectively. Classifier f then consists of a fully-connected layer, followed by a softmax to provide the class scores. The Adam optimizer ($lr = 2e{-}4$) is used to update our model using mini-batch sizes of $m_S = m_T = 500$ for the two domains (50 samples per class in the source mini-batch). The hyper-parameters of DeepJDOT, $\alpha = 0.001$ and $\lambda_t = 0.0001$, are fixed experimentally.

We compare DeepJDOT with the following methods:

– non-adversarial discrepancy methods: DeepCORAL [6], MMD[14], DRCN[40], DSN [41], AssocDA[16], Self-ensemble[42][4],
– adversarial discrepancy methods: DANN[8], ADDA[21],
– adversarial image generation methods: CoGAN[9], UNIT[18], GenToAdapt[19] and I2I Adapt[20].

To ensure fair comparison, we re-implemented the most relevant competitors (CORAL, MMD, DANN, and ADDA). For the other methods, the results are directly reported from the respective articles.

Results The performance of DeepJDOT on the four digits adaptation tasks is reported in Table 1. The first row (`source only`) shows the accuracies on target

[4] we report a comparison against [42] by using minimal data augmentation (corresponding to MT+CT* in Table 1 of [42]). We do not compare against their full model, as they use a much heavier data augmentation and different networks.

test data achieved with classifiers trained on source data without adaptation, and the row (`target only`) reports accuracies on the target test data achieved with classifiers trained on the target training data. This method is considered as an upper bound for our proposed method and can be seen as our gold standard. `StochJDOT` (stochastic adaptation of JDOT) refers to the accuracy of our proposed method, when the discrepancy between source and target domain is computed with an ℓ_2 distance in the original image space. Lastly, `DeepJDOT-source` indicates the source data accuracy, after adapting to the target domain, and can be considered a measure of catastrophic forgetting.

The experimental results show that DeepJDOT achieves accuracies comparable or higher to the current state-of-the-art methods. When the methods in the first block of Table 1 are considered, `DeepJDOT` outperforms the competitors by large margins, with the exception of `DANN` that have similar performance on the MNIST→USPS task. In the more challenging adaptation settings (SVHN→MNIST and MNIST→MNIST-M), the state-of-the-art methods[5] were not able to adapt well to the target domain. Next, when the methods in the second block of Table 1 is considered, our method showed impressive performance, despite `DeepJDOT` not using any complex procedure for generating target images to perform the adaptation.

***t*-SNE embeddings** We visualize the quality of the embeddings for the source and target domain learnt by `DeepJDOT`, `StochJDOT` and `DANN` using t-SNE embedding on the MNIST→MNIST-M adaptation task (Figure 3). As expected, in the source model the samples from the source domain are well clustered and target samples are more scattered. The t-SNE embeddings with the `DANN` were not able to align the distributions well, and this observation also holds for `StochJDOT`. It is noted that `StochJDOT` does not align the distributions, but learns the classifier in target domain directly. The poor embeddings of the target samples with `StochJDOT` shows the necessity of computing the ground metric (cost function) of optimal transport in the deep CNN layers. Finally, `DeepJDOT` perfectly aligns the source domain samples and target domain samples to each other, which explains the good numerical performances reported above. The "tentacle"-shaped and near-perfect separation of the classes in the embedding illustrate the fact that `DeepJDOT` finds an embedding that both aligns the source/target distribution, but also maximizes the margin between the classes.

Ablation study Table 2 reports the results obtained in the USPS→MNIST and MNIST→ MNIST-M cases for models using only parts of our proposed loss (equation (6)). When only the JDOT loss is considered ($\alpha d + L_t$ case), the accuracy drops in both adaptation cases. This behavior might be due to overfitting of the target classifier to the noisy pseudo- (propagated) labels. However, the performance is comparable to non-adversarial discrepancy-based methods

[5] For `ADDA`[21] in the SVHN→MNIST adaptation task the accuracy is reported from the paper, as we were not able to further improve the source only accuracy

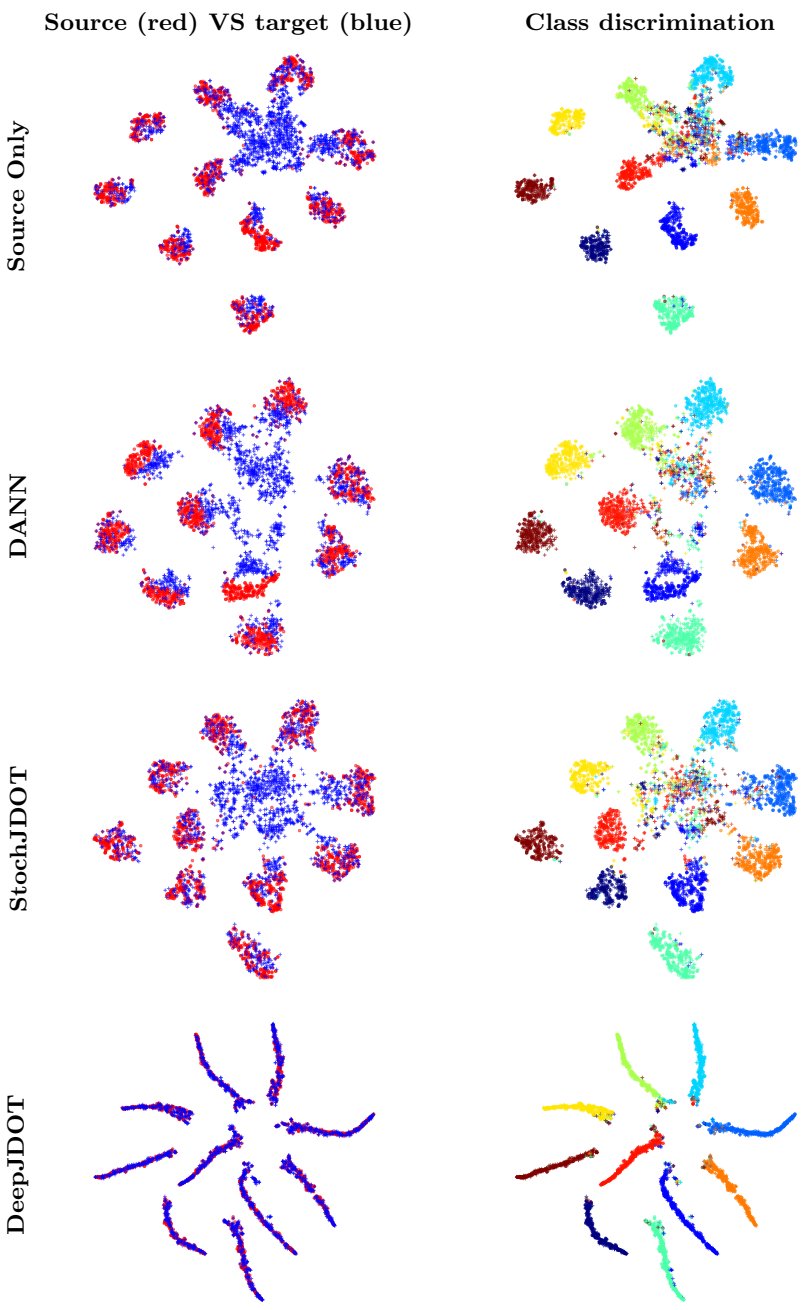

Fig. 3. t-SNE embeddings of 2'000 test samples for MNIST (source) and MNIST-M (target) for `Source only` classifier, `DANN`, `StochJDOT` and `DeepJDOT`. The left column shows domain comparisons, where colors represent the domain. The right column shows the ability of the methods to discriminate classes (samples are colored w.r.t. their classes).

Table 1. Classification accuracy on the target test datasets for the digit classification tasks. *Source only* and *target only* refer to training on the respective datasets without domain adaptation and evaluating on the target test dataset. The accuracies reported in the first block are our own implementations, while the second block reports performances from the respective articles. **Bold** and *italic* indicates the best and second best results. The last line reports the performance of DeepJDOT on the source domain.

Method	Adaptation:source→target			
	MNIST → USPS	USPS → MNIST	SVHN → MNIST	MNIST → MNIST-M
Source only	94.8	59.6	60.7	60.8
DeepCORAL [6]	89.33	91.5	59.6	66.5
MMD [14]	88.5	73.5	64.8	72.5
DANN [8]	*95.7*	90.0	70.8	75.4
ADDA [21]	92.4	93.8	76.0[5]	78.8
AssocDA [16]	-	-	*95.7*	*89.5*
Self-ensemble[4][42]	88.14	92.35	93.33	-
DRCN [40]	91.8	73.6	81.9	-
DSN [41]	91.3	-	82.7	83.2
CoGAN [9]	91.2	89.1	-	-
UNIT [18]	**95.9**	*93.5*	90.5	-
GenToAdapt [19]	95.3	90.8	92.4	-
I2I Adapt [20]	92.1	87.2	80.3	-
StochJDOT	93.6	90.5	67.6	66.7
DeepJDOT (ours)	*95.7*	**96.4**	**96.7**	**92.4**
target only	95.8	98.7	98.7	96.8
DeepJDOT-source	98.5	94.9	75.7	97.8

Table 2. Ablation study of DeepJDOT.

Method	USPS → MNIST	MNIST → MNIST-M
$L_s + (\alpha d + L_t)$	96.4	92.4
$\alpha d + L_t$	86.41	73.6
$L_s + \alpha d$	95.53	82.3

reported in Table 1. On the contrary, when only the feature space distribution is included in Equation (6), i.e. the $L_s + \alpha d$ experiment, the accuracy is close to our full model in USPS→MNIST direction, but drops in the MNIST→ MNIST-M one. Overall the accuracies are improved compared to the original JDOT model, which highlights the importance of including the information from the source domain. Moreover, this also highlights the importance of simultaneously updating the classifier both in the source and target domain. Summarizing, this ablation study showed that the individual components bring complimentary information to achieve the best classification results.

5.2 Office-Home

Dataset The Office-Home dataset [43] contains around $15'500$ images in 65 categories from four different domains: artistic paintings, clipart, product and real-world images.

Table 3. Performance of DeepJDOT on the Office-Home dataset. "Ar" = artistic paintings, "Cl" = clipart, "Pr" = product, "Rw" = real-world images. Performance figures of competitive methods are reported from [43].

Method	Ar→ Cl	Ar→ Pr	Ar→ Rw	Cl→ Ar	Cl→ Pr	Cl→ Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Mean
CORAL[45]	27.10	36.16	44.32	26.08	40.03	40.33	27.77	30.54	50.61	38.48	36.36	57.11	37.91
JDA [46]	25.34	35.98	42.94	24.52	40.19	40.90	25.96	32.72	49.25	35.10	35.35	55.35	36.97
DAN [47]	30.66	42.17	54.13	32.83	47.59	49.58	29.07	34.05	56.70	43.58	38.25	62.73	43.46
DANN [8]	33.33	42.96	54.42	32.26	49.13	49.76	30.44	38.14	56.76	44.71	42.66	64.65	44.94
DAH [43]	31.64	40.75	51.73	34.69	51.93	52.79	29.91	**39.63**	60.71	44.99	45.13	62.54	45.54
DeepJDOT	**39.73**	**50.41**	**62.49**	**39.52**	**54.35**	**53.15**	**36.72**	39.24	**63.55**	**52.29**	**45.43**	**70.45**	**50.67**

Model In this case, we use a pre-trained VGG-16 model [44] with the last layer replaced, but perform no data augmentation. We use $3'250$ samples per domain to compute the optimal couplings. We compared our model with following state-of-the-art methods: CORAL[45], JDA[46], DAN[47], DANN[8], and DAH[43].

Results Table 3 lists the performance of DeepJDOT compared to a series of other adaptation methods. As can be seen, DeepJDOT outperforms all other models by a margin on all tasks, except for the adaptation from domain "product" to "clipart".

5.3 VisDA-2017

Dataset The Visual Domain Adaptation classification challenge of 2017 (VisDA-2017; [48]) requires training a model on renderings of 3D models for each of the 12 classes and adapting to natural images sampled from MS-COCO [49] (validation set) and YouTube BoundingBoxes [50] (test set), respectively. The test set performances reported here were evaluated on the official server.

Model Due to VisDA's strong adaptation complexity, we employ ResNet-50 [51] as a base model, replacing the last layer with two MLPs that map to 512 hidden an then to the 12 classes, respectively. We train a model on the source domain and then freeze it to calculate source feature vectors, adapting an intially identical copy to the target set. We use $4'096$ samples per domain to calculate the couplings. Data augmentation follows the scheme of [42].

Results DeepJDOT's performance on VisDA-2017 is reported in Table 4 along with the baselines (DeepCORAL, DAN) from the evaluation server[6]. Our entry in the evaluation server is mentioned as `oatmil`. We can see that our method achieved better accuracy than the distribution matching methods (DeepCORAL [6], DAN [47]) with all the classes, expect `knife`. We observe a negative transfer for the class `car` for DeepJDOT, however this phenomena is also valid with the most of the current methods (see the evaluation server results). For a fair comparison with the rest of the methods in the evaluation server, we also showed (values in bracket of Table 4) the accuracy difference between the source model

[6] `https://competitions.codalab.org/competitions/17052#results`

Table 4. Performance of DeepJDOT on the VisDA 2017 classification challenge. The scores in the bracket indicate the accuracy difference between the source (unadapted) model and target (adapted) model. The respective values of CORAL and DAN are reported from the evaluation server[6].

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbd	train	truck	Mean
Source only	36.0	4.0	19.9	**94.7**	14.8	0.42	38.7	3.8	37.4	8.1	71.9	6.7	28.0
DeepCORAL [6]	62.5	21.7	66.3	64.6	31.1	36.7	54.2	24.9	73.8	29.9	43.4	34.2	45.3 (19.0)
DAN [47]	55.3	18.4	59.8	68.6	55.3	**41.4**	63.4	30.4	78.8	23.0	62.9	40.2	49.8 (19.5)
DeepJDOT	**85.4**	**50.4**	**77.3**	87.3	**69.1**	14.1	**91.5**	**53.3**	**91.9**	**31.2**	**88.5**	**61.8**	**66.9** (**38.9**)

and target model. Our method is ranked sixth when the mean accuracy is considered, and third when the difference between the source model and target model is considered at the time of publication. It is noted that the performance of our method depends on the capacity of the source model: if a larger CNN is used, the performance of our method could be improved further.

6 Conclusions

In this paper, we proposed the DeepJDOT model for unsupervised deep domain adaptation based on optimal transport. The proposed method aims at learning a common latent space for the source and target distributions, that conveys discriminant information for both domains. This is achieved by minimizing the discrepancy of joint deep feature/labels domain distributions by means of optimal transport. We propose an efficient stochastic algorithm that solves this problem, and despite being simple and easily integrable into modern deep learning frameworks, our method outperformed the state-of-the-art on cross domain digits and office-home adaptation, and provided satisfactory results on the VisDA-2017 adaptation.

Future works will consider the evaluation of this method in multi-domains scenario, as well as more complicated cost functions taking into account similarities of the representations across the embedding layers and/or similarities of labels across different classifiers.

Acknowledgement

This work benefited from the support of Region Bretagne grant and OATMIL ANR-17-CE23-0012 project of the French National Research Agency (ANR). The constructive comments and suggestions of anonymous reviewers are gratefully acknowledged.

References

1. Patel, V.M., Gopalan, R., Li, R., Chellappa, R.: Visual domain adaptation: a survey of recent advances. IEEE SPM **32**(3) (2015) 53–69

2. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV. (2010) 213–226
3. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: An unsupervised approach. In: ICCV. (2011) 999–1006
4. Courty, N., Flamary, R., Tuia, D., Rakotomamonjy, A.: Optimal transport for domain adaptation. IEEE TPAMI **39**(9) (2017) 1853–1865
5. Courty, N., Flamary, R., Habrard, A., Rakotomamonjy, A.: Joint distribution optimal transportation for domain adaptation. In: NIPS. (2017)
6. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: ECCV workshops. (2016) 443–450
7. Luo, Z., Zou, Y., Hoffman, J., Fei-Fei, L.: Label efficient learning of transferable representations across domains and tasks. In: NIPS. (2017)
8. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**(1) (January 2016) 2096–2030
9. Liu, M.Y., Tuzel, O.: Coupled generative adversarial networks. In Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R., eds.: NIPS. (2016) 469–477
10. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: NIPS. (2007) 137–144
11. Jhuo, I.H., Liu, D., Lee, D.T., Chang, S.F.: Robust visual domain adaptation with low-rank reconstruction. In: CVPR. (2012) 2168–2175
12. Hoffman, J., Rodner, E., Donahue, J., Saenko, K., Darrell, T.: Efficient learning of domain-invariant image representations. In: ICLR. (2013)
13. Aljundi, R., Tuytelaars, T.: Lightweight unsupervised domain adaptation by convolutional filter reconstruction. In: ECCV. (2016)
14. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: ICML. (2015) 97–105
15. Long, M., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. In: NIPS. (2016)
16. Haeusser, P., Frerix, T., Mordvintsev, A., Cremers, D.: Associative domain adaptation. In: ICCV. (2017)
17. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: ICCV. (2015)
18. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., eds.: NIPS. (2017) 700–708
19. Sankaranarayanan, S., Balaji, Y., Castillo, C.D., Chellappa, R.: Generate to adapt: Aligning domains using generative adversarial networks. CoRR **abs/1704.01705** (2017)
20. Murez, Z., Kolouri, S., Kriegman, D., Ramamoorthi, R., Kim, K.: Image to Image Translation for Domain Adaptation. ArXiv e-prints (December 2017)
21. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Adversarial discriminative domain adaptation. In: CVPR. (2017)
22. Monge, G.: Mémoire sur la théorie des déblais et des remblais. De l'Imprimerie Royale (1781)
23. Kantorovich, L.: On the translocation of masses. C.R. (Doklady) Acad. Sci. URSS (N.S.) **37** (1942) 199–201
24. Villani, C.: Optimal transport: old and new. Grundlehren der mathematischen Wissenschaften. Springer (2009)
25. Courty, N., Flamary, R., Tuia, D.: Domain adaptation with regularized optimal transport. In: ECML. (2014)

26. Perrot, M., Courty, N., Flamary, R., Habrard, A.: Mapping estimation for discrete optimal transport. In: NIPS. (2016) 4197–4205
27. Redko, I., Habrard, A., Sebban, M.: Theoretical analysis of domain adaptation with optimal transport. In: ECML/PKDD. (2017) 737–753
28. Shen., J., Qu, Y., Zhang, W., Yu, Y.: Wasserstein distance guided representation learning for domain adaptation. In: AAAI. (2018)
29. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: ICML. (2017) 214–223
30. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transportation. In: NIPS. (2013) 2292–2300
31. Genevay, A., Cuturi, M., Peyré, G., Bach, F.: Stochastic optimization for large-scale optimal transport. In: NIPS. (2016) 3432–3440
32. Seguy, V., Damodaran, B, B., Flamary, R., Courty, N., Rolet, A., Blondel, M.: Large-scale optimal transport and mapping estimation. In: ICLR. (2018)
33. Shmelkov, K., Schmid, C., Alahari, K.: Incremental learning of object detectors without catastrophic forgetting. In: ICCV, Venice, Italy (2017)
34. Li, Z., Hoiem, D.: Learning without forgetting. IEEE TPAMI (in press)
35. Genevay, A., Peyré, G., Cuturi, M.: Sinkhorn-autodiff: Tractable wasserstein learning of generative models. arXiv preprint arXiv:1706.00292 (2017)
36. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (Nov 1998) 2278–2324
37. Hull, J.J.: A database for handwritten text recognition research. IEEE TPAMI **16**(5) (May 1994) 550–554
38. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS worksophs. (2011)
39. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE TPAMI **33**(5) (May 2011) 898–916
40. Ghifary, M., Kleijn, W.B., Zhang, M., Balduzzi, D., Li, W.: Deep reconstruction-classification networks for unsupervised domain adaptation. In: ECCV. (2016) 597–613
41. Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., Erhan, D.: Domain separation networks. In: NIPS. (2016) 343–351
42. French, G., Mackiewicz, M., Fisher, M.: Self-ensembling for visual domain adaptation. In: International Conference on Learning Representations. (2018)
43. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: (IEEE) Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
44. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
45. Sun, B., Feng, J., Saenko, K.: Return of frustratingly easy domain adaptation. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. AAAI'16, AAAI Press (2016) 2058–2065
46. Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S.: Transfer feature learning with joint distribution adaptation. In: 2013 IEEE International Conference on Computer Vision. (Dec 2013) 2200–2207
47. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In Bach, F., Blei, D., eds.: Proceedings of the 32nd International Conference on Machine Learning. Volume 37 of Proceedings of Machine Learning Research., Lille, France, PMLR (07–09 Jul 2015) 97–105
48. Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., Saenko, K.: Visda: The visual domain adaptation challenge (2017)

49. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision, Springer (2014) 740–755
50. Real, E., Shlens, J., Mazzocchi, S., Pan, X., Vanhoucke, V.: Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, IEEE (2017) 7464–7473
51. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778