Multi-source Domain Adaptation via Weighted Joint Distributions Optimal Transport

Rosanna Turrisi, Rémi Flamary, Alain Rakotomamonjy, Massimiliano Pontil

Multi-source Domain Adaptation via Weighted Joint Distributions Optimal Transport

Rosanna Turrisi
Istituto Italiano di Tecnologia
Università degli Studi di Ferrara
trrrnn@unife.it

Rémi Flamary
Lagrange, Observatoire de la Côte d'Azur
Université Côte d'Azur
remi.flamary@unice.fr

Alain Rakotomamonjy
Criteo AI Lab
Université de Rouen
alain.rakotomamonjy@univ-rouen.fr

Massimiliano Pontil
Istituto Italiano di Tecnologia
University College London
massimiliano.pontil@iit.it

Abstract

The problem of domain adaptation on an unlabeled *target* dataset using knowledge from multiple labelled *source* datasets is becoming increasingly important. A key challenge is to design an approach that overcomes the covariate and *target* shift both among the sources, and between the *source* and *target* domains. In this paper, we address this problem from a new perspective: instead of looking for a latent representation invariant between *source* and *target* domains, we exploit the diversity of *source* distributions by tuning their weights to the *target* task at hand. Our method, named Weighted Joint Distribution Optimal Transport (WJDOT), aims at finding simultaneously an Optimal Transport-based alignment between the *source* and *target* distributions and a re-weighting of the *sources* distributions. We discuss the theoretical aspects of the method and propose a conceptually simple algorithm. Numerical experiments indicate that the proposed method achieves state-of-the-art performance on simulated and real-life datasets.

1 Introduction

Many machine learning algorithms assume that the test and training datasets are sampled from the same distribution. However, in many real-world applications, new data can exhibit a distribution change (*domain shift*) that degrades the algorithm performance. This shift can be observed for instance on computer vision when changing of background, location, illumination or pose, and in speech recognition for different speakers or recording conditions. To overcome this problem, Domain Adaptation (DA) [1, 2] attempts to leverage labelled data from a *source* domain, in order to learn a classifier for unseen or unlabelled data in a *target* domain. Several DA methods incorporate a distribution discrepancy loss into a neural network to overcome the domain gap. The distance between distributions are usually measured through an adversarial loss [3, 4, 5, 6] or integral probability metrics, such as the maximum mean discrepancy [7, 8]. Recently, DA techniques based on Optimal Transport have been proposed in [9, 10, 11] and justified theoretically in [12].

In this work, we focus on the setting, more common in practice, in which several labelled *sources* are available, denoted in the following as multi-source domain adaptation (MSDA) problem. Many recent approaches motivated by theoretical considerations have been proposed for this problem. For instance, [13, 14] provided theoretical guarantees on how several *source* predictors can be combined using proxy measures, such as the accuracy of an hypothesis. This approach can achieve a low error predictor on the *target* domain, under the assumption that the *target* distribution can be written

Preprint. Under review.

as a convex combination of the *source* distributions. Other recent methods [15, 16, 17] look for a unique hypothesis that minimizes the convex combination of its error on all *source* domains and provide theoretical bounds of the error of this hypothesis on the *target* domain. Those guarantees generally involve some terms depending on the distance between each *source* distribution and the *target* distribution and suggest to find an embedding in which the feature distributions between *sources* and *target* are as close as possible, by using Adversarial Learning [16, 18, 19] or Moment Matching [15]. However, it can be impossible to find an embedding preserving discrimination when the distances between *source*/*target* marginals are small as in Figure 1 where a rotation between the *sources* prevents the existence of such invariant embedding as theorized in [20].

In this paper, we address the MSDA problem following a radically different route. Instead of looking for a latent representation in which all *source* distributions are similar to the *target* one, we embrace the diversity of *source* distributions and look for a convex combination of the joint distribution of *sources* with minimal distance to the *target* one, without referring to a proxy measure such as the accuracy of *source* predictors. After having derived a new generalization bound on the *target* involving that distance, we propose to optimize the Wasserstein distance, defined on the feature/label product space, similar to what was proposed in [10] but between the *target* domain and a weighted sum of the labelled *sources*. A unique feature of our approach is that the weights are learned simultaneously with the classification function, which allows us to distribute the mass based on the similarity of the *sources* with the *target*, both in the feature and in the output spaces. Interestingly our approach estimates weights that provide a measure of domain relatedness and interpretability. We refer to the proposed method as Weighted Joint Distribution Optimal Transport (WJDOT).

The rest of the manuscript is organized as follows. In Section 2, we recall the basics of Optimal Transport (OT) problem and the Joint Distribution Optimal Transport (JDOT). In Section 3, we present a theoretical analysis of multi-source DA and introduce the proposed Weighted Joint Distribution Optimal Transport (WJDOT) method. Finally, in Section 4, we provide experimental results on both synthetic data and real life applications.

Notations We let S be the number of *source* domains, in which both features and labels are available. We suppose that we have access to a differentiable embedding function $g : \mathcal{X} \to \mathcal{G}$, with $\mathcal{G}$ the embedding space. Through the paper all input distributions are in this embedding space. We let p_s be the true distribution in the *source* domain s and p^T the true distribution in the target, both supported on the product space $\mathcal{G} \times \mathcal{Y}$, where $\mathcal{Y}$ is the label space. In practice we only have access to a finite number $\{N_s\}_{s=1}^S$ of samples in the *source* domains leading to the empirical *source* distributions $\hat{p}_s = \frac{1}{N_s} \sum_{i=1}^{N_s} \delta_{g(\mathbf{x}_s^i), \mathbf{y}_s^i}$. In the *target* domain we only have access to a finite number of unlabeled samples in the feature space and to $\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} \delta_{g(\mathbf{x}^i)}$, the empirical *target* marginal distribution. Given a loss function L and a joint distribution p, the expected loss of a function f is defined as $\varepsilon_p(f) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p}[L(\mathbf{y}, f(\mathbf{x}))]$.

2 Optimal Transport and Domain Adaptation

In this section we recall the Optimal Transport problem and the notion of Wasserstein distance, playing a central role in our approach. Then we discuss how they were exploited for Domain adaptation in the Joint Distribution Optimal Transport (JDOT) formulation that will be central in our approach.

Optimal Transport The Optimal transport (OT) problem has been originally introduced by Monge in 1784 [21] and, reformulated as a relaxation by Kantorovich [22]. Let $\hat{\mu}_1 = \sum_i a_1^i \delta_{\mathbf{x}_1^i}$, $\hat{\mu}_2 = \sum_i a_2^i \delta_{\mathbf{x}_2^i}$ be discrete probability measures with $\sum_i a_j^i = 1$ and $a_j^i \geq 0, \forall i, j$. The OT problem searches a transport plan $\gamma \in \Pi(\mu_1, \mu_2) = \{\gamma \geq 0 | \sum_i \gamma_{i,j} = a_2^j, \sum_j \gamma_{i,j} = a_1^i\}$, i.e. the set of joint probabilities with marginals μ_1 and μ_2, that solve the following problem:

$$W_C(\hat{\mu}_1, \hat{\mu}_2) = \min_{\gamma \in \Pi(\mu_1, \mu_2)} \sum_{ij} C_{ij} \cdot \gamma_{ij} \tag{1}$$

where $C_{ij} = c(\mathbf{x}_1^i, \mathbf{x}_2^j)$ represents the cost of transporting mass between $\mathbf{x}_1^i$ and $\mathbf{x}_2^j$ for a given ground cost function $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$. c is often set as the Euclidean distance to recover the classical

W_1 Wasserstein distance. Given a ground cost C, $W_C(\mu_1, \mu_2)$ corresponds to the minimal cost for mapping one distribution to the other and $\gamma^\star$ is the OT matrix describing the relations between *source* and *target* samples. OT and in particular Wasserstein distance have been used with success in numerous machine learning applications such as Generative Adversarial Modeling [23, 24] and Domain Adaptation [9, 10, 25].

Joint Distribution Optimal Transport (JDOT) This method has been proposed in [10] for addressing the problem of unsupervised DA with only one labelled *source* distribution $\hat{p}_1$ and the embedding marginal *target* distribution $\hat{\mu}$. The Kantorovich formulation in Eq. (1) can be expressed by considering the joint distributions instead of the feature marginal ones. However, since no labels are available in the *target* distribution, the authors in [10] proposed to use a proxy joint empirical distribution $\hat{p}^f$ where the labels are replaced by the prediction of a classifier $f : \mathcal{G} \to \mathcal{Y}$, that is,

$$\hat{p}^f = \frac{1}{N} \sum_{i=1}^{N} \delta_{g(\mathbf{x}^i), f(g(\mathbf{x}^i))}. \tag{2}$$

In order to train a meaningful classifier on the *target* domain, the authors proposed to solve the following optimization problem:

$$\min_f \left\{ W_D(\hat{p}_1, \hat{p}^f) = \min_{\pi \in \Pi(\hat{p}_1, \hat{p}^f)} \sum_{ij} D(g(\mathbf{x}_1^i), \mathbf{y}_1^i; g(\mathbf{x}^j), f_2(g(\mathbf{x}^j))) \cdot \pi_{ij} \right\} \tag{3}$$

where the ground cost metric has been designed to measure both embedding and label discrepancy as $D(g(\mathbf{x}_1), \mathbf{y}_1; g(\mathbf{x}_2), f(g(\mathbf{x}_2))) = \beta \|g(\mathbf{x}_1) - g(\mathbf{x}_2)\|^2 + L(\mathbf{y}_1, f(g(\mathbf{x}_2)))$ where L is a loss between classes and β weights the strength of feature loss. JDOT has been supported by generalization error guarantees, see [10] for a discussion. It was later extended to deep learning framework where the embedding g was estimated simultaneously with the classifier f with an efficient stochastic optimization procedure in [11]. One very important aspect of JDOT, that was overlooked by the domain adaptation community is the fact that the optimization problem involves the joint embedding/label distribution. This is in contrast to a large majority of DA approaches [3, 26, 25] using divergences only on the marginal distributions, whereas using simultaneously feature and labels information is the basis of most generalization bounds as discussed in the next section.

3 Multi-source DA with Weighted JDOT (WJDOT)

In this section we present a novel generalization bound for the MSDA problem that depends on a weighting of the *source* distributions. Then, we introduce the WJDOT optimization problem and propose an algorithm to solve it. Finally, we discuss the relation between WJDOT and the state of the art approaches.

3.1 Generalization bound for multi-source DA

The theoretical limits of Domain Adaptation are well studied and well understood since the work of [27] that provided an "impossibility theorem" showing that, if the *target* distribution is too different from the *source* distribution, adaptation is not possible. However in the case of MSDA, one can exploit the diversity of the *source* domains and use only the *sources* close to the *target* distribution, thereby obtaining a better generalization bound. For this purpose, a relevant assumption, already considered in ML [13], is to assume that the *target* distribution is a convex combination of the *source* distributions. The soundness of such an approach is illustrated in the following lemma.

Lemma 1. *For an hypothesis $f \in \mathcal{H}$, denote by $\varepsilon_{p^T}(f)$ and $\varepsilon_{p^{\boldsymbol{\alpha}}}(f)$, the expected loss of f on the target distribution and on the weighted sum of the source distributions, with respect to a loss function L bounded by B. Then we have that*

$$\varepsilon_{p^T}(f) \leq \varepsilon_{p^{\boldsymbol{\alpha}}}(f) + B \cdot D_{TV}\left(p^{\boldsymbol{\alpha}}, p^T\right) \tag{4}$$

where $p^{\boldsymbol{\alpha}} = \sum_{s=1}^{S} \alpha_s p_s$ with $\boldsymbol{\alpha} \in \Delta^S$ is a convex combination of the source distributions, and D_{TV} is the total variation distance.

This simple inequality, whose proof is in the appendix, tells us that the key point for *target* generalization is to have a function f with low error on a combination of the joint *source* distribution and that

combination should be "near" to the *target* distribution. Note that this also holds for single *source* DA problem corroborating the recent findings that just matching marginal distributions may not be sufficient [28]. While the above lemma provides a simple and principled guidance for a multi-source domain adaptation algorithm, it cannot be used for training since it assumes that labels in the *target* domain are known. In the following, we provide generalization bounds in a realistic scenario where no *target* labels are available and a self-labelling strategy is employed to compensate for the missing labels.

Taking inspiration from the result in Lemma 1, we propose a theoretically grounded framework for learning from multiple domain *sources*. Our approach is based on the idea that one can compensate the lack of *target* labels by using an hypothesis labelling function f which provides a joint distribution p^f (2), where f is searched in order to align p^f with a weighed combination of *source* distributions. Following this idea and building upon previous work on single-*source* domain adaptation JDOT [10], we introduce the following generalization bound.

Theorem 1. *Let $\mathcal{H}$ be a space of M-Lipschitz labelling functions. Assume also that the input space is so that $\forall f \in \mathcal{H}, |f(x) - f(x')| \leq M$. Consider the following measure of similarity between $p^{\boldsymbol{\alpha}} = \sum_s \alpha_s p_s$ and p^T introduced in [27, Def. 5]*

$$\Lambda(p^{\boldsymbol{\alpha}}, p^T) = \min_{f \in \mathcal{H}} \varepsilon_{p^{\boldsymbol{\alpha}}}(f) + \varepsilon_{p^T}(f), \tag{5}$$

where the loss function L used in the risk is symmetric and k-Lipschitz and satisfies the triangle inequality. Further, assume that the minimizing function f^* satisfies the Probabilistic Transfer Lipschitzness (PTL) property [10]. Then, for any $f \in \mathcal{H}$, we have

$$\varepsilon_{p^T}(f) \leq W_D\left(p^{\boldsymbol{\alpha}}, p^f\right) + \Lambda(p^{\boldsymbol{\alpha}}, p^T) + kM\phi(\lambda), \tag{6}$$

where $\phi(\lambda)$ is a constant depending on the PTL of $f^\star$.

The PTL property is a reasonable assumption for DA that was introduced in [10] and provides a bound on the probability of finding pair of *source-target* samples of different label within a $1/\lambda$-ball (detailed in supplementary). Note that the quantity $\Lambda(p^{\boldsymbol{\alpha}}, p^T)$ in the bound measures the discrepancy between the true *target* distribution and the "best" combination of the *source* distributions. Minimizing both terms cannot be done when there is no access to labels in the *target* domain but the first term can be minimized *w.r.t.* both f and $\boldsymbol{\alpha}$. The above bound can eventually be refined by introducing sample complexity in the Wasserstein distance as shown in the following theorem.

Theorem 2. *Under the assumptions of Theorem 1, let $\hat{p}_s$ be empirical distributions of N_s samples and $\hat{p}^T$ and empirical distribution with N samples. Then for all $\lambda > 0$, with $\beta = \lambda k$ in the ground metric D we have with probability $1 - \delta$*

$$\varepsilon_{p^T}(f) \leq W_D\left(\hat{p}^{\boldsymbol{\alpha}}, \hat{p}^f\right) + \sqrt{\frac{2}{c'}\log\left(\frac{2}{\delta}\right)}\left(\frac{1}{N} + \sum_s \frac{\alpha_s}{N_s}\right) + \Lambda(p^{\boldsymbol{\alpha}}, p^T) + kM\phi(\lambda). \tag{7}$$

Note that interestingly the $1/N_s$ ratios in the bound are weighted by α_s which means that even if one *source* is poorly sampled it won't have a large impact as soon as the coefficient α_s stays small. The two theorems above indicate that one can minimize the generalization error using a term similar to the JDOT loss, by optimizing both the predictor f and the weights $\boldsymbol{\alpha}$ of the *source* distributions. This is what we propose to do in the following.

3.2 Weighted Joint distribution OT problem

WJDOT optimization problem Our approach aims at finding a function f that aligns the distribution p^f with a convex combination $\sum_{s=1}^S \alpha_s p_s$ of the *source* distributions with convex weights $\boldsymbol{\alpha} \in \Delta^S$ on the simplex. We express the multi-domain adaptation problem as

$$\min_{\boldsymbol{\alpha} \in \Delta^S, f} \quad W_D\left(\hat{p}^f, \sum_{s=1}^S \alpha_s \hat{p}_s\right). \tag{8}$$

Problem above is a minimization of the first term in the bound from Theorem 2 with respect to both f and $\boldsymbol{\alpha}$. The role of the weight $\boldsymbol{\alpha}$ is crucial because it allows in practice to select (when $\boldsymbol{\alpha}$ is sparse) the

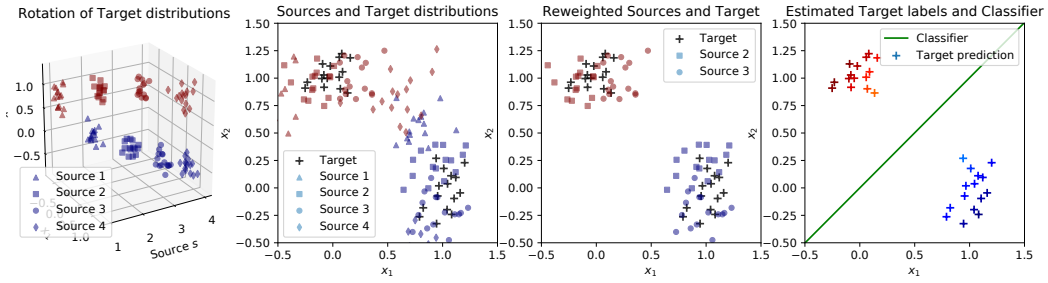

Figure 1: Illustration of WJDOT on 2D simulated data. (left) illustration of 4 *source* distributions p_s corresponding to a rotation increasing with the index. The color of the sample corresponds to the class. (center left) *source* distributions and *target* distribution in black because no class information is available. (center right) weighted sum of *source* distributions using optimal $\boldsymbol{\alpha}^\star = [0, 0.5, 0.5, 0]$ from WJDOT, we can see that only *source* 2 and 3 have a weight > 0 because they are the closest to the *target* distribution in the Wasserstein sense. (right) Final WJDOT classifier and predicted labels for the *target* data.

Algorithm 1 Optimization for WJDOT

Initialise $\boldsymbol{\alpha} = \frac{1}{S}\mathbf{1}_S$ and $\boldsymbol{\theta}$ parameters of $f_{\boldsymbol{\theta}}$ and steps $\mu_{\boldsymbol{\alpha}}$ and $\mu_{\boldsymbol{\theta}}$.
repeat
$\quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \mu_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} W_D \left(p^f, \sum_{s=1}^S \alpha_s p_s\right)$
$\quad \boldsymbol{\alpha} \leftarrow P_{\Delta^S}\left(\boldsymbol{\alpha} - \mu_{\boldsymbol{\alpha}} \nabla_{\boldsymbol{\alpha}} W_D(p^f, \sum_{s=1}^S \alpha_s p_s)\right)$
until Convergence

source distributions that are the closest in the Wasserstein sense and use only those distributions to transfer label knowledge from. An example of the method is provided in Figure 1 showing 4 *source* distributions in 2D obtained from rotation in the 2D space. One interesting property of our approach is that it can adapt to a lot of variability in the *source* distributions as long as the distributions lie in a distribution manifold and this manifold is sampled correctly by the *source* distributions. For instance the linear weights allow to interpolate between *source* distributions and recover the weighted *source* that is the closest to the manifold of distribution hence providing a tightest generalization as shown in the previous section.

Optimization algorithm Problem (8) can be solved with a block coordinate descent similarly to what was proposed in [10]. But with the introduction of the weights $\boldsymbol{\alpha}$ we observed numerically that one can easily get stuck in a local minimum with poor performances. So we proposed the optimization approach in Algorithm 1, that is an alternated projected gradient descent *w.r.t.* the parameters $\boldsymbol{\theta}$ of the classifier $f_{\boldsymbol{\theta}}$ and the weights $\boldsymbol{\alpha}$ of the sources. Note that the sub-gradient of $\nabla_{\boldsymbol{\theta}} W$ is computed by solving the OT problem and using the fixed OT matrix to compute the gradient similarly to [11]. The sub-gradient $\nabla_{\boldsymbol{\alpha}} W$ can be computed in close form from the optimal dual variable of the OT problem. Also note that while we did not need it in the numerical experiments, Algorithm 1 can be performed on mini-batches by sub-sampling the *source* and *target* distribution on very large datasets as suggested in [11] which has been shown recently to provide robust estimators in [29].

Relations with state of the art WJDOT is obviously strongly related to JDOT [10] but opens the door for a more general approach that can adapt to MSDA. There are two simple ways to apply JDOT to multi-source DA. The first one consists in concatenating all the *source* samples into one *source* distribution (equivalent to uniform $\boldsymbol{\alpha}$ if all N_s are equal) and using classical JDOT on the resulting distribution. The second one consists in optimizing a sum of JDOT losses for every *source* distribution but again, this leads to uniform impact of the *sources* on the estimation. It is clear that both approaches are not robust when some *sources* distributions are very different from the *target* (those would have a small weight in WJDOT). There exists a MSDA approach called JCPOT [30] based on [9] that has been proposed to handle only *target* shift (change in proportions between the classes) and satisfies a generalization bound showing that estimating the class proportion in the *target*

distribution is key to recovering good performances. While we did not follow this perspective we claim that WJDOT can also handle the *target* shift as a special case since the reweighting $\boldsymbol{\alpha}$ is directly related to the proportion of classes. The main difference is that JCPOT estimates the proportions of classes using only the feature marginals, whereas WJDOT estimates the proportion and classifier simultaneously by optimizing a Wasserstein distance in the joint embedding/label space. Also note that WJDOT relies on a weighting of the samples where the weight is shared inside the *source* domains. This is a similar approach to Domain Adaptation approaches such as Importance Weighted Empirical Risk Minimization (IWERM) [31] designed for Covariate Shift that use a reweighing of all the samples. One major difference is that we only estimate a relatively small number of weights in $\boldsymbol{\alpha}$ leading to a better posed statistical estimation. It is indeed well known that estimation of continuous density which is necessary for a proper individual reweighting of the samples is a very difficult problem in high dimension.

Finally, as discussed in the introduction, the majority of recent DA approaches based on deep learning [3, 26, 25] relies on the estimation of an embedding that is invariant to the domain which means that the final classifier is shared across all domains when the embedding g is estimated. Those approaches have been extended to multiple *sources* [16, 18, 15] with the objective that the embedded distributions between *sources* and *target* are similar. Our approach differs greatly here for several reasons. First we do not try to cancel the variability across *sources* but to embrace it by allowing the approach to find the *source* domains closest in term of terms of embedding and classifier automatically. There exist numerous examples of *source* variability in real life (such as rotation between the full distributions) that cannot be handled with a global embedding and to the best of our knowledge WJDOT is one of the few generic frameworks that can handle this problem.

4 Numerical experiments

In this section, we first provide some implementation details for WJDOT. We then evaluate the proposed method and compare it with state-of-the-art MSDA methods, on both simulated and real data. For research reproducibility, all the Python/Pytorch [32] code will be released upon publication.

Practical implementation of WJDOT We used in all numerical experiments the WJDOT solver from Algorithm 1. We recall that we suppose in the paper that we have access to a meaningful (as in discriminant) embedding g. This is a realistic scenario due to the wide availability of pre-trained models and advent of reproducible research. Nevertheless we discuss here how to estimate such an embedding when none is available. To keep the variability of the *sources* that is used by WJDOT we propose to estimate an g with the Multi-Task Learning framework originally proposed in [33], i.e.

$$\min_{g,\{f_s\}_{s=1}^S} \quad \sum_{s=1}^S \frac{1}{N_s} \sum_{i=1}^{N_s} \mathcal{L}(f_s \circ g(\mathbf{x}_s^i), \mathbf{y}_s^i). \tag{9}$$

This approach for estimating an embedding g makes sense because it promotes a g that is discriminant for all tasks but allows a variability thanks to the task specific final classifiers f_s which is an assumption at the core of WJDOT. We refer to `WJDOT` where the embedding g is learned with the above procedure as $\texttt{WJDOT}_{mtl}$. Note that this is a two step procedure.

Another important question, especially when performing unsupervised domain adaptation, is the question of how to perform validation of the parameters and early stopping. In unsupervised DA this is always a difficult question due to the lack of *target* samples for validation. To overcome the problem, we use the sum of the squared errors (SSE) between the estimated outputs $f(X)$ and their estimated cluster centroids on the *target* data. We also explored another strategy, based on the classifier accuracy on the sources, that is discussed and reported in the supplementary material.

Compared methods We compare our approach with the following MSDA methods among which two non obvious extension of the JDOT formulation. `CJDOT` consists in concatenating all the *source* samples into one *source* distribution. `MJDOT` consists in optimizing the sum $\sum_s W(p_s, p^f)$ of JDOT objective for all sources. For both JDOT variants, we employ the SSE criterion discussed above to validate both parameters and early stopping. Importance Weighted Empirical Risk Minimization (`IWERM`) [31] that is a variant of ERM where the samples are weighted by the ratio of the *target* and *source* densities minimizing the sum of the `IWERM` objective for each sources. `DCTN` is the Deep cocktail network of [18] where adversarial learning is employed to learn a feature extractor, domain

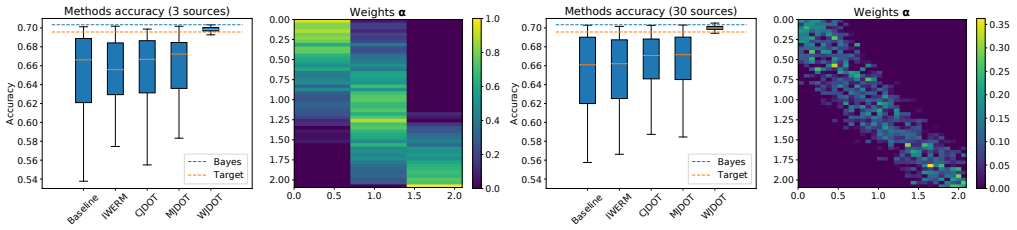

Figure 2: Simulated dataset. Methods' accuracy and recovered $\boldsymbol{\alpha}$ weights for an increasing rotation angle: (left) $S = 3$ and (right) $S = 30$ sources.

discriminators and *source* classifiers. The domain discriminator provides multiple source-target-specific perplexity scores that are used to weight the source-specific classifier predictions and produce the *target* estimation. Finally `M`^{3}`SDA` is the Moment matching approach proposed for MSDA in [15], in which an embedding is learned by aligning moments of the *source* and *target* distributions. Please note that both in `DCTN` and `M`^{3}`SDA` the embedding learning is the core of the methods and hence they are not feasible for a fixed embedding g. For this reason, we compare with these methods only when the g has to be estimated. We also provide performances for `Baseline` that trains a classifier that maximizes performance among all *source* domains. This approach measure the ability to train a unique classifier that is robust to the domain and perform well on target. Finally, we also compare to two unrealistic approaches that use labels in target: `Baseline+Target` is similar to `Baseline` but also use labels in the *target* domain. `Target` trains a classifier using only *target* labels and is more prone to overfitting since less samples are available. Since we have access to labels for the two last approaches, we validate the model by using the classification accuracy on the *target* validation set.All methods are compared on the same dataset split in training (70%), validation (20%) and testing (10%) but the validation set is used only for `Baseline+Target` and `Target`.

Simulated data We consider a classification problem similar to what is illustrated in Figure 1, but with 3 classes, i.e. $\mathcal{Y} = \{0, 1, 2\}$, and in 3D. For the *sources* and *target* we generate N_s and N samples from $S + 1$ Gaussian distributions rotated of angle $\theta_s \in [0, \frac{3}{2}\pi]$ around the x-axis. As the data is already linearly separated, we set g as the identity function in this experiment. We carried out many experiments in order to see the effect of different parameters such as the number of *source* domains S, of *source* samples N_s and of *target* samples N. Each experiment has been repeated 50 times.

We report in Fig. 2 the accuracy of all methods with $N_s = N = 300$ for $S = 3$ (left) and $S = 30$ (right). All competing methods are clearly outperformed by `WJDOT` both in term of performance and variance even for a limited number of sources. Interestingly WJDOT can even outperform `Target` due to its access to a larger number of samples. Another important aspect of WJDOT is the obtained weights $\boldsymbol{\alpha}$ that can be used for interpretation. We show in Fig. 2 that the estimated weights tend to be sparse and put more mass on *sources* that have a similar angle *i.e.* we recover automatically the closest *sources* in the joint distribution manifold. Note that we only report the method's performances on those two configurations the results for other experiments can be found in the supplementary material.

Object recognition The Caltech-Office dataset [34, 35, 36, 9] contains four different domains: Amazon , Caltech [37], Webcam and DSLR. The variability of the different domains come from several factors: presence/absence of background, lightning conditions, noise, etc. We use for the embedding function g the output of the 7th layer of a pre-trained DeCAF model [38], similarly to what was done in [9], resulting into an embedding space $\mathcal{G} \in \mathbb{R}^{4096}$. For f, we employ a one-layer neural network. Training is performed with Adam optimizer with 0.9 momentum and $\epsilon = e^{-8}$. Learning rate and $\ell 2$ regularization on the parameters are validated for all methods.In JDOT extensions and WJDOT, we also validate the β parameter weighting the feature distance in the cost of Eq. (3).

The performance of the different methods are reported in Table 1. We can see that WJDOT is state of the art providing the best Average Rank (AR). Note that the DeCAF pre-trained embedding was originally designed in part to minimize the divergence across domains which as discussed is not the best configuration for WJDOT but it still performs very well showing the robustness of WJDOT to the embedding. Moreover, we observed that for each adaptation problem WJDOT provides one-hot vector

Method	Amazon	dslr	webcam	Caltech10	AR
Baseline	93.13 ± 0.07	94.12 ± 0.00	89.33 ± 1.63	82.65 ± 1.84	4.0
IWERM [31]	93.30 ± 0.75	$\mathbf{100.00 \pm 0.00}$	89.33 ± 1.16	$\mathbf{91.19 \pm 2.57}$	2.25
CJDOT [10]	93.71 ± 1.57	93.53 ± 4.59	$\mathbf{90.33 \pm 2.13}$	85.84 ± 1.73	2.75
MJDOT [10]	94.12 ± 1.57	97.65 ± 2.88	90.27 ± 2.48	84.72 ± 1.73	2.50
WJDOT	$\mathbf{94.23 \pm 0.90}$	$\mathbf{100.00 \pm 0.00}$	89.33 ± 2.91	85.93 ± 2.07	**1.75**
Target	95.77 ± 0.31	88.35 ± 2.76	99.87 ± 0.65	89.75 ± 0.85	-
Baseline+Target	94.78 ± 0.48	99.88 ± 0.82	100.00 ± 0.00	91.89 ± 0.69	-

Table 1: Accuracy of all methods on the Caltech Office Dataset. The average rank of the method across target domains is reported in the last column.

Method	F16	Buccaneer2	Factory2	Destroyerengine	AR
Baseline	69.67 ± 8.78	57.33 ± 7.57	83.33 ± 9.13	87.33 ± 6.72	7.25
IWERM [31]	72.22 ± 3.93	58.33 ± 5.89	85.00 ± 6.23	81.64 ± 3.33	6.75
IWERM_{mtl} [31]	75.00 ± 0.00	66.67 ± 0.00	$\mathbf{100.00 \pm 0.00}$	98.33 ± 3.33	2.75
DCTN [18]	66.67 ± 3.61	68.75 ± 3.61	87.50 ± 12.5	94.44 ± 7.86	5.00
M^3SDA [15]	70.00 ± 4.08	61.67 ± 4.08	85.00 ± 11.05	83.33 ± 0.00	6.50
CJDOT [10]	59.50 ± 13.95	50.00 ± 0.00	83.33 ± 0.00	91.67 ± 0.00	7.75
CJDOT_{mtl} [10]	83.83 ± 5.11	$\mathbf{74.83 \pm 1.17}$	$\mathbf{100.00 \pm 0.00}$	95.74 ± 16.92	2.25
MJDOT[10]	66.33 ± 9.57	50.00 ± 0.00	83.33 ± 0.00	91.67 ± 0.00	7.50
MJDOT_{mtl}[10]	86.00 ± 4.55	72.83 ± 5.73	97.67 ± 3.74	97.74 ± 8.28	2.50
WJDOT	83.33 ± 0.00	58.33 ± 6.01	87.00 ± 6.05	89.00 ± 4.84	5.25
WJDOT_{mtl}	$\mathbf{87.17 \pm 4.15}$	$\mathbf{74.83 \pm 1.20}$	99.67 ± 1.63	$\mathbf{99.67 \pm 1.63}$	**1.25**
Target	73.67 ± 6.09	69.17 ± 7.50	77.33 ± 4.73	73.17 ± 9.90	-
Baseline+Target	71.06 ± 9.31	67.62 ± 11.92	85.33 ± 11.85	79.53 ± 10.05	-

Table 2: Accuracy of all methods on the Music-Speech Dataset. The average rank of the method across target domains is reported in the last column.

$\boldsymbol{\alpha}$ (provided in supplementary) suggesting that only one *source* is needed for the *target* adaptation. Interestingly the source selected by WJDOT is for each target is the one that was reported with the best performance for Single source DA in [9] which shows that WJDOT can automatically find the relevant sources with no supervision.

Music-speech discrimination We now consider the music-speech discrimination task introduced in [39], which includes 64 music and speech tracks of 30 seconds each. We generated 14 noisy datasets by combining the raw tracks with different types of noises from a noise dataset[1]. The noisy datasets have been synthesised by PyDub python library [40]. We then used the libROSA python library [41] to extract 13 MFCCs, computed every 10ms from 25ms Hamming windows followed by a z-normalization per track. We chose each of the four noisy datasets (F16, Bucaneer2, Factory2, Destroyerengine) as *target* domains, considering the remaining noisy datasets and the clean dataset as labelled *source* domains. The feature extraction g is a Bidirectional Long Short-Term Memory (BLSTM) recurrent network with 2 hidden layers containing each 50 memory blocks. The f classifier is learned as one feed-forward layer. Model and training details are reported in the supplementary materials.

We report in Table 2, the mean and standard deviation accuracy on the testing set of each *target* dataset over 50 trials, as well as the Average Rank for each method. First note that on this hard adaptation problem the Baseline+Target approach only slightly improves the Baseline, and most of the methods performance shows large variance. As expected, WJDOT_{mtl} significantly outperforms WJDOT confirming the importance of estimating an embedding g exploiting the *source* variability. WJDOT_{mtl} achieves a 1.25 Average Rank outperforming all the other MSDA methods and also presents low standard deviation, showing robustness to small sample size. Surprisingly, WJDOT_{mtl} even outperforms both the Target and Baseline+Target methods, where the labels are available.

[1] Available at http://spib.linse.ufsc.br/noise.html

5 Conclusion

We presented a novel approach for multi-source DA that relies on OT for propagating labels from the *sources* and a weighting of the *source* domains so as to be able to select the best *sources* for the *target* task at hand in order to get a better prediction. We provided results that show that the proposed approach is theoretically grounded. Finally we presented numerical experiments that illustrate the good performance of the method on both simulated and real-world benchmark datasets. Future works will investigate a regularization of $\boldsymbol{\alpha}$ and estimating simultaneously the embedding g with WJDOT instead of pre-training it with multitask learning. The embedding could indeed be updated for each new *target* which suggests an incremental formulation for WJDOT that could be valuable in practice.

Broader Impact

This work investigates the problem of domain adaptation with multiple *sources* by modeling the variability of the *sources* to better predict on a *target* domain. It could be used to get better specialized AI in several applications such as personal assistants, voice recognition of even bio-metric security. One end application that provided the initial motivation for this study was being able to adapt voice recognition software to speech impaired people and it is still a planned application. As all AI approaches it can be used to put some people at disadvantage and may have consequences. However since the paper in mainly methodological, this will mainly depend on the application.

Finally our approach was not designed to handle bias in the data and since one can specialize even more the method to individuals, there is a risk that a systematic bias in the *source* domains could lead to more important bias in the final decisions.

Acknowledgments and Disclosure of Funding

This work was partially funded through the projects OATMIL ANR-17-CE23-0012 and 3IA Cote d'Azur Investments ANR-19-P3IA-0002 of the French National Research Agency (ANR) as well as a grant from SAP SE and 5x1000, assigned to the University of Ferrara - tax return 2017.

References

[1] James J. Jiang, "A literature survey on domain adaptation of statistical classifiers," 2007.

[2] Wouter M. Kouw and Marco Loog, "A review of single-source unsupervised domain adaptation," *CoRR*, vol. abs/1901.05335, 2019.

[3] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[4] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," *CoRR*, vol. abs/1607.03516, 2016.

[5] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko, "Simultaneous deep transfer across domains and tasks," *CoRR*, vol. abs/1510.02192, 2015.

[6] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, "Adversarial discriminative domain adaptation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[7] Mingsheng Long, Jianmin Wang, and Michael I. Jordan, "Unsupervised domain adaptation with residual transfer networks," *CoRR*, vol. abs/1602.04433, 2016.

[8] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell, "Deep domain confusion: Maximizing for domain invariance," *CoRR*, vol. abs/1412.3474, 2014.

[9] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy, "Optimal transport for domain adaptation," *CoRR*, vol. abs/1507.00504, 2015.

[10] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., pp. 3730–3739. Curran Associates, Inc., 2017.

[11] Bharath Bhushan Damodaran, Benjamin Kellenberger, Remi Flamary, Devis Tuia, and Nicolas Courty, "DeepJDOT: Deep Joint Distribution Optimal Transport for Unsupervised Domain Adaptation," in *ECCV 2018 - 15th European Conference on Computer Vision*, Munich, Germany, Sept. 2018, vol. 11208 of *LNCS*, pp. 467–483, Springer, European Conference on Computer Vision 2018 (ECCV-2018).

[12] Ievgen Redko, Amaury Habrard, and Marc Sebban, "Theoretical analysis of domain adaptation with optimal transport," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part II*, Michelangelo Ceci, Jaakko Hollmén, Ljupco Todorovski, Celine Vens, and Saso Dzeroski, Eds. 2017, vol. 10535 of *Lecture Notes in Computer Science*, pp. 737–753, Springer.

[13] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh, "Domain adaptation with multiple sources," in *Advances in neural information processing systems*, 2009, pp. 1041–1048.

[14] Judy Hoffman, Mehryar Mohri, and Ningshan Zhang, "Algorithms and theory for multiple-source adaptation," in *Advances in Neural Information Processing Systems*, 2018, pp. 8246–8256.

[15] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang, "Moment matching for multi-source domain adaptation," *arXiv preprint arXiv:1812.01754*, 2018.

[16] Han Zhao, Shanghang Zhang, Guanhang Wu, José M. F. Moura, Joao P Costeira, and Geoffrey J Gordon, "Adversarial multiple source domain adaptation," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., pp. 8559–8570. Curran Associates, Inc., 2018.

[17] Junfeng Wen, Russell Greiner, and Dale Schuurmans, "Domain aggregation networks for multi-source domain adaptation," *ArXiv*, vol. abs/1909.05352, 2019.

[18] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin, "Deep cocktail network: Multi-source unsupervised domain adaptation with category shift," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[19] Chuang Lin, Sicheng Zhao, Lei Meng, and Tat-Seng Chua, "Multi-source domain adaptation for visual sentiment classification," *ArXiv*, vol. abs/2001.03886, 2020.

[20] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon, "On learning invariant representations for domain adaptation," in *Proceedings of the 36th International Conference on Machine Learning*, Kamalika Chaudhuri and Ruslan Salakhutdinov, Eds., Long Beach, California, USA, 09–15 Jun 2019, vol. 97 of *Proceedings of Machine Learning Research*, pp. 7523–7532, PMLR.

[21] Gaspard Monge, "Mémoire sur la théorie des déblais et de remblais," in *Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année*. 1781.

[22] L. V. Kantorovich, "On the translocation of masses," in *Journal of Mathematical Sciences*, 2006.

[23] Martin Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[24] Aude Genevay, Gabriel Peyré, and Marco Cuturi, "Learning generative models with sinkhorn divergences," *arXiv preprint arXiv:1706.00292*, 2017.

[25] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[26] Baochen Sun and Kate Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European conference on computer vision*. Springer, 2016, pp. 443–450.

[27] Shai Ben-David, Tyler Lu, Teresa Luu, and Dávid Pál, "Impossibility theorems for domain adaptation," in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 129–136.

[28] Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton, "Domain adaptation with asymmetrically-relaxed distribution alignment," in *Proceedings of the 36th International Conference on Machine Learning*, Kamalika Chaudhuri and Ruslan Salakhutdinov, Eds., Long Beach, California, USA, 09–15 Jun 2019, vol. 97 of *Proceedings of Machine Learning Research*, pp. 6872–6881, PMLR.

[29] Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval, and Nicolas Courty, "Learning with minibatch wasserstein: asymptotic and gradient properties," *arXiv preprint arXiv:1910.04091*, 2019.

[30] Ievgen Redko, Nicolas Courty, Rémi Flamary, and Devis Tuia, "Optimal transport for multi-source domain adaptation under target shift," in *International Conference on Artificial Intelligence and Statistics (AISTAT)*, 2019.

[31] Massahi Sugiyama, Matthias Krauledat, and Klaus-Robert M˜uller, "Covariate shift adaptation my importance weighted cross validation," *J. Mach. Learn. Res.*, vol. 8, pp. 985–1005, Dec. 2007.

[32] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," 2017.

[33] Rich Caruana, "Multitask Learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[34] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell, "Adapting Visual Category Models to New Domains," in *Computer Vision – ECCV 2010*, Kostas Daniilidis, Petros Maragos, and Nikos Paragios, Eds., Berlin, Heidelberg, 2010, pp. 213–226, Springer Berlin Heidelberg.

[35] R Gopalan, Ruonan Li, and Rama Chellapa, "Domain adaptation for object recognition: An unsupervised approach," in *2011 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 999–1006, IEEE.

[36] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman, "Geodesic flow kernel for unsupervised domain adaptation.," in *CVPR*. 2012, pp. 2066–2073, IEEE Computer Society.

[37] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," Tech. Rep. 7694, California Institute of Technology, 2007.

[38] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*, 2014, pp. 647–655.

[39] George Tzanetakis and Perry Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, 2002.

[40] James Robert, Marc Webbie, et al., "Pydub," 2018.

[41] Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and Music Signal Analysis in Python," in *Proceedings of the 14th Python in Science Conference*, Kathryn Huff and James Bergstra, Eds., 2015, pp. 18 – 24.

[42] François Bolley, Arnaud Guillin, and Cédric Villani, "Quantitative concentration inequalities for empirical measures on non-compact spaces," *Probability Theory and Related Fields*, vol. 137, no. 3-4, pp. 541–593, 2007.

A Proofs

A.1 Proof of Lemma 1

Lemma 1. *For an hypothesis $f \in \mathcal{H}$, denote as $\varepsilon_{p^T}(f)$ and $\varepsilon_{p^{\boldsymbol{\alpha}}}(f)$, the expected loss of f on the target and on the weighted sum of the source domains, with respect to a loss function L bounded by B. We have*

$$\varepsilon_{p^T}(f) \leq \varepsilon_{p^{\boldsymbol{\alpha}}}(f) + B d_{TV}\left(p^{\boldsymbol{\alpha}}, p^T\right). \tag{10}$$

where $p^{\boldsymbol{\alpha}} = \sum_{s=1}^{S} \alpha_s p_s$ with $\boldsymbol{\alpha} \in \Delta^S$ is a convex combination of the source distributions, and d_{TV} is the total variation distance.

Proof. We define the error of an hypothesis f with respect to a loss function $L(\cdot,\cdot)$ and a joint probability distribution $p(x,y)$ as

$$\varepsilon_p = \int p(x,y)L(y,f(x))dxdy$$

then using simple arguments, we have

$$\begin{aligned}
\varepsilon_{p^T}(f) &= \varepsilon_{p^T}(f) + \varepsilon_{p^{\boldsymbol{\alpha}}}(f) - \varepsilon_{p^{\boldsymbol{\alpha}}}(f) \\
&\leq \varepsilon_{p^{\boldsymbol{\alpha}}}(f) + |\varepsilon_{p^T}(f) - \varepsilon_{p^{\boldsymbol{\alpha}}}(f)| \\
&\leq \varepsilon_{p^{\boldsymbol{\alpha}}}(f) + \int |p^{\boldsymbol{\alpha}}(x,y) - p^T(x,y)||L(y,f(x)|dxdy \\
&\leq \varepsilon_{p^{\boldsymbol{\alpha}}}(f) + B\int \big|p^{\boldsymbol{\alpha}}(x,y) - p^T(x,y)\big|dxdy
\end{aligned} \tag{11}$$

and using the definition of the total variation distance between distribution concludes the proof. □

A.2 Proof of Theorem 1

The proof of this theorem follows the same steps as the one proposed by Courty et al. [10] and we reproduce it here for a sake of completeness.

Definition 1. ***Probabilistic Transfer Lipschitzness*** *Let p_s and p^T be respectively the source and target distributions. Let $\phi : \mathbb{R} \to [0,1]$. A labeling function $f : \Omega \to \mathbb{R}$ and a joint distribution $\Pi(p_s, p^T)$ over p_s and p^T are Φ-Lipschitz transferable if for all $\lambda > 0$, we have*

$$Pr_{(x_1,x_2)\sim\Pi(p_s,p^T)}\big[|f(x_1) - f(x_2)|] > \lambda D(x_1,x_2)\big] \leq \Phi(\lambda)$$

with d being a metric on Ω

As stated in Courty et al. [10], given function f and coupling Π, this property and definition gives a bound on the probability of finding couple (source-target) of examples that are differently labeled in a $(1/\lambda)$ ball with respect to Π and the metric D .

Theorem 1. *Let $\mathcal{H}$ be a space of M-Lipschitz labelling functions. Assume also that the input space is so that $\forall f \in \mathcal{H}, |f(x) - f(x')| \leq M$. Consider the following measure of similarity between $p^{\boldsymbol{\alpha}} = \sum_s \alpha_s p_s$ and p^T introduced in [27, Def. 5],*

$$\Lambda(p^{\boldsymbol{\alpha}}, p^T) = \min_{f\in\mathcal{H}} \varepsilon_{p^{\boldsymbol{\alpha}}}(f) + \varepsilon_{p^T}(f) \tag{12}$$

where the loss function L used in the risk is symmetric and k-Lipschitz and satisfies the triangle inequality. Further, assume that the minimizing function $f^\star$ satisfies the Probabilistic Transfer Lipschitzness (PTL) property [10]. Then, for any $f \in \mathcal{H}$, we have

$$\varepsilon_{p^T}(f) \leq W_D\left(p^{\boldsymbol{\alpha}}, p^f\right) + \Lambda(p^{\boldsymbol{\alpha}}, p^T) + kM\phi(\lambda). \tag{13}$$

where $\phi(\lambda)$ is a constant depending on the PTL of $f^\star$.

Proof.

$$\begin{aligned}
\varepsilon_{p^T}(f) &= \mathbb{E}_{(x,y)\sim p^T} L(y,f(x)) \\
&\leq \mathbb{E}_{(x,y)\sim p^T}[L(y,f^\star(x)) + L(f^\star(x),f(x))] \\
&= \varepsilon_{p^T}(f^\star) + \mathbb{E}_{(x,y)\sim p^T} L(f^\star(x),f(x)] \\
&= \varepsilon_{p^T}(f^\star) + \mathbb{E}_{(x,y)\sim p^f} L(f^\star(x),f(x)] \\
&= \varepsilon_{p^T}(f^\star) + \varepsilon_{p^f}(f^\star) + \varepsilon_{p^{\boldsymbol{\alpha}}}(f^\star) - \varepsilon_{p^{\boldsymbol{\alpha}}}(f^\star) \\
&\leq |\varepsilon_{p^f}(f^\star) - \varepsilon_{p^{\boldsymbol{\alpha}}}(f^\star)| + \varepsilon_{p^{\boldsymbol{\alpha}}}(f^\star) + \varepsilon_{p^T}(f^\star)
\end{aligned}$$

where the second equality comes from the symmetry of the loss function and the third one is due to the fact that f is a one-to-one mapping and thus $\mathbb{E}_{(x,y)\sim p^T} L(f^\star(x),f(x)] = \mathbb{E}_{(x,y)\sim p^f} L(f^\star(x),f(x)]$.https://www.overleaf.com/project/5e171532c504fe000157d8ba

Now, if we analyze the first term we have $\varepsilon_{p^f}(f^\star) - \varepsilon_{p^{\boldsymbol{\alpha}}}(f^\star)|$

$$
\begin{aligned}
&= \left| \int_{\Omega \times C} L(y, f^\star(x)) |p^T(x,y) - p^{\boldsymbol{\alpha}}(x,y) dx dy \right| \\
&= \left| \int_{\Omega \times C} L(y, f^\star(x)) d(p^T - p^{\boldsymbol{\alpha}}) \right| \\
&\leq \int_{(\Omega \times C)^2} |L(y_t^f, f(x_t)) - L(y_{\boldsymbol{\alpha}}, f^\star(x_{\boldsymbol{\alpha}})| d\Pi^\star((x_{\boldsymbol{\alpha}}, y_{\boldsymbol{\alpha}}), (x_t, y_t^f)) \qquad (14) \\
&= \int_{(\Omega \times C)^2} \Big| L(y_t^f, f(x_t)) - L(y_t^f, f^\star(x_{\boldsymbol{\alpha}}) \\
&\quad + L(y_t^f, f^\star(x_{\boldsymbol{\alpha}}) - L(y_{\boldsymbol{\alpha}}, f^\star(x_{\boldsymbol{\alpha}}) \Big| d\Pi^\star((x_{\boldsymbol{\alpha}}, y_{\boldsymbol{\alpha}}), (x_t, y_t^f)) \\
&\leq \int_{(\Omega \times C)^2} \Bigg[\Big| L(y_t^f, f(x_t)) - L(y_t^f, f^\star(x_{\boldsymbol{\alpha}}) \Big| \\
&\quad + \Big| L(y_t^f, f^\star(x_{\boldsymbol{\alpha}}) - L(y_{\boldsymbol{\alpha}}, f^\star(x_{\boldsymbol{\alpha}}) \Big| \Bigg] d\Pi^\star((x_{\boldsymbol{\alpha}}, y_{\boldsymbol{\alpha}}), (x_t, y_t^f)) \\
&\leq \int_{(\Omega \times C)^2} \left[k \big| f^\star(x_t) - f^\star(x_{\boldsymbol{\alpha}}) \big| + \Big| L(y_t^f, f^\star(x_{\boldsymbol{\alpha}}) - L(y_{\boldsymbol{\alpha}}, f^\star(x_{\boldsymbol{\alpha}}) \Big| \right] d\Pi((x_{\boldsymbol{\alpha}}, y_{\boldsymbol{\alpha}}), (x_t, y_t^f)) \qquad (15) \\
&\leq kM\phi(\lambda) + \int_{(\Omega \times C)^2} \left[k\lambda D(x_t, x_{\boldsymbol{\alpha}}) + \Big| L(y_t^f, f^\star(x_{\boldsymbol{\alpha}}) - L(y_{\boldsymbol{\alpha}}, f^\star(x_{\boldsymbol{\alpha}}) \Big| \right] d\Pi((x_{\boldsymbol{\alpha}}, y_{\boldsymbol{\alpha}}), (x_t, y_t^f)) \qquad (16) \\
&\leq kM\phi(\lambda) + \int_{(\Omega \times C)^2} \left[\beta D(x_t, x_{\boldsymbol{\alpha}}) + L(y_t^f, y_{\boldsymbol{\alpha}}) \Big| \right] d\Pi((x_{\boldsymbol{\alpha}}, y_{\boldsymbol{\alpha}}), (x_t, y_t^f)) \qquad (17) \\
&= kM\phi(\lambda) + W_1(p^{\boldsymbol{\alpha}}, p^f) \qquad (18)
\end{aligned}
$$

Inequality in line (14) is due to the Kantorovitch-Rubinstein theorem stating that for any coupling $\Pi \in \Pi(p^{\boldsymbol{\alpha}}, p^T)$, the following inequality holds:

$$
\left| \int_{\Omega \times C} L(y, f^\star(x)) d(p^T - p^{\boldsymbol{\alpha}}) \right| \leq \left| \int_{(\Omega \times C)^2} |L(y_t^f, f(x_t)) - L(y_{\boldsymbol{\alpha}}, f^\star(x_{\boldsymbol{\alpha}})| d\Pi((x_{\boldsymbol{\alpha}}, y_{\boldsymbol{\alpha}}), (x_t, y_t^f)) \right|
$$

followed by an application of the triangle inequality. Since, the above inequality applies for any coupling, it applies also for $\Pi^\star$. Inequality (15) is due to the assumption that the loss function is k-Lipschitz in its second argument. Inequality (16) that $f^\star$ and $\Pi^\star$ verify the probabilistic Lipschitzness property with probability $1 - \phi(\lambda)$. In addition, taking into account that the difference between 2 samples with respect to $f^\star$ is bounded by M, we have the term $kM\phi(\lambda$ that covers the regions where PTL assumption does not hold. Inequality (17) is obtained from the symmetry of $D(\cdot, \cdot)$, the triangle inequality on the loss and by posing $k\lambda = \beta$ □

A.3 Proof of Theorem 2

First we need to prove the following Lemma

Lemma 2. *For any distributions $\hat{p}_s, p_s$ and $\boldsymbol{\alpha} \in \Delta^S$ in the simplex we have*

$$
W_D \left(\sum_s \alpha_s \hat{p}_s, \sum_s \alpha_s p_s \right) \leq \sum_s \alpha_s W_D \left(\hat{p}_s, p_s \right)
$$

Proof. First we recall that the Wasserstein Distance between two distribution is

$$
W_D(p^1, p^2) = \min_{\pi \in \Pi(p^1, p^2)} \int D(\mathbf{v}, \mathbf{v}') \pi(\mathbf{v}, \mathbf{v}') d\mathbf{v} d\mathbf{v}' \qquad (19)
$$

where $\Pi(p^1, p^2) = \{\pi | \int \pi(\mathbf{v}, \mathbf{v}')d\mathbf{v}' = p^1(\mathbf{v}), \int \pi(\mathbf{v}, \mathbf{v}')d\mathbf{v} = p^2(\mathbf{v}')\}$ are the marginal. constraints.

We can now prove Theorem 2 that is recalled hein the following.

Theorem 2. *Under the assumptions of Theorem 1, let $\hat{p}_s$ be empirical distributions of N_s samples with $s = 1, \cdots, S$ and $\hat{p}^T$ and empirical distribution with N samples. Then for all $\lambda > 0$, with $\beta = \lambda k$ we have with probability $1 - \delta$*

$$\varepsilon_{p^T}(f) \leq W_D\left(\hat{p}^{\boldsymbol{\alpha}}, \hat{p}^f\right) + \sqrt{\frac{2}{c'}\log\left(\frac{2}{\delta}\right)}\left(\frac{1}{N} + \sum_s \frac{\alpha_s}{N_s}\right) + \Lambda(p^{\boldsymbol{\alpha}}, p^T) + 2kM\phi(\lambda) \tag{20}$$

Proof. Let π_s be the OT matrix solution of $W_D(\hat{p}_s, p_s)$ satisfying the marginal constraints $\pi_s \in \Pi(\hat{p}_s, p_s)$. It is clear because of the linearity of the marginal constraints that $\sum_s \alpha_s \pi_s \in \Pi(\sum_s \alpha_s \hat{p}_s, \sum_s \alpha_s p_s)$ which means that $\sum_s \alpha_s \pi_s$ is a feasible point of the optimization problem of $W_D\left(\sum_s \alpha_s \hat{p}_s, \sum_s \alpha_s p_s\right)$. Since the objective function is also linear it means that the OT loss for $\sum_s \pi_s$ is equal to $\sum_s \alpha_s W_D\left(\hat{p}_s, p_s\right)$ and will be greater or equal to $W_D\left(\sum_s \alpha_s \hat{p}_s, \sum_s \alpha_s p_s\right)$. □

In order to prove Theorem 2 first we show that

$$\begin{aligned} W_D\left(\sum_s \alpha_s p_s, p^f\right) &\leq W_D\left(\sum_s \alpha_s \hat{p}_s, \hat{p}^f\right) + W_D(\hat{p}^f, p^f) + W_D\left(\sum_s \alpha_s \hat{p}_s, \sum_s \alpha_s p_s\right) \\ &\leq W_D\left(\sum_s \alpha_s \hat{p}_s, \hat{p}^f\right) + W_D(\hat{p}^f, p^f) + \sum_s \alpha_s W_D\left(\hat{p}_s, p_s\right) \end{aligned}$$

where the last line is obtained from Lemma 2. Using the well known convergence property of the Wasserstein distance proven in [42] we find the following bound with probability $1 - \delta$ we have

$$\varepsilon_{p^T}(f) \leq W_D\left(\sum_s \alpha_s \hat{p}_s, \hat{p}^f\right) + \sqrt{\frac{2}{c'}\log\left(\frac{2}{\delta}\right)}\left(\frac{1}{N} + \sum_s \frac{\alpha_s}{N_s}\right) + \Lambda(p^{\boldsymbol{\alpha}}, p^T) + 2kM\phi(\lambda) \tag{21}$$

with c' corresponding to all *source* and *target* distributions under similar conditions as in [10]. □

B Numerical experiments

B.1 Simulated data

We generate a data set (X_0, Y_0) by drawing X_0 from a 3-dimensional Gaussian distribution with 3 cluster centers and standard deviation $\sigma = 0.8$. We keep the same number of examples for each cluster. To simulate the S *sources*, we apply S rotations to the input data X_0 around the x-axis. More precisely, we draw S equispaced angles θ_s from $[0, \frac{3}{2}\pi]$ and we get $X_s = \{\mathbf{x}_s^i\}$ as

$$\mathbf{x}_s^{i\top} = \mathbf{x}_0^{i\top} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta_s) & -sin(\theta_s) \\ 0 & sin(\theta_s) & cos(\theta_s) \end{bmatrix}. \tag{22}$$

To generate the *target* domain X, we follow the same procedure by randomly choosing an angle $\theta \in [0, \frac{3}{2}\pi]$. We keep the label set fixed, i.e. $Y_s = Y = Y_0$. In the following we report all the experiment we carried out on the simulated data, in which we also investigate to replace the exact Wasserstein distance by the the Bures-Wasserstein distance

$$BW(\mu_1, \mu_2)^2 = \|\mathbf{m}_1 - \mathbf{m}_2\|^2 + \text{Trace}\left(\Sigma_1 + \Sigma_2 - 2\left(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2}\right)^{1/2}\right), \tag{23}$$

where the $\mathbf{m}_i, \Sigma_i$ are respectively the first and second order moments of distribution μ_i for $i \in \{1, 2\}$. The BW distance has the advantage of having a complexity linear in the number of samples that can scale better to large dataset. We label this method variant with (B), while we refer to the exact OT as (E).

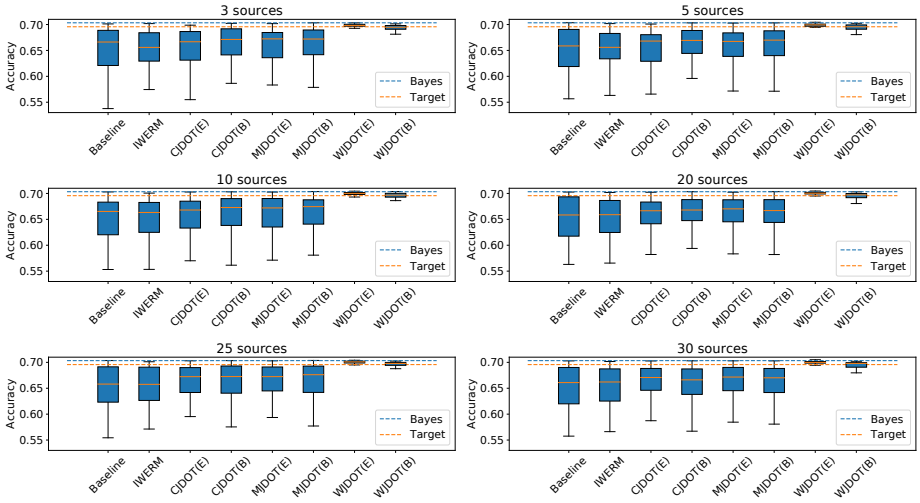

Figure 3: Methods' accuracy for varying the number of *sources* S.

Varying the number of *sources*. We keep the number of samples fixed and we vary the number of *sources* $S \in \{3, 5, 10, 20, 25, 30\}$. In Fig. 3 we report the accuracy of the different methods.

Varying the number of *source* samples. We fix the number of *sources* equal to 20 and the number of *target* samples to 300. Fig 4 and 5 show the methods accuracy for varying the number of *source* samples N_s in $[60, 180, 300]$ and the recovered $\boldsymbol{\alpha}$ weight for sample size 300, respectively.

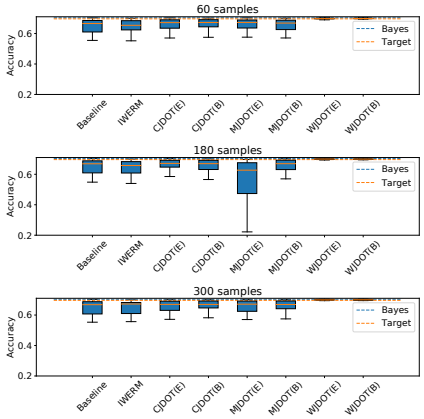

Figure 4: Methods' accuracy for varying the number of *source* domain samples.

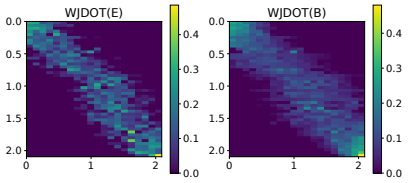

Figure 5: Recovered $\boldsymbol{\alpha}$ for an increasing rotation angle ($N = 300$).

Varying number of the *target* samples. We fix the number of *sources* equal to 20 and the number of samples $N_s = 300$, for each $s \in \{1, \cdots, S\}$. We let vary the number of *target* samples N in $[60, 180, 300]$ (Fig. 6).

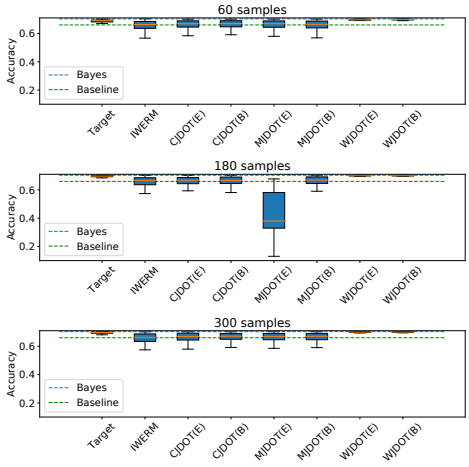

Figure 6: Methods' accuracy for varying the number of *target* samples

Varying the number of samples of all domains We fix the number of *sources* equal to 20. We let vary the number of *source* and *target* samples in $[60, 180, 300]$, by keeping $N_s = N$ for each $s \in \{1, \cdots, S\}$. We report the methods' accuracy in Fig. 4.

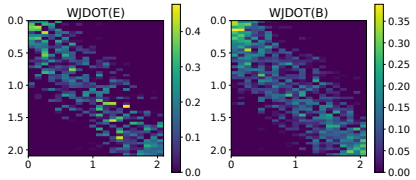

Figure 7: Recovered $\boldsymbol{\alpha}$ for an increasing rotation angle in small sample size is available ($N_s = N = 60$).

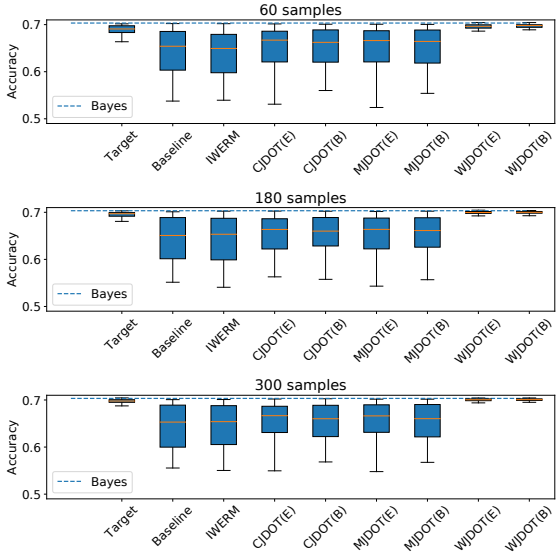

Figure 8: Methods' accuracy for varying the number of *source* and *target* samples

B.2 Object recognition

In Table 3 we report the *source* weights provided by `WJDOT`. In all cases, $\boldsymbol{\alpha}$ is a one-hot vector suggesting that only one *source* is meaningfully related to the *target* domain.

$\boldsymbol{\alpha}$	Amazon	dslr	webcam	Caltech10
Amazon	-	0	0	1
dslr	0	-	1	0
webcam	0	1	-	0
Caltech10	1	0	0	-

Table 3: $\boldsymbol{\alpha}$ weights

In the following, we propose an alternative strategy to the one proposed in Sec. 4 for the network parameters and early stopping validation. In particular, we use the weighted average accuracy of the trained classifier on the *source* distributions weighted by $\boldsymbol{\alpha}$, i.e.

$$\sum_{s=1}^{S} \alpha_s ACC(f, p_s). \tag{24}$$

To refer to this approach, we denote as $\texttt{WJDOT}^{acc}$, $\texttt{CJDOT}^{acc}$, $\texttt{MJDOT}^{acc}$ the WJDOT and the two JDOT extensions respectively. Let us remark that $\texttt{WJDOT}^{acc}$ is a way to reuse the weights $\boldsymbol{\alpha}$ that define the closest *source* distribution which are those that can give a better estimate of the performance of the current classifier. Table 4 is a full version of Table 1 in the paper, in which we also report the accuracy obtained by employing this validation strategy. We can observe that $\texttt{WJDOT}^{acc}$ provides good performances, comparable with both WJDOT and the other MSDA methods, but WJDOT still remains the state of the art.

Method	Amazon	dslr	webcam	Caltech10	AR
Baseline	93.13 ± 0.07	94.12 ± 0.00	89.33 ± 1.63	82.65 ± 1.84	5.75
IWERM [31]	93.30 ± 0.75	$\mathbf{100.00 \pm 0.00}$	89.33 ± 1.16	$\mathbf{91.19 \pm 2.57}$	2.75
$\texttt{CJDOT}^{acc}$ [10]	92.27 ± 0.83	97.06 ± 2.94	90.33 ± 2.33	86.19 ± 0.09	3.75
CJDOT [10]	93.74 ± 1.57	93.53 ± 4.59	90.33 ± 2.13	85.84 ± 1.73	3.75
$\texttt{MJDOT}^{acc}$ [10]	93.61 ± 0.04	98.82 ± 2.35	$\mathbf{91.00 \pm 1.53}$	85.22 ± 1.48	3.25
MJDOT [10]	94.12 ± 1.57	97.65 ± 2.88	90.27 ± 2.48	84.72 ± 1.73	3.75
$\texttt{WJDOT}^{acc}$	93.61 ± 0.09	$\mathbf{100.00 \pm 0.00}$	86.00 ± 2.91	85.49 ± 1.69	3.75
WJDOT	$\mathbf{94.23 \pm 0.90}$	$\mathbf{100.00 \pm 0.00}$	89.33 ± 2.91	85.93 ± 2.07	**2.25**
Target	95.77 ± 0.31	88.35 ± 2.76	99.87 ± 0.65	89.75 ± 0.85	-
Baseline+Target	94.78 ± 0.48	99.88 ± 0.82	100.00 ± 0.00	91.89 ± 0.69	-

Table 4: Accuracy on Caltech Office Dataset

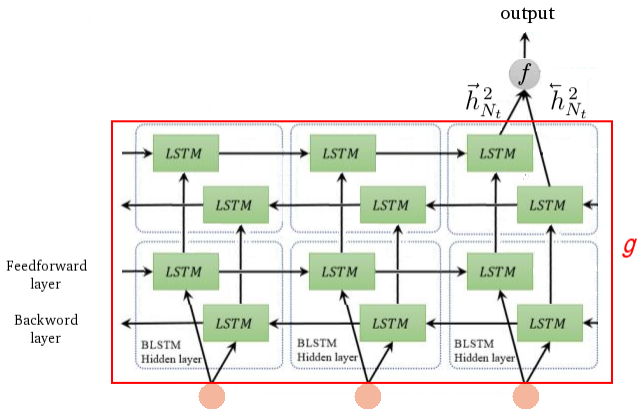

Figure 9: BLSTM architecture. A similar architecture is used for the multi-task learning approach: we use the same embedding function g and T classification functions f_t.

Music-speech discrimination The BLSTM-based model we adopted is shown in Fig. 9. Weights were initialized with Xavier initialization. Training is performed with Adam optimizer with 0.9

momentum and $\epsilon = e^{-8}$. Learning rate exponentially decays every epoch. We grid-researches the initial learning rate value and the decay rate.

In Table 5 we show the MSDA performances in the music-speech discrimination. In particular, for WJDOT and JDOT variants the validation strategy described in formula 24 has been employed. The Average Rank shows that $WJDOT$ is state of the art in music-speech discrimination with both validation strategies.

Method	**F16**	**Buccaneer2**	**Factory2**	**Destroyerengine**	**AR**
`Baseline`	69.67 ± 8.78	57.33 ± 7.57	83.33 ± 9.13	87.33 ± 6.72	9.25
`IWERM` [31]	72.22 ± 3.93	58.33 ± 5.89	85.00 ± 6.23	81.64 ± 3.33	8.75
`IWERM`$_{mtl}$ [31]	75.00 ± 0.00	66.67 ± 0.00	$\mathbf{100.00 \pm 0.00}$	98.33 ± 3.33	4.75
`DCTN` [18]	66.67 ± 3.61	68.75 ± 3.61	87.50 ± 12.5	94.44 ± 7.86	7.00
`M`^{3}`SDA` [15]	70.00 ± 4.08	61.67 ± 4.08	85.00 ± 11.05	83.33 ± 0.00	8.50
`CJDOT` [10]	59.50 ± 13.95	50.00 ± 0.00	83.33 ± 0.00	91.67 ± 0.00	9.75
`CJDOT`$_{mtl}$ [10]	83.83 ± 5.11	74.83 ± 1.17	$\mathbf{100.00 \pm 0.00}$	95.74 ± 16.92	3.25
`CJDOT`$^{acc}_{mtl}$ [10]	79.83 ± 4.74	74.83 ± 1.17	99.67 ± 1.63	$\mathbf{100.00 \pm 0.00}$	2.50
`MJDOT`[10]	66.33 ± 9.57	50.00 ± 0.00	83.33 ± 0.00	91.67 ± 0.00	9.50
`MJDOT`$_{mtl}$[10]	86.00 ± 4.55	72.83 ± 5.73	97.67 ± 3.74	97.74 ± 8.28	3.50
`MJDOT`$^{acc}_{mtl}$[10]	77.67 ± 5.12	69.00 ± 4.72	99.67 ± 1.63	99.83 ± 1.17	3.50
`WJDOT`	83.33 ± 0.00	58.33 ± 6.01	87.00 ± 6.05	89.00 ± 4.84	6.50
`WJDOT`$_{mtl}$	$\mathbf{87.17 \pm 4.15}$	74.83 ± 1.20	99.67 ± 1.63	99.67 ± 1.63	**2.00**
`WJDOT`$^{acc}_{mtl}$	83.00 ± 4.07	$\mathbf{75.00 \pm 0.00}$	$\mathbf{100.00 \pm 0.00}$	98.83 ± 3.34	**2.00**
`WJDOT`^{acc}	83.33 ± 0.00	58.33 ± 6.01	87.00 ± 6.05	89.00 ± 4.84	6.50
`Target`	73.67 ± 6.09	69.17 ± 7.50	77.33 ± 4.73	73.17 ± 9.90	-
`Baseline+Target`	71.06 ± 9.31	67.62 ± 11.92	85.33 ± 11.85	79.53 ± 10.05	-

Table 5: Accuracy on Music-Speech Dataset