Supplementary Material

The supplementary material is organized as follows. In Section A we provide proof of **Lemma 1**, **Lemma 2** and **Theorem 1**. For reader's convenience the results are repeated in this supplementary material. Section B recalls the MSDA-WJDOT algorithm and defines the projection to the simplex implemented in the algorithm. Finally, in Section C we present additional numerical experiments.

A PROOFS

A.1 PROOF OF LEMMA 1

Lemma 1. *For any hypothesis $f \in \mathcal{H}$, denote as $\varepsilon_{p_T}(f)$ and $\varepsilon_{p_S^{\boldsymbol{\alpha}}}(f)$, 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*

$$\varepsilon_{p_T}(f) \leq \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f) + B \cdot D_{TV}\left(p_S^{\boldsymbol{\alpha}}, p_T\right) \tag{1}$$

where $p_S^{\boldsymbol{\alpha}} = \sum_{j=1}^{J} \alpha_j p_{S,j}$ is a convex combination of the source distributions with weights $\boldsymbol{\alpha} \in \Delta^J$, 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

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

then using simple arguments, we have

$$\begin{aligned}
\varepsilon_{p_T}(f) &= \varepsilon_{p_T}(f) + \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f) - \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f) \\
&\leq \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f) + |\varepsilon_{p_T}(f) - \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f)| \\
&\leq \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f) + \int |p_S^{\boldsymbol{\alpha}}(x,y) - p_T(x,y)||L(y, f(x)| dx dy \\
&\leq \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f) + B \int \left|p_S^{\boldsymbol{\alpha}}(x,y) - p_T(x,y)\right| dx dy
\end{aligned} \tag{2}$$

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

A.2 PROOF OF THEOREM 1

The proof of this theorem follows the same steps as the one proposed by [4] and we reproduce it here for a sake of completeness.

Definition 1 (Probabilistic Transfer Lipschitzness – PLT Property). *Let p_S and p_T be respectively the source and target distributions. Let $\phi : \mathbb{R} \to [0,1]$. A labeling function $f : \mathcal{G} \to \mathbb{R}$ and a joint distribution $\pi \in \Pi(p_S, p_T)$ over p_S and p_T are ϕ-Lipschitz transferable if for all $\lambda > 0$, we have*

$$\mathrm{Prob}_{(x_S, x_T)\sim\pi}\left[|f(x_S) - f(x_T)|] > \lambda D(x_S, x_T)\right] \leq \phi(\lambda)$$

with D being a metric on $\mathcal{G}$.

This property provides a bound on the probability of finding a couple of source-target examples that are differently labeled in a $(1/\lambda)$-ball with respect to π and the metric D.

Definition 2. (Similarity measure) *Let $\mathcal{H}$ be a space of M-Lipschitz labelling functions. Assume that, for every $f \in \mathcal{H}$ and $x, x' \in \mathcal{G}$, $|f(x) - f(x')| \leq M$. The similarity between $p_S^{\boldsymbol{\alpha}}$ and p_T can defined [1, Def. 5] as*

$$\Lambda(p_S^{\boldsymbol{\alpha}}, p_T) = \min_{f \in \mathcal{H}} \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f) + \varepsilon_{p_T}(f), \tag{3}$$

where the risk is measured w.r.t. to a symmetric and k-Lipschitz loss function that satisfies the triangle inequality.

Lemma 2. *Let $\mathcal{H}$ be the space described in Definition 2 and assume that the function f^* minimizing the Similarity measure in Eq. 3 satisfies the PTL property. Then, for any $f \in \mathcal{H}$, we have*

$$\varepsilon_{p_T}(f) \leq W_D\left(p_S^{\boldsymbol{\alpha}}, p_T^f\right) + \Lambda(p_S^{\boldsymbol{\alpha}}, p_T) + kM\phi(\lambda), \tag{4}$$

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

Proof. We have that

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

where the second equality comes from the symmetry of the loss function and the third one is due to the fact that $\mathbb{E}_{(x,y)\sim p_T} L(f^\star(x), f(x)) = \mathbb{E}_{(x,y)\sim p_T^f} L(f^\star(x), f(x)) = \mathbb{E}_{x\sim\mu_T} L(f^\star(x), f(x))$ since the label y is not used in the expectation.

Now, we analyze the first term in the r.h.s. of the last inequality. Note that samples drawn from p_T^f distribution can be expressed as $(x_T, y_T^f) \sim p_T^f$ with $y_T^f = f(x_T)$.

$$\begin{aligned}
|\varepsilon_{p_T^f}(f^\star) - \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f^\star)| &= \left|\int_{\mathcal{G}\times\mathbb{R}} L(y, f^\star(x))(p_T^f(x,y) - p_S^{\boldsymbol{\alpha}}(x,y))dxdy\right| \\
&= \left|\int_{\mathcal{G}\times\mathbb{R}} L(y, f^\star(x))d(p_T^f - p_S^{\boldsymbol{\alpha}})\right| \\
&\leq \int_{(\mathcal{G}\times\mathbb{R})^2} \left|L(y_T^f, f^\star(x_T)) - L(y_{\boldsymbol{\alpha}}, f^\star(x_{\boldsymbol{\alpha}}))\right| d\pi^\star((x_{\boldsymbol{\alpha}}, y_{\boldsymbol{\alpha}}), (x_T, y_T^f)) \qquad (5) \\
&\leq \int_{(\mathcal{G}\times\mathbb{R})^2} \Bigg[\left|L(y_T^f, f^\star(x_T)) - L(y_T^f, f^\star(x_{\boldsymbol{\alpha}}))\right| \\
&\qquad + \left|L(y_T^f, f^\star(x_{\boldsymbol{\alpha}})) - L(y_{\boldsymbol{\alpha}}, f^\star(x_{\boldsymbol{\alpha}}))\right|\Bigg] d\pi^\star((x_{\boldsymbol{\alpha}}, y_{\boldsymbol{\alpha}}), (x_T, y_T^f)) \\
&\leq \int_{(\mathcal{G}\times\mathbb{R})^2} \left[k\left|f^\star(x_T) - f^\star(x_{\boldsymbol{\alpha}}))\right| + \left|L(y_T^f, f^\star(x_{\boldsymbol{\alpha}})) - L(y_{\boldsymbol{\alpha}}, f^\star(x_{\boldsymbol{\alpha}})\right|\right] d\pi^\star((x_{\boldsymbol{\alpha}}, y_{\boldsymbol{\alpha}}), (x_T, y_T^f)) \qquad (6) \\
&\leq kM\phi(\lambda) + \int_{(\mathcal{G}\times\mathbb{R})^2} \left[k\lambda D(x_T, x_{\boldsymbol{\alpha}}) + \left|L(y_T^f, f^\star(x_{\boldsymbol{\alpha}})) - L(y_{\boldsymbol{\alpha}}, f^\star(x_{\boldsymbol{\alpha}}))\right|\right] d\pi^\star((x_{\boldsymbol{\alpha}}, y_{\boldsymbol{\alpha}}), (x_T, y_T^f)) \qquad (7) \\
&\leq kM\phi(\lambda) + \int_{(\mathcal{G}\times\mathbb{R})^2} \left[\beta D(x_T, x_{\boldsymbol{\alpha}}) + \left|L(y_T^f, y_{\boldsymbol{\alpha}})\right|\right] d\pi^\star((x_{\boldsymbol{\alpha}}, y_{\boldsymbol{\alpha}}), (x_T, y_T^f)) \qquad (8) \\
&= kM\phi(\lambda) + W_D(p_S^{\boldsymbol{\alpha}}, p_T^f). \qquad (9)
\end{aligned}$$

Inequality in line (5) is due to the Kantorovich-Rubinstein theorem stating that for any coupling $\pi \in \Pi(p_S^{\boldsymbol{\alpha}}, p_T)$ the following inequality holds

$$\left|\int_{\mathcal{G}\times\mathbb{R}} L(y, f^\star(x))d(p_T^f - p_S^{\boldsymbol{\alpha}})\right| \leq \left|\int_{(\mathcal{G}\times\mathbb{R})^2} |L(y_T^f, f^\star(x_T)) - L(y_{\boldsymbol{\alpha}}, f^\star(x_{\boldsymbol{\alpha}})| d\pi((x_{\boldsymbol{\alpha}}, y_{\boldsymbol{\alpha}}), (x_T, y_T^f))\right|,$$

followed by an application of the triangle inequality. Since, the above inequality applies for any coupling, it applies also for $\pi^\star$. Inequality (6) is due to the assumption that the loss function is k-Lipschitz in its second argument. Inequality (7) derives from the PTL property with probability $1-\phi(\lambda)$ of $f^\star$ and $\pi^\star$. In addition, taking into account that the difference between two 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 (8) is obtained from the symmetry of $D(\cdot,\cdot)$, the triangle inequality on the loss and by posing $k\lambda=\beta$. $\square$

First we need to prove the following Lemma.

Lemma 3. *For any distributions $\hat{p}_{S,j}, p_{S,j}$ and $\boldsymbol{\alpha} \in \Delta^J$ in the simplex we have*

$$W_D\left(\sum_{j=1}^J \alpha_j \hat{p}_{S,j}, \sum_{j=1}^J \alpha_j p_{S,j}\right) \leq \sum_{j=1}^J \alpha_j W_D\left(\hat{p}_{S,j}, p_{S,j}\right).$$

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

$$W_D(p,p') = \min_{\pi\in\Pi(p,p')} \int D(\mathbf{v},\mathbf{v}')\pi(\mathbf{v},\mathbf{v}')d\mathbf{v}d\mathbf{v}', \tag{10}$$

where $\Pi(p,p') = \{\pi | \int \pi(\mathbf{v},\mathbf{v}')d\mathbf{v}' = p(\mathbf{v}), \int \pi(\mathbf{v},\mathbf{v}')d\mathbf{v} = p'(\mathbf{v}')\}$. Let $\pi^*_{S,j}$ be the optimal OT matrix between $\hat{p}_{S,j}$ and $p_{S,j}$. It is obvious to see that $\sum_{j=1}^J \alpha_j \pi^*_{S,j}$ respects the marginal constraints for $W_D\left(\sum_{j=1}^J \alpha_j \hat{p}_{S,j}, \sum_{j=1}^J \alpha_j p_{S,j}\right)$, i.e. $\sum_{j=1}^J \alpha_j \pi^*_{S,j} \in \Pi\left(\sum_{j=1}^J \alpha_j \hat{p}_{S,j}, \sum_{j=1}^J \alpha_j p_{S,j}\right)$. Hence, $\sum_{j=1}^J \alpha_j \pi^*_{S,j}$ is a feasible solution for the OT problem and, consequently, the cost for this feasible solution is greater or equal than the optimal value $W_D\left(\sum_{j=1}^J \alpha_j \hat{p}_{S,j}, \sum_{j=1}^J \alpha_j p_{S,j}\right)$. Since $\int D(\mathbf{v},\mathbf{v}') \sum_{j=1}^J \alpha_j \pi^*_{S,j}(\mathbf{v},\mathbf{v}')d\mathbf{v}d\mathbf{v}' = \sum_{j=1}^J \alpha_j W_D\left(\hat{p}_{S,j}, p_{S,j}\right)$ we recover the Lemma above. $\square$

We can now prove **Theorem 1**, which we also restate for the convenience of the reader.

Theorem 1. *Under the assumptions of Lemma 2, let $\hat{p}_{S,j}$ be j-th source empirical distributions of N_j samples and $\hat{p}_T$ the empirical target distribution with N_T samples. Then for all $\lambda > 0$, with $\beta = \lambda k$ in the ground metric D we have with probability $1-\eta$*

$$\varepsilon_{p_T}(f) \leq W_D\left(\hat{p}_S^{\boldsymbol{\alpha}}, \hat{p}_T^f\right) + \sqrt{\frac{2}{c'}\log\frac{2}{\eta}}\left(\frac{1}{N_T} + \sum_{j=1}^J \frac{\alpha_j}{N_j}\right) + \Lambda(p_S^{\boldsymbol{\alpha}}, p_T) + kM\phi(\lambda). \tag{11}$$

Proof. By the triangle inequality we have that

$$\begin{aligned} W_D\left(\sum_{j=1}^J \alpha_j p_{S,j}, p_T^f\right) &\leq W_D\left(\sum_{j=1}^J \alpha_j \hat{p}_{S,j}, \hat{p}_T^f\right) + W_D(\hat{p}_T^f, p_T^f) + W_D\left(\sum_{j=1}^J \alpha_j \hat{p}_{S,j}, \sum_{j=1}^J \alpha_j p_{S,j}\right) \\ &\leq W_D\left(\sum_{j=1}^J \alpha_j \hat{p}_j, \hat{p}_T^f\right) + W_D(\hat{p}_T^f, p_T^f) + \sum_{j=1}^J \alpha_j W_D\left(\hat{p}_{S,j}, p_{S,j}\right) \end{aligned}$$

where the last inequality follows from Lemma 3. Using the well known convergence property of the Wasserstein distance proven in [2] we find the following bound with probability $1-\eta$

$$\varepsilon_{p_T}(f) \leq W_D\left(\sum_{j=1}^J \alpha_j \hat{p}_{S,j}, \hat{p}_T^f\right) + \sqrt{\frac{2}{c'}\log\left(\frac{2}{\eta}\right)}\left(\frac{1}{N_T} + \sum_{j=1}^J \frac{\alpha_j}{N_j}\right) + \Lambda(p_S^{\boldsymbol{\alpha}}, p_T) + 2kM\phi(\lambda) \tag{12}$$

with c' corresponding to all *source* and *target* distributions under similar conditions as in [4]. $\square$

B THE ALGORITHM

We recall here the algorithm we proposed to solve the MSDA-WJDOT problem (Algorithm 1). P_{Δ^J} is the projection to the simplex $\Delta^J = \{\boldsymbol{\alpha} \in \mathbb{R}^J | \sum_{j=1}^J \alpha_j = 1, \alpha_j \geq 0\}$ defined as

$$P_{\Delta^J}(\boldsymbol{w}) = \underset{\boldsymbol{\alpha} \in \Delta^J}{\text{argmin}} \|\boldsymbol{w} - \boldsymbol{\alpha}\|. \tag{13}$$

We implemented it by using Algorithm 2, firstly proposed in [5].

Algorithm 1 Optimization for MSDA-WJDOT

Initialise $\boldsymbol{\alpha} = \frac{1}{J}\mathbf{1}_J$ and $\boldsymbol{\theta}$ parameters of $f_{\boldsymbol{\theta}}$ and steps $\mu_{\boldsymbol{\alpha}}$ and $\mu_{\boldsymbol{\theta}}$.
repeat
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \mu_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} W_D\left(\hat{p}_T^f, \sum_{j=1}^J \alpha_j \hat{p}_{S,j}\right)$
$\boldsymbol{\alpha} \leftarrow P_{\Delta^J}\left(\boldsymbol{\alpha} - \mu_{\boldsymbol{\alpha}} \nabla_{\boldsymbol{\alpha}} W_D(\hat{p}_T^f, \sum_{j=1}^J \alpha_j \hat{p}_{S,j})\right)$
until Convergence

Algorithm 2 Projection to the simplex [5]

Sort $\boldsymbol{w}$ into $\boldsymbol{u}$: $u_1 \geq \cdots \geq u_J$.
Set $K := \max_{1 \leq k \leq J}\{k | (\sum_{j=1}^k u_j - 1/k < u_k\}$.
Set $\tau := (\sum_{j=1}^K u_j - 1)/K$.
For $j = 1, \dots, J$ set $\alpha_j := \max\{w_j - \tau, 0\}$.

C NUMERICAL EXPERIMENTS

C.1 SIMULATED DATA

Domain shift 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 J *sources*, we apply J rotations to the input data X_0 around the x-axis. More precisely, we draw J equispaced angles θ_j from $[0, \frac{3}{2}\pi]$ and we get $X_j = \{\mathbf{x}_j^i\}$ as

$$\mathbf{x}_j^{i\top} = \mathbf{x}_0^{i\top} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta_j) & -sin(\theta_j) \\ 0 & sin(\theta_j) & cos(\theta_j) \end{bmatrix}. \tag{14}$$

To generate the *target* domain X_T, we follow the same procedure by randomly choosing an angle $\theta_T \in [0, \frac{3}{2}\pi]$. We keep the label set fixed, i.e. $Y_j = Y_T = Y_0$. Note that in this case the embedding function g is the identity function and, hence, $\mathcal{X} \equiv \mathcal{G}$. 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

$$BW(\mu_S, \mu_T)^2 = \|\mathbf{m}_S - \mathbf{m}_T\|^2 + \text{Trace}\left(\Sigma_S + \Sigma_T - 2\left(\Sigma_S^{1/2}\Sigma_T\Sigma_S^{1/2}\right)^{1/2}\right), \tag{15}$$

where the $\mathbf{m}_S, \Sigma_S$ are respectively the first and second order moments of distribution μ_S (and similarly for $\mathbf{m}_T, \Sigma_T$). 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).

In the following, we investigate the performance of MSDA-WJDOT at varying of the number of *sources* J, *source* samples N_j, and *target* samples N_T. We compare the proposed approch with other MSDA methods and with the `Baseline`, `Target`, `Bayes` classification.

- *Varying the number of sources:* we keep the number of samples fixed in both *sources* and *target* datasets (s.t. $N_j = N_T$ $\forall j$) and we vary the number of *sources* $J \in \{3, 5, 10, 20, 25, 30\}$. In Fig. 1 we report the accuracy of the different methods.

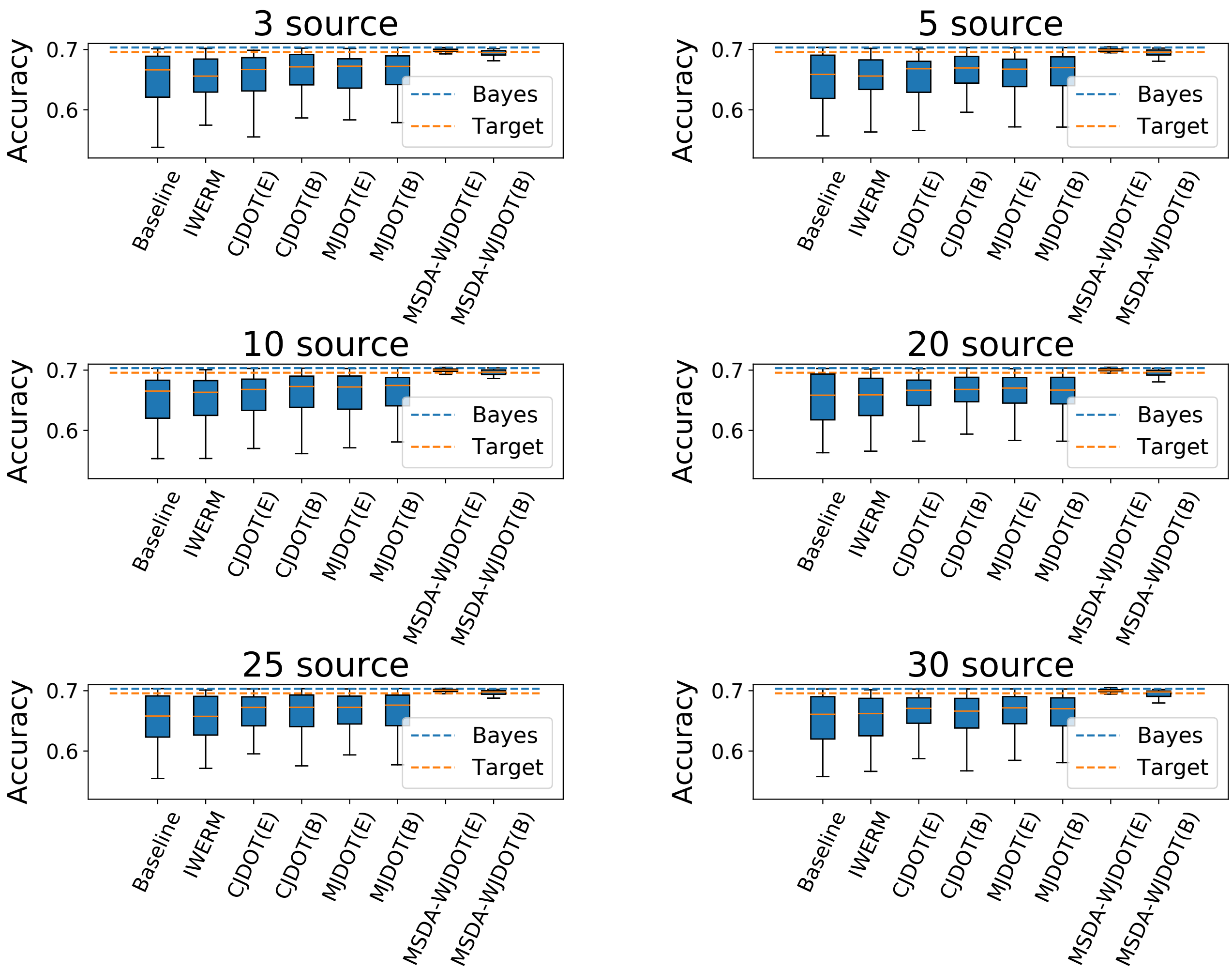

Figure 1: Methods' accuracy for varying the number of *sources* J.

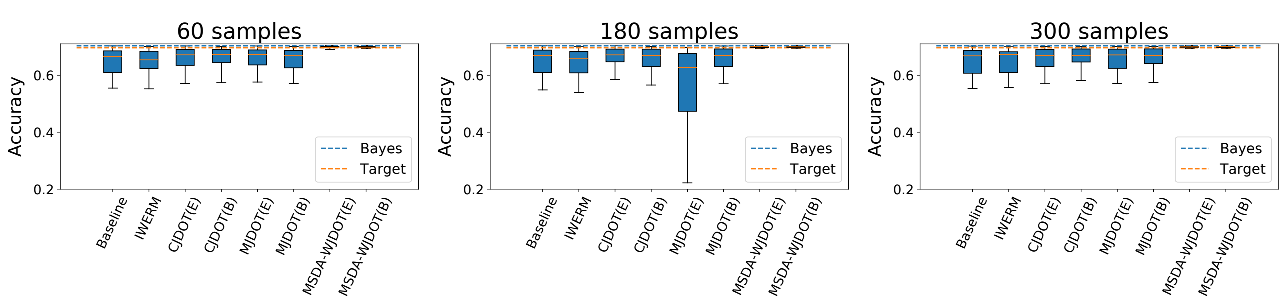

Figure 2: Methods' accuracy for varying the number of *source* samples.

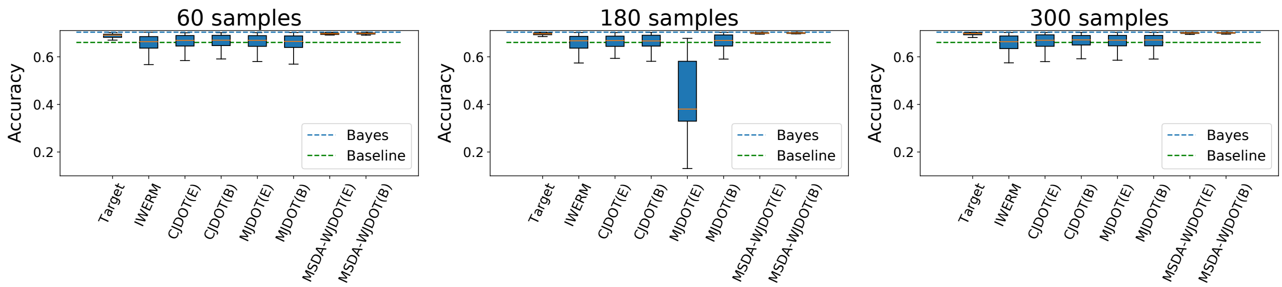

Figure 3: Methods' accuracy for varying the number of *target* samples

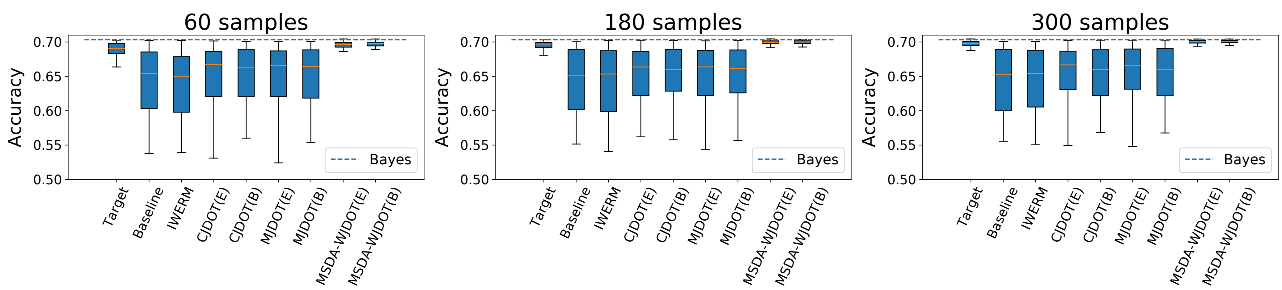

Figure 4: Methods' accuracy for varying the number of *source* and *target* samples

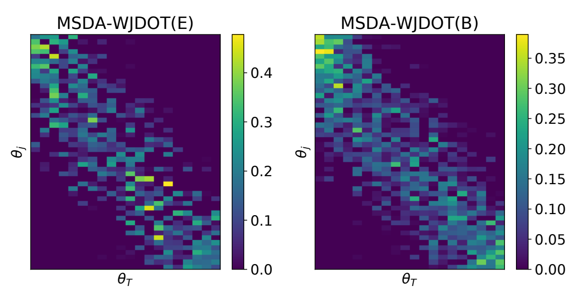

Figure 5: Recovered $\boldsymbol{\alpha}$ with small sample size ($N_j = N_T = 60$).

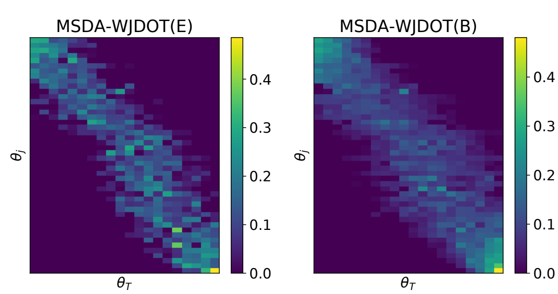

Figure 6: Recovered $\boldsymbol{\alpha}$ for $N_j = N_T = 300$.

- *Varying the number of source samples:* we fix the number of *sources* J equal to 20 and the number of *target* samples N_T to 300. Fig 2 and 6 show the methods accuracy for varying the number of *source* samples N_j in $\{60, 180, 300\}$

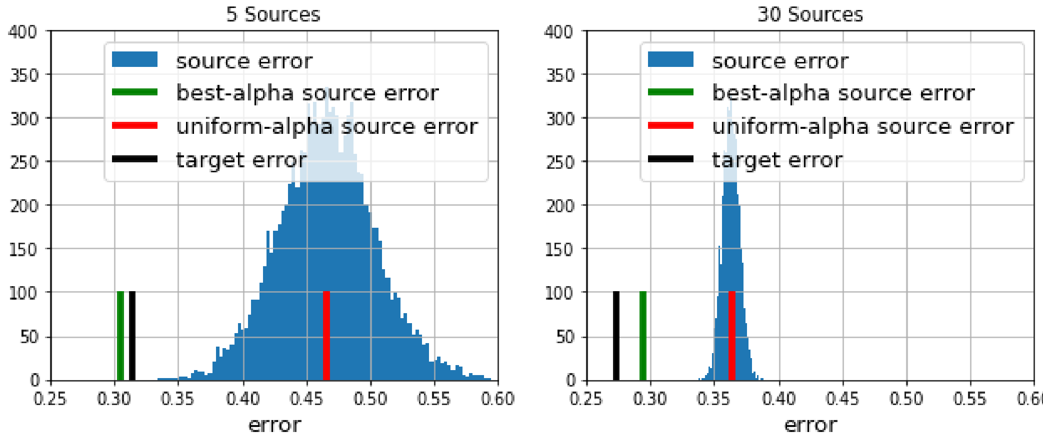

Figure 7: Examples of source error and target error when the function f is the function learned by our approach (instead of the one minimizing Λ in (**??**)). The blue curve represents an histogram of the α-weighted source error for 10000 random α. The x-axis represents the value of the error and the y-axis the count. The green line corresponds to the source error for the learned α, red one gives the error for an uniform alpha and the black one represents the target error (the height of the lines has been arbitrarily set for a sake of clarity). We can see that for both 5 (Left) and 30 sources (Right) the learned $alpha$ leads to lower source error even though α has been optimized for aligning joint distributions.

and the recovered $\boldsymbol{\alpha}$ weight for $N_j = 300$, respectively.

- *Varying the number of the target samples:* we fix $J = 20$ and $N_j = 300$, with $1 \leq j \leq J$. We let vary the number of *target* samples N_T in $\{60, 180, 300\}$ (Fig. 3).
- *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 $N_j = N_T$ with $1 \leq j \leq J$. We report the methods' accuracy in Fig. 2.

In all experiments `MSDA-WJDOT` significantly outperfoms `CJDOT`, `MJDOT`, `IWERM` and the `Baseline`. Both `MSDA-WJDOT(E)` and `MSDA-WJDOT(B)` provide a better or at least comparable performance w.r.t. the `Target` method, in which the labels of the *target* dataset are used. In Fig. 5 and 6 we show the recovered weights $\boldsymbol{\alpha}$ for $N_j = N_T = 60$ and $N_j = N_T = 300$, respectively. In both cases, the x- axis reports different random *target* angles in the $[0, \frac{3}{2}\pi]$ interval (ordered by increasing angles), whereas the y-axis represents the *source* angles ordered such that $\theta_j \leq \theta_{j+1}$, $1 \leq j \leq J-1$. As we can see, the weights are higher along the diagonal meaning that `MSDA-WJDOT` always rewards the *sources* with angle closest to θ_T.

C.2 REAL DATA

In the section, we introduce a new strategy for the validation, in alternative to the one based on SSE proposed in Sec. 3.2. We propose to employ the accuracy of the learned classifier f on the *source* datasets and weighted by $\boldsymbol{\alpha}$, i.e.

$$\sum_{j=1}^{J} \alpha_j ACC_{S,j}(f), \tag{16}$$

with $ACC_{S,j}(f) = \frac{\#\{f(x_j^i)=y_j^i\}}{N_j}$. To refer to this approach, we denote as `MSDA-WJDOT`^{acc}, `CJDOT`^{acc}, `MJDOT`^{acc} the MSDA-WJDOT and the two JDOT extensions respectively. Let us remark that `MSDA-WJDOT`^{acc} is a way to reuse the weights $\boldsymbol{\alpha}$ that provide the closest *source* distributions which, hence, are supposed to give a better estimate of the performance of the current classifier.

Object recognition In Table 3 we report the *source* weights provided by MSDA-WJDOT. In all cases, $\boldsymbol{\alpha}$ is a one-hot vector suggesting that only one *source* is meaningfully related to the *target* domain. This is in line with the results on single-source DA found in [3] in which the *source* domain providing the highest accuracy corresponds to the one selected by `MSDA-WJDOT`.

Target	*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

Table 4 is a full version of Table 1 in the paper, in which we also show the accuracy obtained by employing the validation strategy introduced in Eq. 16. We can observe that $\texttt{MSDA-WJDOT}^{acc}$ provides good performances, comparable with both `MSDA-WJDOT` and the other MSDA methods, but `MSDA-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	6.75
`IWERM` [9]	93.30 ± 0.75	$\mathbf{100.00 \pm 0.00}$	89.33 ± 1.16	$\mathbf{91.19 \pm 2.57}$	3.25
$\texttt{CJDOT}^{acc}$ [4]	92.27 ± 0.83	97.06 ± 2.94	90.33 ± 2.33	86.19 ± 0.09	4.50
`CJDOT` [4]	93.74 ± 1.57	93.53 ± 4.59	90.33 ± 2.13	85.84 ± 1.73	4.50
$\texttt{MJDOT}^{acc}$ [4]	93.61 ± 0.04	98.82 ± 2.35	91.00 ± 1.53	85.22 ± 1.48	3.75
`MJDOT` [4]	94.12 ± 1.57	97.65 ± 2.88	90.27 ± 2.48	84.72 ± 1.73	4.50
$\texttt{JCPOT}^{*}$ [8]	79.23 ± 3.09	81.77 ± 2.81	93.93 ± 0.60	77.91 ± 0.45	7.25
$\texttt{WBT}^{*}$ [6]	59.86 ± 2.48	60.99 ± 2.15	64.13 ± 2.38	62.80 ± 1.61	9.50
$\texttt{WBT}^{*}_{reg}$ [6]	92.74 ± 0.45	95.87 ± 1.43	$\mathbf{96.57 \pm 1.76}$	85.01 ± 0.84	5.00
$\texttt{MSDA-WJDOT}^{acc}$	93.61 ± 0.09	$\mathbf{100.00 \pm 0.00}$	86.00 ± 2.91	85.49 ± 1.69	4.25
`MSDA-WJDOT`	$\mathbf{94.23 \pm 0.90}$	$\mathbf{100.00 \pm 0.00}$	89.33 ± 2.91	85.93 ± 2.07	**2.75**
`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 4: Accuracy on Caltech Office Dataset. Results of methods marked by * are from [6].

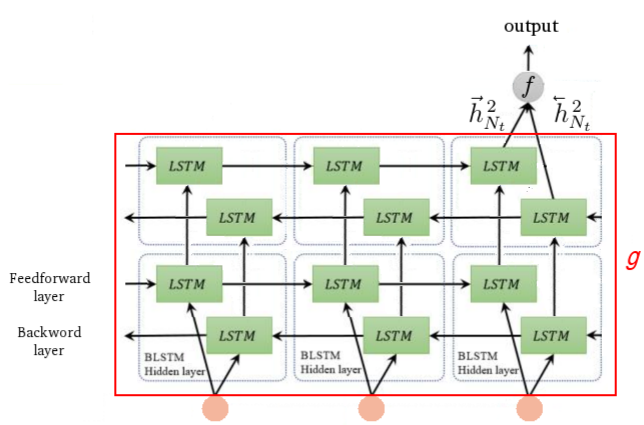

Figure 8: BLSTM architecture. A similar architecture is used for the multi-task learning approach: we use the same embedding function g and J classification functions f_j.

Music-speech discrimination The model we adopted is shown in Fig. 8, where g is a two-layers Bidirectional Long Short-Term Memory (BLSTM) that feeds the one feed-forward layer f with the last hidden state. 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-research 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 MSDA-WJDOT and JDOT variants the validation strategy described in formula 16 has been employed. Results show that, although this is a valid strategy, early stopping based on SSE described in Sec. 4 always outperforms. The Average Rank shows that MSDA-WJDOT is state of the art in music-speech discrimination.

Method	F16	Buccaneer2	Factory2	Destroyerengine	AR
Baseline	69.67 ± 8.78	57.33 ± 7.57	83.33 ± 9.13	87.33 ± 6.72	11.25
IWERM [9]	72.22 ± 3.93	58.33 ± 5.89	85.00 ± 6.23	81.64 ± 3.33	10.75
IWERM_{mtl} [9]	75.00 ± 0.00	66.67 ± 0.00	**100.00 ± 0.00**	98.33 ± 3.33	5.50
DCTN [10]	66.67 ± 3.61	68.75 ± 3.61	87.50 ± 12.5	94.44 ± 7.86	8.50
$\text{M}^{\mathbf{3}}\text{SDA}$ [7]	70.00 ± 4.08	61.67 ± 4.08	85.00 ± 11.05	83.33 ± 0.00	10.25
CJDOT [4]	59.50 ± 13.95	50.00 ± 0.00	83.33 ± 0.00	91.67 ± 0.00	11.50
CJDOT_{mtl} [4]	83.83 ± 5.11	74.83 ± 1.17	**100.00 ± 0.00**	95.74 ± 16.92	4.00
CJDOT_{mtl}^{acc} [4]	79.83 ± 4.74	74.83 ± 1.17	99.67 ± 1.63	**100.00 ± 0.00**	3.50
MJDOT[4]	66.33 ± 9.57	50.00 ± 0.00	83.33 ± 0.00	91.67 ± 0.00	11.50
MJDOT_{mtl}[4]	86.00 ± 4.55	72.83 ± 5.73	97.67 ± 3.74	97.74 ± 8.28	4.00
MJDOT_{mtl}^{acc}[4]	77.67 ± 5.12	69.00 ± 4.72	99.67 ± 1.63	99.83 ± 1.17	4.75
JCPOT^{*}[8]	79.23 ± 3.09	81.77 ± 2.81	93.93 ± 0.60	77.91 ± 0.45	7.50
WBT^{*}[6]	59.86 ± 2.48	60.99 ± 2.15	64.13 ± 2.38	62.80 ± 1.61	13.00
WBT_{reg}^{*}[6]	**92.74 ± 0.45**	**95.87 ± 1.43**	96.57 ± 1.76	85.01 ± 0.84	4.25
MSDA-WJDOT	83.33 ± 0.00	58.33 ± 6.01	87.00 ± 6.05	89.00 ± 4.84	8.00
MSDA-WJDOT_{mtl}	87.17 ± 4.15	74.83 ± 1.20	99.67 ± 1.63	99.67 ± 1.63	**2.75**
$\text{MSDA-WJDOT}_{mtl}^{acc}$	83.00 ± 4.07	75.00 ± 0.00	**100.00 ± 0.00**	98.83 ± 3.34	3.50
MSDA-WJDOT^{acc}	83.33 ± 0.00	58.33 ± 6.01	87.00 ± 6.05	89.00 ± 4.84	8.00
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. Results of methods marked by * are from [6].