# Feature Selection for Unsupervised Domain Adaptation using Optimal Transport

Léo Gautheron, Ievgen Redko, Carole Lartizien

# Feature Selection for Unsupervised Domain Adaptation using Optimal Transport

Léo Gautheron[1⋆], Ievgen Redko[2], and Carole Lartizien[2]

[1] Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School
Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France
leo.gautheron@univ-st-etienne.fr
[2] Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne,
CNRS, Inserm, CREATIS UMR 5220, U1206, F-69621, LYON, France
{name.surname}@creatis.insa-lyon.fr

**Abstract.** In this paper, we propose a new feature selection method for unsupervised domain adaptation based on the emerging *optimal transportation theory*. We build upon a recent theoretical analysis of optimal transport in domain adaptation and show that it can directly suggest a feature selection procedure leveraging the shift between the domains. Based on this, we propose a novel algorithm that aims to sort features by their similarity across the source and target domains, where the order is obtained by analyzing the coupling matrix representing the solution of the proposed optimal transportation problem. We evaluate our method on a well-known benchmark data set and illustrate its capability of selecting correlated features leading to better classification performances. Furthermore, we show that the proposed algorithm can be used as a pre-processing step for existing domain adaptation techniques ensuring an important speed-up in terms of the computational time while maintaining comparable results. Finally, we validate our algorithm on clinical imaging databases for computer-aided diagnosis task with promising results.

## 1 Introduction

The majority of well-known machine learning algorithms used in real-world applications are built upon the common strategy often known as empirical risk minimization. This strategy suggests that a classifier that minimizes the loss over the observed samples is expected to generalize and thus to perform well on any other sample coming from the same probability distribution. However, this assumption is often violated in practice where a training sample may be different from new unseen data collected afterwards. For instance, one may consider the spam filtering problem. It is quite intuitive to suggest that a given user will be targeted with spam messages depending on its browsing history and that a classifier distinguishing between spam and non spam messages may not be equally efficient for two different users if it does not adapt correctly. In order to tackle this problem, a new learning paradigm called domain adaptation was proposed [4].

⋆ Most of the work in this paper was carried out when L.G. was affiliated with CREATIS.

The main goal of domain adaptation is to provide methodological frameworks and algorithms that allow to reuse a classifier learned in one area, usually called *source domain*, in a different yet similar area usually called *target domain*. According to the domain adaptation theory presented in [4,3], the efficiency of a given adaptation algorithm depends on its capacity to reduce the discrepancy between the probability distributions of the considered source and target samples and on the existence of a good hypothesis (or classifier) that can minimize both source and target error functions. While finding this optimal hypothesis is a very difficult problem, most domain adaptation algorithms concentrate solely on reducing the discrepancy between two domains based on the observed samples. To this end, several papers [23,17,14,25] proposed to solve the domain adaptation problem by addressing it as a feature selection task. Indeed, for the general adaptation scenario, it is reasonable to assume that the shift between source and target domains may be caused by a changing behavior of a subset of features that characterize the data in both domains. In this case, identifying these features can help to reduce the discrepancy between the source and target domains samples and to allow efficient adaptation.

In this paper, we propose a new feature selection algorithm for unsupervised domain adaptation that allows to rank features based on their similarity across the source and target domains. Our key underlying idea is to solve the optimal transportation problem between the marginal distributions of features in the two domains in order to obtain a coupling matrix given by their joint probability distribution. The goal, then, is to use this coupling matrix to identify the most correlated features by analyzing the diagonal of the coupling matrix where higher coupling values indicate strong correlations between the source and target features. We note that contrary to the state-of-the-art methods that proceed by learning a new richer feature representation before identifying the invariant features, our method performs feature selection directly in the input space. This choice leads to more interpretable results and to a better understanding of the adaptation phenomenon as transformed features cannot directly point out to those descriptors that vary between the two domains. Furthermore, the shifted features identified by our method can be eliminated in order to speed-up domain adaptation algorithms whose running time often inherently depends on the dimensionality of the input data. This latter point is quite important as domain adaptation algorithms are often deployed for high-dimensional data arising from computer vision applications. Despite its advantages, our method does not aim to outperform the state-of-the-art classification results obtained by powerful feature transformation domain adaptation methods as most of them use a very rich class of mappings to find a new data representation. To this end, the foremost goal of this paper is to show that the proposed feature selection method is not a competitor of the state-of-the-art algorithms but is a complementary tool that provides important benefits both in terms of computational efficiency and better understanding of data. All the results presented in our paper are given in order to illustrate this rather than its superiority in terms of classification accuracy.

The rest of this paper is organized as follows: in Section 2 we present a short state-of-the-art on feature selection methods in domain adaptation. Section 3 is devoted to the introduction of basic elements related to the optimal transportation theory that are used later. In Section 4, we show how a theoretical analysis of domain adaptation with optimal

transport can be used to derive a new adaptation algorithm based on feature selection. Based on this, we describe the proposed method and the details of its algorithmic implementation. Section 5 presents experimental evaluations of the proposed method on both a benchmark computer vision data set and a clinical imaging database for computer-aided diagnosis task. Section 6 summarizes our paper by outlining its main contributions and giving the possible future perspectives of this work.

## 2 Related works

As classical feature selection methods [10] are not designed to work well under the assumption of distribution's shift, several methods were specifically proposed in the literature for feature selection in the context of domain adaptation. For instance, in [14], the authors search a latent low-dimensional subspace for two domains by jointly preserving the data structure and by selecting a subset of the latent features through a row-sparsity inducing regularization. While being quite effective in terms of classification results, this method, however, has two important drawbacks. First, it does not identify the original features that contribute to efficient adaptation but rather learns their embedding where the distributions' discrepancy is minimized. Second, its optimization procedure makes use of eigenvalue decomposition which has a high computational cost in large-scale applications. Another example of feature selection methods in domain adaptation are [25] and [17]. The contribution of the former paper consists in learning a least squares SVM in order to further remove the features that incur the smallest loss of the classification margin between the classes. The method described in the latter paper proposes to solve an optimization problem with two terms: the first term maximizes the relevance between source features and labels using the Hilbert-Schmidt Independence Criterion while the second term minimizes the shift between the domains using kernel embeddings. Contrary to our algorithm, the above mentioned methods are supervised as they both use annotations in the target domain. Finally, the method that is the most similar to ours is the feature selection algorithm for transfer learning presented in [23]. In this paper, the authors use a parametric maximum mean discrepancy distance in order to find a weight matrix that allows to identify invariant and shifting features in the original space. As we show in Section 4.1, this method and our contribution are closely related from the theoretical point of view, even though our method remains much more computationally attractive.

## 3 Preliminary knowledge

In this section we give a brief overview of the basic elements related to the optimal transportation theory that are used later.

### 3.1 Optimal Transport

The theory of optimal transport has been introduced by Gaspard Monge in the $18^{th}$ century and was recently revisited in [24]. In essence, this theory gives a mathematically founded tool that allows to align arbitrary probability distributions in an optimal way.

In the discrete case, it can be formalized as follows. Let $\hat{\mu}_S = \frac{1}{N_S}\sum_{i=1}^{N_S}\delta_{x_i^S}$ and $\hat{\mu}_T = \frac{1}{N_T}\sum_{i=1}^{N_T}\delta_{x_i^T}$ be two empirical probability measures defined as uniformly weighted sums of Diracs with mass at locations defined on two point sets $\boldsymbol{S} = \{\boldsymbol{x}_i^S \in \mathbb{R}^d\}_{i=1}^{N_S}$ and $\boldsymbol{T} = \{\boldsymbol{x}_i^T \in \mathbb{R}^d\}_{i=1}^{N_T}$ drawn from arbitrary probability distributions $\mu_S$ and $\mu_T$. The Monge-Kantorovich problem consists in finding a probabilistic coupling $\gamma$ defined as a joint probability distribution over $\boldsymbol{S} \times \boldsymbol{T}$ that minimizes the cost of transport w.r.t. a metric $c : \boldsymbol{S} \times \boldsymbol{T} \to \mathbb{R}_+$:

$$\gamma^* = \underset{\gamma \in \Pi(\hat{\mu}_S, \hat{\mu}_T)}{\arg\min} \langle \gamma, C \rangle_F, \tag{1}$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius dot product, $\Pi(\hat{\mu}_S, \hat{\mu}_T) = \{\gamma \in \mathbb{R}_+^{N_S \times N_T} | \gamma \mathbf{1} = \hat{\mu}_S, \gamma^T \mathbf{1} = \hat{\mu}_T\}$ is a set of doubly stochastic matrices and $C$ is a dissimilarity matrix, i.e., for $\boldsymbol{x}_i^S \in \boldsymbol{S}$ and $\boldsymbol{x}_j^T \in \boldsymbol{T}$, we have $C_{ij} = c(\boldsymbol{x}_i^S, \boldsymbol{x}_j^T)$ which defines the energy needed to move a probability mass from $\boldsymbol{x}_i^S$ to $\boldsymbol{x}_j^T$. This problem admits a unique solution $\gamma^*$ and defines a metric on the space of probability measures (called the Wasserstein distance) as follows:

$$W(\hat{\mu}_S, \hat{\mu}_T) = \min_{\gamma \in \Pi(\hat{\mu}_S, \hat{\mu}_T)} \langle \gamma, C \rangle_F. \tag{2}$$

Despite its elegance and simplicity, the formulation of optimal transport given in Equation 1 (abbreviated **OT**) is a Linear Programming problem that does not scale well because of its computational complexity.

### 3.2 Entropy-regularized optimal transport

In order to solve this issue, [6] proposed to add the entropic regularization of $\gamma$ to the Equation 1 leading to the following optimization problem:

$$\gamma^* = \underset{\gamma \in \Pi(\hat{\mu}_S, \hat{\mu}_T)}{\arg\min} \langle \gamma, C \rangle_F - \frac{1}{\lambda} E(\gamma), \tag{3}$$

where $E(\gamma) := -\sum_{ij} \gamma_{ij} \log \gamma_{ij}$. The regularized optimal transport (abbreviated **OT2**) allows the source instances to be transported more or less uniformly to the target instances based on a hyper-parameter $\lambda$ and can be optimized efficiently with the linear time Sinkhorn-Knopp algorithm [12].

### 3.3 Optimal transport and domain adaptation

The use of optimal transport for domain adaptation has been studied for the first time in [5]. In this work, the authors present a new variant of optimal transport (abbreviated **OT3**) based on Equation 3 by adding a class regularization $\ell_{\frac{1}{2},1}$:

$$\gamma^* = \underset{\gamma \in \Pi(\hat{\mu}_S, \hat{\mu}_T)}{\arg\min} \langle \gamma, C \rangle_F - \frac{1}{\lambda} E(\gamma) + \eta \Omega(\gamma), \tag{4}$$

where the $\Omega(\gamma) = \sum_j \sum_{\mathcal{L}} \|\gamma(I_{\mathcal{L}}, j)\|_1^{1/2}$ term prevents the source instances with different labels to be transported to the same target instance. $I_{\mathcal{L}}$ represents the list of sample indexes in $\boldsymbol{S}$ with label $\mathcal{L}$, and $j$ goes through the sample indexes in $\boldsymbol{T}$.

Using the optimal coupling matrix $\gamma^*$ found with Equation 1, 3 or 4, the authors propose to transport the source samples by solving for each of them:

$$\hat{x}_i^S = \arg\min_{x \in \mathbb{R}^d} \sum_j \gamma_{ij}^* c(x, x_j^T). \tag{5}$$

In case of the squared Euclidean distance, the closed form solution of this problem can be written as:

$$\boldsymbol{S_a} = \text{diag}\left((\gamma^* \mathbf{1})^{-1}\right) \gamma^* \boldsymbol{T}. \tag{6}$$

When the marginals $\hat{\mu}_S$ and $\hat{\mu}_T$ are uniform (in practice, this is always the case for us), the Equation 6 is simplified to

$$\boldsymbol{S_a} = N_s \gamma^* \boldsymbol{T}. \tag{7}$$

With this computation, each source instance is represented as the weighted barycenter of the target instances with which it has the highest values in $\gamma^*$.

For a graphical comparison of the **OT**, **OT2** and **OT3** algorithms, we refer the reader to the Supplementary material.

## 4 Proposed approach

In this section we present our main contribution. We start by formally introducing a theoretical result that we use to derive our algorithm.

### 4.1 Theoretical insight

From a theoretical point of view, domain adaptation problem is often formalized as follows: we define a domain as a pair consisting of a distribution $\mu_D$ on $\mathcal{X}$ and a labeling function $f_D : \mathcal{X} \to [0, 1]$. A hypothesis class $H$ is a set of functions so that $\forall h \in H, h : \mathcal{X} \to \{0, 1\}$. Using the proposed notations, the definition of an error function can be given as follows.

**Definition 1.** *Given a convex loss-function l, the probability according to the distribution $\mu_D$ that a hypothesis $h \in H$ disagrees with a labeling function $f_D$ (which can also be a hypothesis) is defined as*

$$\epsilon_D(h, f_D) = \mathbb{E}_{x \sim \mu_D} \left[ l(h(x), f_D(x)) \right].$$

When the source and target error functions are defined w.r.t. $h$ and $f_S$ or $f_T$, we use the shorthand $\epsilon_S(h, f_S) = \epsilon_S(h)$ and $\epsilon_T(h, f_T) = \epsilon_T(h)$.

The use of optimal transport in domain adaptation was first theoretically analyzed in [18]. In this paper, the authors proved that under some mild assumptions imposed on the form of the transport cost function, the source and target error function can be related through the following inequality

$$\epsilon_T(h) \leq \epsilon_S(h) + W(\mu_S, \mu_T) + \lambda, \tag{8}$$

where $\lambda$ is the combined error of the ideal hypothesis $h^*$ that minimizes $\epsilon_S(h) + \epsilon_T(h)$. This result shows that in order to upper bound the error of a classifier in the target domain, one has to minimize the source error function and the discrepancy between the source and target distributions given by the Wasserstein distance.

Below, we use this result as a starting point in order to develop our approach. To this end, we first notice that the source and target domains can be equivalently seen as 2-dimensional product spaces $\mathcal{X}_S \times \mathcal{F}_S$ and $\mathcal{X}_T \times \mathcal{F}_T$, where $\mathcal{X}_S$ (resp. $\mathcal{X}_T$) and $\mathcal{F}_S$ (resp. $\mathcal{F}_T$) denote the source (resp. target) instance and feature spaces. In this case, the probability distributions $\mu_S$ and $\mu_T$ are also product measures supported on $\mathcal{X}_S \times \mathcal{F}_S$ and $\mathcal{X}_T \times \mathcal{F}_T$ and can be written as $\mu_S^X \times \mu_S^f$ and $\mu_T^X \times \mu_T^f$, respectively. Using the results proved in [22] for concentration of measures in product spaces, we can upper bound the Wasserstein distance between $\mu_S$ and $\mu_T$ as follows:

$$W(\mu_S, \mu_T) \leq W(\mu_S^f, \mu_T^f) + \int_{\mathcal{F}_S} W(\mu_S^X | \mu_S^f, \mu_T^X) d\mu_S^f.$$

Note that in this inequality, measures $\mu_S^f$ (resp. $\mu_T^f$) and $\mu_S^X$ (resp. $\mu_T^X$) can be used interchangeably. We can see that the first term in the right-hand side stands for the Wasserstein distance between the measures defined on the feature spaces while the second term is the expectation of the Wasserstein distance between the source instances measure conditionally on the source features measure $\mu_S^X | \mu_S^f$ and the target instance measure $\mu_T^X$. Now, by plugging it into the learning bound proposed in Equation 8, we obtain

$$\epsilon_T(h) \leq \epsilon_S(h) + W(\mu_S^f, \mu_T^f) + \int_{\mathcal{F}_S} W(\mu_S^X | \mu_S^f, \mu_T^X) d\mu_S^f + \lambda.$$

This inequality shows that when one considers probability measures over a product space of instances and features spaces, successful adaptation necessitates the minimization of the discrepancy between features distributions $\hat{\mu}_S^f$, $\hat{\mu}_T^f$ as well as that of the instances distributions $\mu_S^X$, $\mu_T^X$ conditionally on the source features measure $\mu_S^f$. Thus, it naturally leads to a two-stage procedure where the first goal is to reduce the discrepancy between the features sets of the two domains while the second is to apply an appropriate domain adaptation algorithm between their instances described by an optimal set of features obtained at the first stage.

In what follows, we introduce our method based on the idea of finding a coupling that aligns the distributions of features across the source and target domains. As suggested by the obtained bound, the selected features minimizing the $W(\mu_S^f, \mu_T^f)$ can be used then by a domain adaptation algorithm applied to the source and target samples of a reduced dimensionality. We also note that the Wasserstein distance here can be replaced, in practice, by the popular maximum mean discrepancy distance [9] often used in domain adaptation as both of them belong to a larger class of integral probability metrics defined over different functional classes. In this case, the feature selection algorithm proposed in [23][3] also indirectly minimizes the discrepancy between the marginals $\mu_S^f$ and $\mu_T^f$.

[3] Unfortunately, we were unable to use this method as a baseline in our experiments due to the lack of implementation details in their paper and the absence of a publicly available code.

Nevertheless, the computational complexity of the proposed optimization procedure is polynomial thus making its use prohibitive in real-world applications.

### 4.2 Problem setup

Until now the optimal transport was used in order to align empirical measures $\hat{\mu}_S$ and $\hat{\mu}_T$ defined based on the observable samples $\boldsymbol{S} \in \mathbb{R}^{N_S \times d}$ and $\boldsymbol{T} \in \mathbb{R}^{N_T \times d}$. The interpolation step performed using Equation 7 aims at re-weighting the source instances so that their distribution matches the one of the target samples. The geometric interpretation is that, to minimize the divergence between $\mu_S$ and $\mu_T$, we can associate the source samples with the target samples with which they have the highest coupling values.

As mentioned in the previous section, the idea of our method is to go from the sample space to the feature space. To this end, we now consider that $\boldsymbol{S}$ and $\boldsymbol{T}$ are drawn from 2-dimensional product spaces $\mathcal{X}_S \times \mathcal{F}_S$ and $\mathcal{X}_T \times \mathcal{F}_T$, where $\mathcal{X}_S, \mathcal{X}_T \subseteq \mathbb{R}^d$ while $\mathcal{F}_S \subseteq \mathbb{R}^{N_S}$ and $\mathcal{F}_T \subseteq \mathbb{R}^{N_T}$. In this case, we can define two empirical probability measures

$$\hat{\mu}_S^f = \frac{1}{d}\sum_{i=1}^{d} \delta_{f_i^S} \text{ and } \hat{\mu}_T^f = \frac{1}{d}\sum_{i=1}^{d} \delta_{f_i^T}$$

based on the source and target features $\{f^S\}_{i=1}^d \in \mathcal{F}_S$, $\{f^T\}_{i=1}^d \in \mathcal{F}_T$, respectively. Our goal now would be to transport $\hat{\mu}_S^f$ to $\hat{\mu}_T^f$ by solving the entropic regularized optimal transportation problem given as follows:

$$\gamma^{*f} = \underset{\gamma^f \in \Pi(\hat{\mu}_S^f, \hat{\mu}_T^f)}{\arg\min} \langle \gamma^f, C^f \rangle_F - \frac{1}{\lambda} E\left(\gamma^f\right), \tag{9}$$

where $C_{ij}^f = \|f_i^S - f_j^T\|_2^2$.

In what follows, we show that the solution of this problem can lead to a principally different domain adaptation method that is based on feature selection approach rather than on the original instance re-weighting one.

### 4.3 Finding a shared feature representation

At this point one may notice that in order to apply optimal transport between $\hat{\mu}_S^f$ and $\hat{\mu}_T^f$, it is necessary to calculate the cost matrix $C^f$ which is possible only if the numbers of source and target instances are equal. Furthermore, as source and target features are described by supposedly shifted distributions, aligning them directly using any arbitrary sets of instances may not be appropriate due to the differences in the representation spaces that may exist across the two domains. In order to tackle both of these problems, we propose to find a matching between the sample number $i, \forall i = \{1, \ldots, N_S\}$ describing the source features and the sample number $j, \forall j = \{1, \ldots, N_T\}$ describing the target features based on the original optimal transportation problem. More formally, based on the solution $\gamma^*$ of the optimization problem given by Equation 1, we define the optimal subset of target instances $\boldsymbol{T_u}$ as:

$$\boldsymbol{T_u} := \{x_j \in \boldsymbol{T} | j = \arg\max \gamma_{ij}^*, i \in \{1, \ldots, N_S\}\}. \tag{10}$$

**Algorithm 1:** Sample selection in target domain

**Input** : $\boldsymbol{S} \in \mathbb{R}^{N_S \times d}$, $\boldsymbol{T} \in \mathbb{R}^{N_T \times d}$

**Output** : $\boldsymbol{T_u} \in \mathbb{R}^{N_S \times d}$ - optimal subset of target instances

$\mathtt{S} = \mathtt{zscore}(\mathtt{S})$; $\mathtt{T} = \mathtt{zscore}(\mathtt{T})$

$\gamma^* \leftarrow \mathtt{OT}(\mathtt{S}, \mathtt{T})$

$\mathtt{T_u} \leftarrow \{\mathtt{x_j} \in \mathtt{T} | \mathtt{j} = \underset{\mathtt{i}=1,\ldots,\mathtt{N_S}}{\operatorname{argmax}} \gamma^*_{\mathtt{ij}}\}$

**Algorithm 2:** Feature ranking for domain adaptation

**Input** : $\boldsymbol{S} \in \mathbb{R}^{N_S \times d}$, $\boldsymbol{T} \in \mathbb{R}^{N_T \times d}$

**Output** : List $F$ of $d$ most similar features from $\boldsymbol{S}$ and $\boldsymbol{T}$

$\mathtt{T_u} \leftarrow \mathtt{Algorithm1}(\boldsymbol{S}, \boldsymbol{T})$

$\mathtt{S^T} = \mathtt{zscore}(\mathtt{S^T})$; $\mathtt{T_u^T} = \mathtt{zscore}(\mathtt{T_u^T})$

$\gamma^{*\mathtt{f}} = \mathtt{OT2}(\mathtt{S^T}, \mathtt{T_u^T}, \lambda = 1)$

$\mathtt{F}$= $\mathtt{argSortDesc}(\{\{\gamma^{*\mathtt{f}}\}_{\mathtt{ii}} | \mathtt{i} \in [1, \mathtt{d}]\})$

This particular choice of the algorithm **OT** rather than its regularized versions (**OT2** and **OT3**) is explained by the fact that we are interested in a sparse matching between the two sets, i.e., the one limiting the spread of mass[4].

This process, summarized in Algorithm 1[5], is a required preliminary step consisting in finding which examples will be used to describe the features in the source and target domains. The selection stage used to obtain $\boldsymbol{T_u}$ relies on the intrinsic capacity of the coupling matrix to describe the probability of associating each source instance with each target instance based on their similarity.

### 4.4 Feature selection

Now, we let $\boldsymbol{T} := \boldsymbol{T_u}$ meaning that in Equation 9 the target features are described by the set $\boldsymbol{T_u}$ of the sample instances. Note that if $N_S > N_T$, we invert the roles of $\boldsymbol{S}$ and $\boldsymbol{T}$ in Algorithm 1 and instead let $\boldsymbol{S} := \boldsymbol{S_u}$. Furthermore, in a highly imbalanced classification setting, or in the presence of a large number of instances, we advise to first select a subset of source instances by balancing the samples according to their classes before applying Algorithm 1. This selection allows to capture a class information from the source domain without needing labeled samples from the target domain, and thus is still unsupervised w.r.t. the target domain.

We now solve the problem given in Equation 9 and obtain the optimal coupling $\gamma^{*f} \in \mathbb{R}^{d \times d}$. Similar to what we have done at the sample selection step, we analyze the values of the coupling matrix in order to determine the less shifted features across the two domains. The important difference, however, is that we sort the features by analyzing only the diagonal of the coupling matrix. This peculiarity is explained by the fact that the values on the diagonal correspond to the similarities between the same features in the shared source and target representation space. By transporting the features with the **OT2** algorithm, each source feature is transported to its nearest target features. Because of this, if a given feature is shifted across the two domains, then its mass will be uniformly spread on the target features so that its mass on the corresponding target feature will be rather small. Similarly, if a feature is similar between the source and target domains,

[4] The empirical justification of using the **OT** algorithm for sample selection is given in the Supplementary material.

[5] $\mathtt{zscore}(\mathtt{X})$: for each column of $X$, subtract its mean and divide by its standard deviation

then the majority of the mass of this source feature should be found on its corresponding target feature.

Based on this idea, we propose to construct the ordered list of features $F$, where the feature number $i$ in $F$ is the one having the $i^{\text{th}}$ highest coupling value on the diagonal of the coupling matrix, i.e.:

$$F = \arg\operatorname{sort}(\{\{\gamma^{*f}\}_{ii} | i \in \{1, \dots, d\}). \tag{11}$$

By varying the parameter $\lambda$ in **OT2**, we can spread the mass of a source feature more or less uniformly when transporting it to the target features. Even though one may obtain different coupling values for different values of $\lambda$, it does not affect the order of features returned in $F$ allowing us to fix $\lambda = 1$ in all empirical evaluations to avoid hyper-parameter tuning.

The pseudo-code given in Algorithm 2 summarizes our feature selection method. After having obtained the ordered list of features $F$, we can use its $d^* < d$ first features for the classification problem at hand. It is worth noting that the proposed method can be applied as a pre-processing before using any domain adaptation algorithm to discard the features that are completely different across the two domains. On the other hand, it can also be applied in “no adaptation setting" to select the common features between training and test data.

## 5 Experimental evaluations

In this section, we provide an empirical study of the proposed algorithm based on the benchmark computer vision Office/Caltech data set and on clinical imaging database for computer-aided diagnostic task. Note that the optimal transport algorithms **OT** from Algorithm 1, **OT2** from Algorithm 2 and **OT3** are available in the Python POT library[6], making our method straightforward to implement. Nevertheless, we make the Python implementation and the data used in our experiments (except the medical data set) publicly available[7] for the sake of reproducibility.

### 5.1 Experiments on visual domain adaptation data

The main assumption of our method is that not all features are equally useful for adapting a classifier from source domain to the target one. This is especially the case for data sets described by features calculated using the Bag-of-Words (BoW) methods, such as, for instance, the features of the Office [19]/Caltech [8] data set.

*Office/Caltech data set* For this data set, the classification task is to assign an image to a class based on its content. It is composed of 4 domains $A$, $C$, $W$ and $D$ containing 958, 1123, 295 and 157 images, respectively belonging each to one of 10 different classes. These domains form 12 domain adaptation pairs.

[6] `https://github.com/rflamary/POT`
[7] `https://leogautheron.github.io`

In what follows, we use three different types of features: (1) SURF features [2] of size 800 constructed using the BoW method; (2) CaffeNet features [11] that are obtained by feeding the images to a pre-trained neural network based on the prominent AlexNet [13]; (3) GoogleNet features [21] obtained in the way identical to CaffeNet features using GoogleNet network. In order to obtain CaffeNet and GoogleNet features, these two neural networks were first trained on ImageNet, a large data set containing millions of images distributed across 1000 different classes. We removed their classification layer of size 1000 to use the output of the previous layer, giving 4096 features for CaffeNet and 1024 features for GoogleNet. Note that we downloaded the pre-trained networks from the Caffe website [11] before using them to extract the features on our images, and this without doing any fine-tuning or any other modification of the networks apart from removing their last layer.

The experimental protocol used to evaluate the proposed method is based on the one presented in [5]. For each adaptation pair $\boldsymbol{S} \rightarrow \boldsymbol{T}$, we randomly sample 20 images per class (8 if $\boldsymbol{S}$ is $D$). This gives us 200 images (resp. 80) for $\boldsymbol{S}$. All images from $\boldsymbol{T}$ are considered. We then apply Algorithm 2 with $\boldsymbol{S}$ and $\boldsymbol{T}$ to obtain the ordered list of features $F$. For an increasing number of features $d$, we use the $d$ first features of $F$ to, first adapt $\boldsymbol{S}$ to $\boldsymbol{T}$, and then use a 1-nearest neighbor classifier with the source adapted data as training set to compute the classification accuracy on the target data. We repeat this 19 times and report mean accuracies for each pair.

*Classification results* The classification results for CaffeNet features[8] are given in Table 1. From this table, we see that by selecting 512 features having the highest similarity between the source and target domains, we obtain a mean accuracy of 79.2% across the 12 adaptation pairs compared to 74.4% accuracy obtained using all 4096 features. This behaviour is further confirmed by Figure 1 that illustrates the obtained classification results for a number of features varying between 128 and 4096. From it, we also observe that our method outperforms random feature selection while selecting the least similar features gives worse performances in all cases.

The general comparison for CaffeNet features, SURF and GoogleNet features is given in Table 2. As before, we observe an important difference between taking the first most similar and dissimilar features across the two domains and note that better performances are obtained by taking a reduced number of features. Another noticeable point is that the performances of SURF features are far behind CaffeNet features, itself slightly worse than GoogleNet features. Even by taking a small number of $1024/32 = 32$ GoogleNet features, we obtain a mean accuracy of 80.0% which is at least as good as all the other configurations using SURF and CaffeNet features. To summarize, the presented results clearly show that the order of features returned by our method is directly correlated with their adaptation capacities.

We saw in the previous experiment that our method works for different types of features with the best performances obtained using GoogleNet descriptors. However, these performances were achieved without applying any adaptation algorithm. To this end, we present in Figure 2 the impact of using an adaptation algorithm that takes as

[8] Due to the space limitations, we present the same detailed results for GoogleNet and SURF features in the Supplementary material.

| DA pairs | ↘ 512 | ↗ 512 | 4096 |
|---|---|---|---|
| A→C | **74.9±2.0** | 29.8±2.4 | 71.7±3.5 |
| A→D | **78.8±3.5** | 20.4±2.8 | 76.0±3.5 |
| A→W | **77.6±1.9** | 20.2±3.5 | 66.0±4.6 |
| C→A | **83.7±1.8** | 38.7±4.5 | 82.1±2.2 |
| C→D | **76.2±3.6** | 24.1±3.4 | 74.2±4.9 |
| C→W | **75.4±3.5** | 20.3±3.2 | 70.3±5.3 |
| D→A | **75.4±2.1** | 20.8±3.8 | 68.7±2.9 |
| D→C | 65.0±2.6 | 21.5±2.5 | **66.6±1.8** |
| D→W | **92.6±2.0** | 32.8±5.1 | 91.9±1.9 |
| W→A | **81.5±1.2** | 18.8±2.4 | 68.3±3.0 |
| W→C | **72.2±1.1** | 23.4±2.1 | 61.2±2.1 |
| W→D | **96.5±1.5** | 49.7±3.2 | 96.3±1.0 |
| Mean | **79.2±2.2** | 26.7±3.3 | 74.4±3.0 |

Table 1

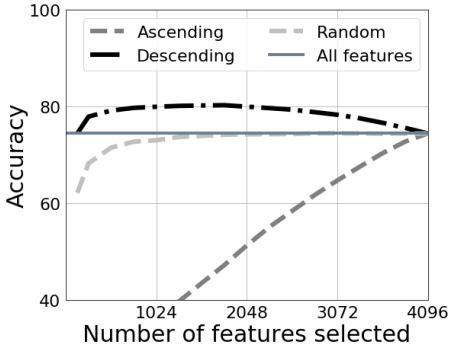


Fig. 1

Results for CaffeNet features. Table 1 presents mean ± standard deviation of recognition accuracy with no adaptation. Here, ↘ $X$ (resp. ↗) indicates the use of the $X$ first features sorted by decreasing (resp. ascending) similarity computed with Algorithm 2. In Figure 1, we plot the mean accuracies from Table 1. Our method corresponds to the 'Descending' curve consisting in selecting the features ordered by decreasing similarity between source and target domains.

| #features | SURF | CaffeNet | GoogleNet |
|---|---|---|---|
| ↘ $d/32$ | 21.3±2.4 | 74.4±2.9 | 80.0±2.6 |
| ↗ $d/32$ | 12.7±2.0 | 20.6±3.0 | 24.2±3.3 |
| ↘ $d/8$ | 25.7±2.6 | 79.2±2.2 | 86.9±1.8 |
| ↗ $d/8$ | 14.0±2.2 | 26.7±3.3 | 48.1±3.9 |
| ↘ $d/2$ | 29.9±2.5 | 80.0±2.2 | 88.1±1.8 |
| ↗ $d/2$ | 16.2±2.5 | 51.3±4.4 | 77.2±2.6 |
| d | 27.9±2.2 | 74.4±3.0 | 86.8±1.8 |

Table 2: Mean accuracies over the 12 DA pairs without applying adaptation using 3 different type of features: SURF (d=800), CaffeNet (d=4096) and GoogleNet (d=1024).

input a reduced set of GoogleNet features returned by our method. Several important conclusions can be made based on these results. First, we notice that our algorithm does not improve the classification results compared to the performance of the **OT3** algorithm with a randomly selected subset of features. As explained in the introduction, **OT3** algorithm finds a new latent projection of source data in order to leverage the shift between the two domains. In this case, eliminating shifted features does not directly contribute to an improved classification performance as **OT3** algorithm can handle the

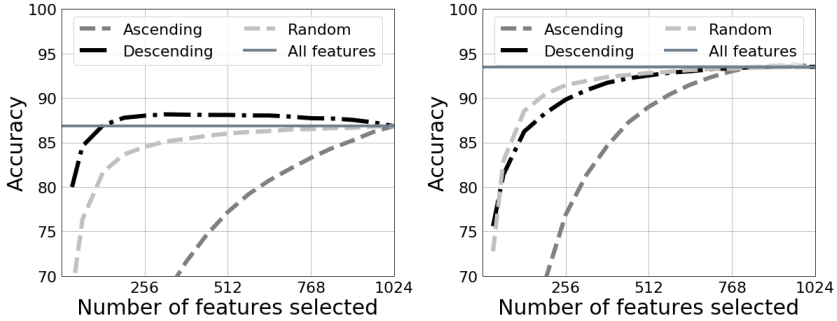


Fig. 2: Mean accuracies over the 12 DA pairs with GoogleNet features using no adaptation (**left**) and the **OT3** adaptation algorithm (**right**). We note that the classification performances are better with all features when we apply an adaptation algorithm: 86.8% without adaptation compared to 93.5% with.

reduction of shift between the two domains pretty well on its own. However, we can also observe the performance of **OT3** algorithm with a reduced "Ascending" set of features reaches its maximal value sooner than when no adaptation is performed. This is explained by the fact that the **OT3** algorithm successfully adapts the most shifted features. It is quite intuitive to assume that by selecting a subset of features, we decrease the computational complexity of the adaptation and classification algorithms that are used later. To support this claim, we present an additional study of the impact of reducing the number of features on both computational time and classification performance for several adaptation algorithms below.

*Running time speed-up* For this experiment, we evaluated the gain in computational time of different adaptation algorithms as a function of the number of features selected by our method. To this end, we compared the “no adaptation" setting with four state-of-the-art adaptation algorithms: **CORAL** [20], **SA** [7], **TCA** [16] and **OT3** [5]. We fixed the subspace dimensions of **SA** and **TCA** to 80 (or to the number of feature selected when smaller than 80) while for **OT3** we set $\lambda = 2$ and $\eta = 1$. Even if from Table 2 we obtained the best performances with GoogleNet features, we select for this experiment the CaffeNet features to better see the computational gain because they have the largest dimensionality (4096).

The results of this evaluation are presented in Table 3. From these results, we see that by selecting 2048 out of 4096 most similar features, we are able to obtain slightly better classification performances for all adaptation methods compared to the case when all features are used. What’s more, the computation time required by the algorithms greatly decrease. When only 512 features are used, an even more impressive speed up is obtained with a very slight drop in performance for the last three methods. These results confirm that our method is capable of finding subsets of similar features between source and target domains that can give comparable and sometimes even improved classification performances while decreasing considerably the computation time required for adaptation methods to converge.

| Method | ↘512 | | ↘1024 | | ↘2048 | | 4096 | |
|---|---|---|---|---|---|---|---|---|
| No adapt. | 79.2±2.2 | 0.00s | 79.9±2.3 | 0.00s | 80.0±2.2 | 0.00s | 74.4±3.0 | 0.00s |
| **CORAL** | 80.5±1.8 | 110.43s | 80.8±1.9 | 587.69s | 80.4±1.7 | 3996.20s | 80.1±1.7 | 29930.39s |
| **SA** | 81.8±2.0 | 13.25s | 82.5±1.8 | 32.09s | 82.9±1.7 | 66.71s | 83.0±1.7 | 169.71s |
| **TCA** | 83.5±2.2 | 221.08s | 85.0±1.9 | 223.62s | 85.8±1.8 | 229.48s | 85.9±1.7 | 242.71s |
| **OT3** | 84.2±2.4 | 19.50s | 86.7±1.9 | 31.76s | 88.8±1.5 | 54.07s | 88.8±1.4 | 97.47s |

Table 3: Mean recognition accuracies in %, standard deviation and sum of total computational time (over the 12 DA pairs and 19 iterations) in seconds for different adaptation algorithms using the CaffeNet features.

| Class | #voxels 1.5T | #voxels 3T |
|---|---|---|
| Non cancer | 363,222 | 846,556 |
| Cancer | 56,126 | 140,840 |
| Total | 419,348 | 987,396 |

Table 4: Repartition of the MRI voxels between the Cancer and Non Cancer classes in source and target domains.

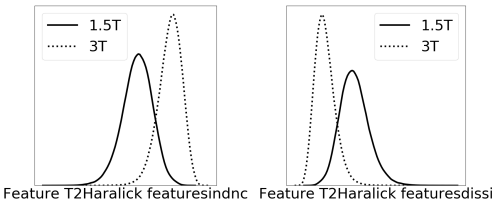


Fig. 3: Example distribution of 2 features illustrating the shift between the source and target domains.

### 5.2 Experiments on medical imaging data set

We now proceed to the evaluation of our method on a clinical data set of multiparametric magnetic resonance images (mp-MRI) collected to train a computer-aided diagnosis system for prostate cancer mapping [15,1]. This system learns a binary decision model in a multidimensional feature space based on training samples (voxels) from different classes of interest. This model is then used to generate cancer probability maps.

*Data description* The considered database consists of 90 mp-MRI exams acquired with different imaging protocols on two different scanners (49 patients on a 1.5T scanner and 41 on a 3T scanner), thus producing heterogeneous data sets. Each individual voxel is described by a binary label (Cancer, Non Cancer) and a set of 95 handcrafted features consisting of image descriptors, texture coefficients, gradients and other visual characteristics (more details in [15]). Some of these 95 features have a clear shift between the two domains, as illustrated in Figure 3. The number of available instances in both domains is shown in Table 4. Our goal is to learn a classifier on annotated 1.5T voxels, representing the source domain, performing well on 3T voxels, considered as the target domain, without using labels from the latter one.

*Evaluation protocol* We first randomly sample a set $\boldsymbol{S}$ of 1500 voxels equiproportionally from the 49 1.5T exams and both classes of interest. Then, we use Algorithm 1 on $\boldsymbol{S}$

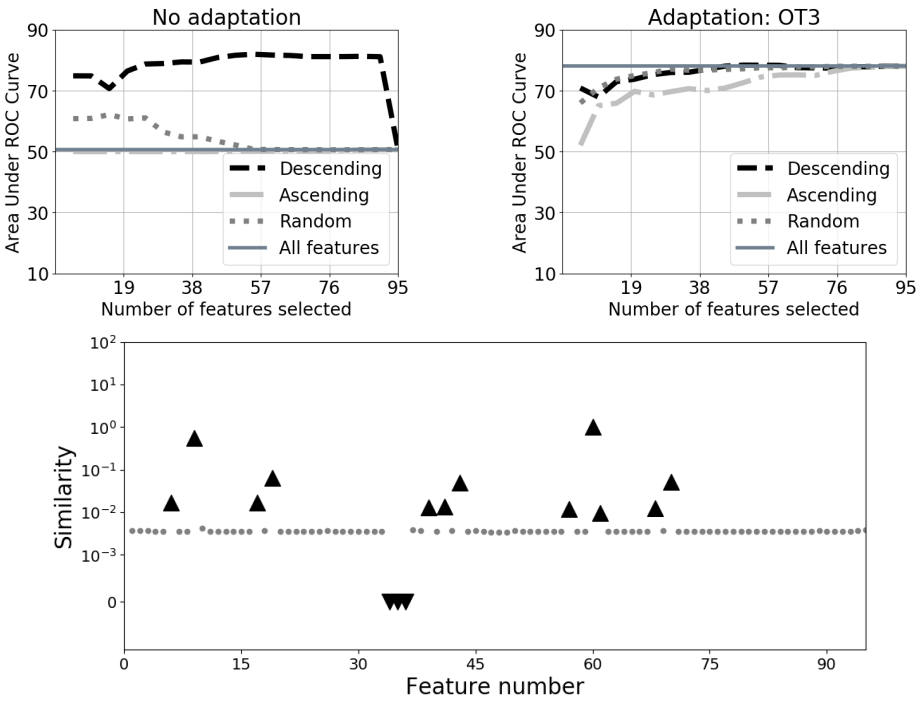


Fig. 4: Performance of our method on the clinical MRI database with no adaptation (top row, left) and using the **OT3** algorithm (top row, right). The log-scaled similarity of features across the two domains estimated by our algorithm is given in the bottom row. We observe that our method correctly identifies the three most shifted features that lead to an important drop in classifier's performance.

and on $\boldsymbol{T}$ as 20000 randomly sampled voxels from the 41 3T exams to obtain $\boldsymbol{T_u}$. This step is followed by the adaptation of $\boldsymbol{S}$ to $\boldsymbol{T_u}$, training a linear SVM on $\boldsymbol{S_a}$ and testing it on all voxels from the 3T target domain.

We used the area under the ROC Curve (AUC) as the diagnostic performance measure. This is due to the fact that both the source and the target domains data exhibit an important class imbalance with $86\%$ of non-cancer voxels. In this case, the classification accuracy used in the previous experiments does not provide a truthful picture of the classifier's performance. Our feature selection method is used as a standalone method and in combination with the **OT3** adaptation algorithm. As before, we repeat this process 20 times, and we report the mean AUC over the 20 iterations.

*Obtained results* The results for this data set are shown in Figure 4. When all the 95 features are used, we obtain an AUC of 50% without adaptation, corresponding to the worst possible performance with no distinction between Cancer and Non cancer classes. By applying our feature selection algorithm (the "Descending" curve) in a standalone manner, we are able to reach an AUC of 80% with a significant drop in

performance when the 3 more dissimilar features are added. On the other hand and similar to the Office/Caltech data set, using our feature selection algorithm before applying an adaptation algorithm reduces greatly the number of features needed to achieve comparable performance. This benefit presents an important computational gain when high-dimensional data sets are considered. Finally, we argued that one of the strengths of our method is its ability to identify the original features causing the shift between the source and target domains. To this end, we plot in Figure 4 the coupling values used to order features by their similarity across the two domains. From this Figure, we can see that our algorithm allows to identify the three most shifted features that lead to a significant performance drop observed previously.

## 6 Conclusions and future perspectives

In this paper, we presented a new feature selection method for domain adaptation based on optimal transport. Building upon a recent theoretical work on optimal transport in domain adaptation, we proposed a feature selection method that transports the empirical distribution of features in the source domain to that of the target one in order to obtain a coupling matrix representing their joint distribution. This coupling matrix is further used to identify the subset of features that remain unshifted across the two domains. We evaluated our method on both benchmark and real-world data sets and showed its efficiency in identifying the subset of features that successfully reduces the discrepancy between the two domains. Furthermore, we illustrated the usefulness of our method in reducing the computational time of several state-of-the-art methods that converge faster when taking as input a reduced set of features returned by our algorithm.

The possible future investigations that may follow up the presented work are many. First of all, we would like to combine our feature selection algorithm with a feature-transformation domain adaptation algorithm in a way such that the projection of data and the selection of features would be performed simultaneously. The potential interest of this joint approach would be to reduce the computational complexity of the adaptation methods and to improve their performance while maintaining the ease of interpretability of the obtained results. On the other hand, it would be also very interesting to extend the proposed framework to the general transfer learning scenario where the source and target domains tasks are not necessarily the same. In this case, the feature selection algorithm would have to take into account the discriminative power of each source feature in the target domain. Solving this problem in an unsupervised setting is a very challenging task that would require an efficient feature expressiveness measure to be introduced. We believe that this future perspective would be of a great interest in many real-world applications, notably the health-care one, where the manual labeling of the produced MRI scans represents an important bottleneck due to its highly time-consuming nature.

## References

1. Aljundi, R., Lehaire, J., Prost-Boucle, F., Rouvière, O., Lartizien, C.: Transfer learning for prostate cancer mapping based on multicentric MR imaging databases. In: Machine Learning Meets Medical Imaging workshop at ICML. pp. 74–82 (2015)

2. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: ECCV (2006)
3. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.: A theory of learning from different domains. Machine Learning **79**, 151–175 (2010)
4. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: NIPS. pp. 137–144 (2007)
5. Courty, N., Flamary, R., Tuia, D.: Domain adaptation with regularized optimal transport. In: ECML/PKDD. pp. 274–289 (2014)
6. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: NIPS. pp. 2292–2300 (2013)
7. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: ICCV. pp. 2960–2967 (2013)
8. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: An unsupervised approach. In: ICCV. pp. 999–1006 (2011)
9. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. Journal of machine learning research **13**, 723–773 (2012)
10. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of machine learning research **3**, 1157–1182 (2003)
11. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: International conference on Multimedia. pp. 675–678 (2014)
12. Knight, P.A.: The sinkhorn–knopp algorithm: convergence and applications. SIAM Journal on Matrix Analysis and Applications **30**(1), 261–275 (2008)
13. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1097–1105 (2012)
14. Li, J., Zhao, J., Lu, K.: Joint feature selection and structure preservation for domain adaptation. In: IJCAI. pp. 1697–1703 (2016)
15. Niaf, E., Rouvière, O., Mège-Lechevallier, F., Bratan, F., Lartizien, C.: Computer-aided diagnosis of prostate cancer in the peripheral zone using multiparametric mri. Physics in medicine and biology **57**(12), 3833–51 (2012)
16. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks **22**(2), 199–210 (2011)
17. Persello, C., Bruzzone, L.: Kernel-based domain-invariant feature selection in hyperspectral images for transfer learning. IEEE Transactions on Geoscience and Remote Sensing **54**(5), 2615–2626 (2016)
18. Redko, I., Habrard, A., Sebban, M.: Theoretical analysis of domain adaptation with optimal transport. In: ECML/PKDD. pp. 737–753 (2016)
19. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV. pp. 213–226 (2010)
20. Sun, B., Feng, J., Saenko, K.: Return of frustratingly easy domain adaptation. In: AAAI. p. 8 (2016)
21. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. pp. 1–9 (2015)
22. Talagrand, M.: Concentration of measure and isoperimetric inequalities in product spaces. Publications Mathématiques de l'I.H.E.S. **81**, 73–205 (1995)
23. Uguroglu, S., Carbonell, J.: Feature selection for transfer learning. In: ECML/PKDD. pp. 430–442 (2011)
24. Villani, C.: Optimal transport: old and new, vol. 338. Springer Science & Business Media (2008)
25. Yin, Z., Wang, Y., Liu, L., Zhang, W., Zhang, J.: Cross-subject eeg feature selection for emotion recognition using transfer recursive feature elimination. Frontiers in neurorobotics **11** (2017)

# Supplementary material: Feature Selection for Unsupervised Domain Adaptation using Optimal Transport

## A Comparison of Optimal Transport based methods

We begin this Supplementary material by comparing in Figure 5 the three optimal transport algorithms introduced and used in the main paper. We see that the basic **OT** associates one target instance to one source instance while with **OT2** each source point's mass is divided and transported to its closest target points. By adding the class regularization **OT3**, we prevent to transport the mass of source instances of different classes to the same target instance.

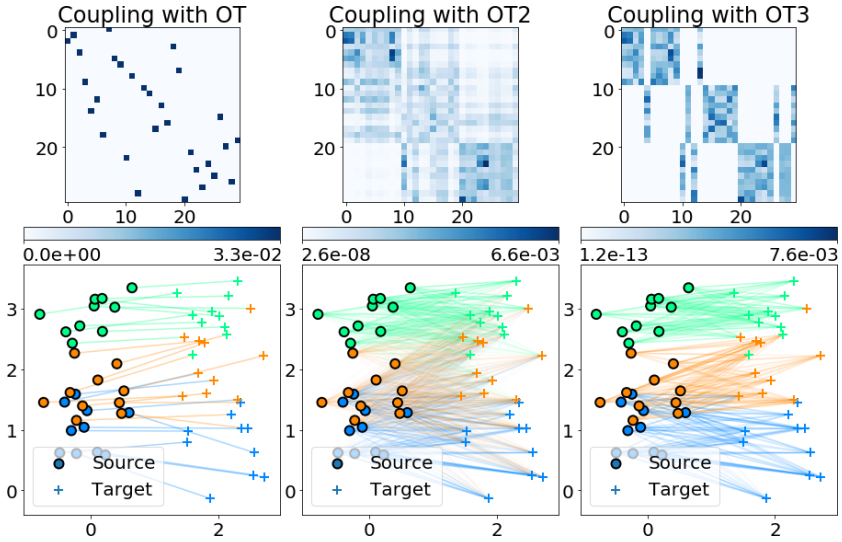


Fig. 5: Comparison of the 3 variants of optimal transport: **OT** on the left, **OT2** in the middle ($\lambda = 1$), **OT3** on the right ($\lambda = 1$, $\eta = 1$). First row shows $\gamma^*$ with highest coupling values seen as darkest blue. Second row shows the source and target points composed of 3 classes in 3 colors. The coupling between them are shown as segments.

# B Additional experiments

We now present a series of additional empirical evaluations that illustrate different important properties of the proposed algorithm. We start with a study of the impact that the algorithm used to find the shared representation for features can have on the obtained performance.

## B.1 Comparison of different samples selection strategies

As explained in the main paper, our method requires to select a set of examples that describe the features in the source and target domains before computing the similarity between them (the features). For our method, we propose to select these examples using the **OT** algorithm between the source and target samples. In Table 5, we evaluate two other sample selection methods on the Office/Caltech data set: first one is based on the random selection of the examples while the second uses a 1-Nearest-Neighbor (1NN) algorithm instead of the **OT** method. The computation of the features' rank is then done in the same way as presented in the main article.

| #features | Random | **OT** | 1NN |
|---|---|---|---|
| ↘128 | 42.7±6.0 | **74.6±3.4** | 72.8±2.9 |
| ↗128 | 43.9±5.6 | 20.8±2.7 | 22.3±3.1 |
| ↘512 | 68.1±4.5 | **79.3±2.6** | 79.1±2.7 |
| ↗512 | 60.8±5.3 | 27.1±3.3 | 27.6±3.4 |
| ↘2048 | 75.9±3.2 | **80.1±2.2** | 79.6±2.7 |
| ↗2048 | 68.9±4.8 | 52.3±4.2 | 50.4±4.4 |
| 4096 | 75.2±3.0 | 75.2±3.0 | 75.2±3.0 |

Table 5: Mean accuracies over the 12 adaptation pairs without applying adaptation on CaffeNet features obtained using different sample selection methods. Our proposed selection method is **OT**.

From this table, we can see that random selection of instances gives poor results for different numbers of features considered in our study. On the other hand, we observe that both **OT** and 1NN algorithm provide close performances in identifying both similar and dissimilar features with a slight superiority of the optimal transport based method. In order to separate the two, we demonstrate in Figure 6 the pitfalls of 1NN based selection that can occur when the vast majority of source points are associated with a handful of target instances.

From this figure, we can see that for the two considered toy data sets, the selection of target instances based on the 1NN algorithm leads to a distribution that does not reflect the true distribution of the target data. If we would have selected points in the target

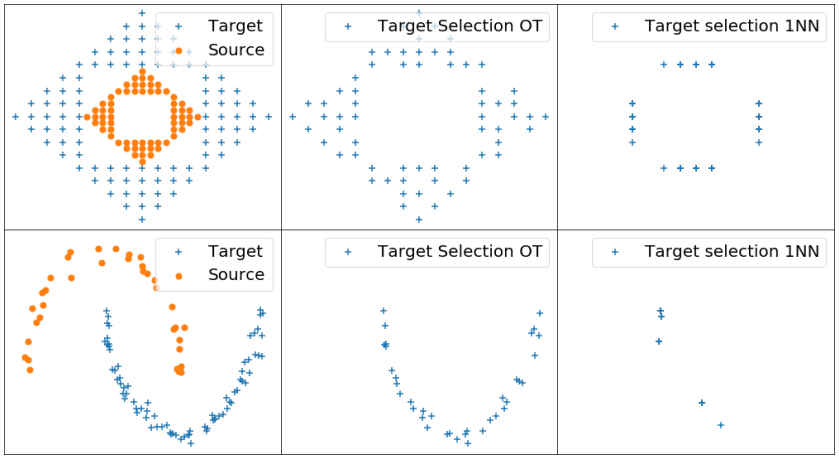


Fig. 6: Two toy examples where we generated a source and a target distribution (left) before using the sample selection procedure in the target domain using the **OT** algorithm (in the middle) and the 1NN selection (on the right).

domain randomly, we would still have the same target distribution, but as we have shown previously in Table 5, the random selection gives worse classification performances. The proposed strategy for the selection of target samples through the **OT** algorithm allows to obtain both good classification performances and to preserve the target data distribution. On the other hand, the computation of the **OT** has a squared space complexity compared to the linear complexity of the 1NN selection. Consequently, we note that the use of the sample selection with the 1NN algorithm can present a good alternative for large-scale machine learning problems.

### B.2 Classification results for SURF and GoogleNet descriptors

In the main paper, we only presented the classification results for the CaffeNet features. To this end, Table 6 and Figure 7 provide the same results for two other types of descriptors considered in our work that are SURF and GoogleNet features.

We observe the same behavior as for the Caffe features where our method gives better or almost identical performances on almost all domain adaptation pairs with significantly less features used. Thus, they confirm our claim about the efficiency of the proposed method for domain adaptation.

| | SURF features | | | GoogleNet features | | |
|---|---|---|---|---|---|---|
| DA pairs | ↘400 | ↗400 | 800 | ↘256 | ↗256 | 1024 |
| A→C | **25.4±2.4** | 15.4±1.5 | 23.1±1.6 | **85.7±1.2** | 64.7±2.4 | 84.6±1.1 |
| A→D | **24.5±2.9** | 16.2±3.0 | 21.9±2.4 | 86.7±2.4 | 68.6±4.9 | **88.4±2.5** |
| A→W | **27.5±2.2** | 16.2±2.6 | 26.0±2.1 | **85.4±3.1** | 51.8±5.9 | 83.5±2.8 |
| C→A | **24.8±1.4** | 14.1±2.2 | 21.2±2.4 | 90.4±1.2 | 74.5±3.4 | **90.6±1.7** |
| C→D | **25.5±3.7** | 15.5±2.8 | 22.8±3.6 | 88.3±2.7 | 68.5±4.3 | **88.6±2.7** |
| C→W | **23.3±3.0** | 13.9±2.2 | 20.6±3.5 | **86.2±2.7** | 54.3±4.6 | 83.3±2.4 |
| D→A | 25.7±2.0 | 15.8±2.9 | **26.7±1.7** | **84.2±2.0** | 46.4±4.2 | 82.3±1.6 |
| D→C | 23.8±1.9 | 16.0±2.1 | **24.8±1.5** | **80.5±1.7** | 46.9±2.8 | 77.8±2.5 |
| D→W | **53.6±3.5** | 22.1±3.4 | 53.3±2.7 | 96.5±1.1 | 81.5±3.4 | **97.4±0.8** |
| W→A | **23.7±1.9** | 15.6±1.9 | 23.1±1.5 | **89.7±0.8** | 55.8±2.5 | 87.0±1.2 |
| W→C | 18.1±1.7 | 12.0±1.6 | **19.5±1.0** | **83.7±1.1** | 49.9±2.5 | 79.4±1.2 |
| W→D | **63.4±3.6** | 21.7±3.4 | 52.4±2.6 | 98.9±0.8 | 93.4±1.8 | **99.2±0.5** |
| Mean | **29.9±2.5** | 16.2±2.5 | 27.9±2.2 | **88.0±1.7** | 63.0±3.5 | 86.8±1.8 |

Table 6: The arrays give the recognition accuracies in % and standard deviation with no adaptation for SURF and GoogleNet features.

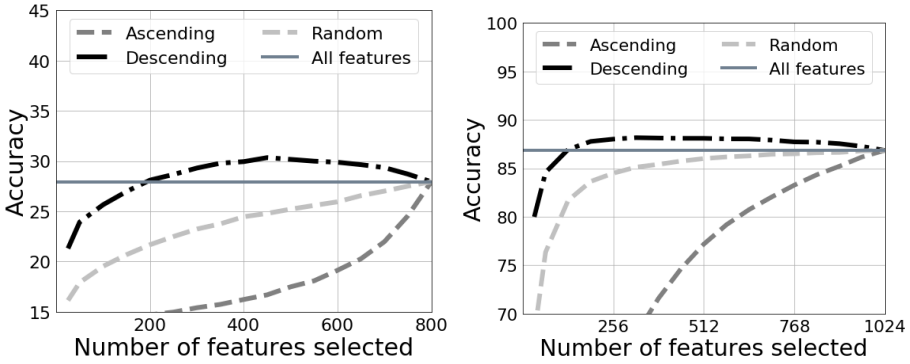


Fig. 7: Mean accuracy results from Table 6 for SURF (**left**) and GoogleNet (**right**) features.