ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230459

Multi-Source Domain Adaptation Through Dataset Dictionary Learning in Wasserstein Space

Eduardo Montesuma[a;*], **Fred Maurice Ngole Mboula**[a] **and Antoine Souloumiac**[a]

[a]Université Paris-Saclay, CEA, List, F-91120 Palaiseau France

Abstract. This paper seeks to solve Multi-Source Domain Adaptation (MSDA), which aims to mitigate data distribution shifts when transferring knowledge from multiple labeled source domains to an unlabeled target domain. We propose a novel MSDA framework based on dictionary learning and optimal transport. We interpret each domain in MSDA as an empirical distribution. As such, we express each domain as a Wasserstein barycenter of dictionary atoms, which are empirical distributions. We propose a novel algorithm, DaDiL, for learning via mini-batches: (i) atom distributions; (ii) a matrix of barycentric coordinates. Based on our dictionary, we propose two novel methods for MSDA: DaDil-R, based on the reconstruction of labeled samples in the target domain, and DaDiL-E, based on the ensembling of classifiers learned on atom distributions. We evaluate our methods in 3 benchmarks: Caltech-Office, Office 31, and CRWU, where we improved previous state-of-the-art by 3.15%, 2.29%, and 7.71% in classification performance. Finally, we show that interpolations in the Wasserstein hull of learned atoms provide data that can generalize to the target domain.

1 Introduction

Traditional Machine Learning (ML) works under the assumption that training and test data follow a single probability distribution. Indeed, the Empirical Risk Minimization (ERM) framework of [37] measures generalization regarding an unknown probability distribution from which training and test data are sampled. Nonetheless, as [25] remarks, this is seldom the case in realistic applications due to changes in how the data is acquired. This results in a change in the data distribution, or distributional shift that motivates the field of Domain Adaptation (DA).

DA is an important framework where one assumes labeled data from a source domain and seeks to adapt models to an unlabeled target domain. When multiple source domains are available, one has a Multi-Source DA (MSDA) setting. This problem is more challenging as one has multiple distributional shifts co-occurring, that is, between sources and between sources and the target. In this work, we assume that the domain shifts have regularities that can be learned and leveraged for MSDA. In this context, Optimal Transport (OT) is a mathematical theory useful for DA, as it allows for the comparison and matching probability distributions. Previous works employed OT for the single-source case, as in [6, 5, 9], and MSDA as in [17, 18, 35].

In parallel, Dictionary Learning (DiL) expresses a set of vectors as weighted combinations of dictionary elements, named atoms. Previous works proposed OT for DiL over histogram data, such as [29] and [32]. Nonetheless, when data is high-dimensional, modeling distributions as histograms is intractable due the curse of dimensionality, which limits the use of previous DiL works for MSDA.

Contributions. In this paper we propose a novel DiL framework (section 4), for distributions represented as point clouds. We further explore (section 4.2) two ways of using DiL for MSDA, by reconstructing labeled samples in the target domain, and by ensembling classifiers learned with labeled data from atoms. In addition, we justify these methods theoretically through results in the literature [25, Theorem 2], and through novel theoretical results (i.e., theorem 2). To the best of our knowledge this is the first work to propose a DiL of point clouds, and to explore the connections between DiL of distributions and MSDA.

Paper Organization. Section 2 covers the related literature. Section 3 covers the necessary background, i.e., DA, OT and DiL concepts. Section 4 presents our framework. Section 5 explores our experiments in MSDA. Section 6 discusses our results. Finally, section 7 concludes our paper.

2 Related Work

There are mainly two methodologies in DA. The first, shallow DA, leverages pre-existing feature extractors and performs adaptation either by re-weighting or transforming source domain data to resemble target domain data. The second, deep DA, uses source and target domain data during the training of a Deep Neural Net (DNN), so that learned features are independent of distributional shift.

There are at least 3 classes of shallow DA methods: (i) importance re-weighting strategies [34], which give importance to source samples similar to the target domain, (ii) projection-based methods [21], which seek a sub-space where distributions share common characteristics, and (iii) OT-based methods [6], which use OT for matching, or calculating distances between distributions.

For deep DA, methods penalize encoder parameters that map source and target distributions to different locations in the latent space. As a consequence, deep DA is more complex than shallow DA, since encoder parameters are free. Examples of deep DA methods include [13], who uses an adversarial loss, and [9, 33], who use OT as a loss function between distributions of latent representations.

For MSDA, some works generalize previous single-source baselines. For instance, [23] proposes a moment-matching strategy across the different domains. [35] proposes weighting source domains linearly, then applying the Joint Distribution Optimal Transport (JDOT) strategy of [5]. This approach combines notions of importance weighting and OT-based DA. [17, 18] generalize the approach of [6],

* Corresponding Author. Email: eduardo.fernandes-montesuma@cea.fr
Accepted as a conference paper at the 26th European Conference on Artificial Intelligence.

Please check ArXiv or contact the authors for any appendices or supplementary material mentioned in the paper.

by first calculating a Wasserstein barycenter of the different source domains, then transporting the barycenter to the target domain.

In parallel, DiL is a representation learning technique, that was previously used in DA by [16]. However, classic DiL lacks a probabilistic interpretation. In this context, OT offers a probabilistic foreground for DiL, when data is represented through histograms. This is done by either using the Sinkhorn divergence [7] as the objective function [29], or by aggregating atoms in a Wasserstein space [32]. Nonetheless, in the context of DA it is computationally intractable to bin the feature space, which is commonly high-dimensional. This issue hinders the applicability of previous DiL approaches for DA. In contrast, we propose a new OT-inspired DiL framework for point clouds, which makes it suitable for MSDA.

3 Background

3.1 Domain Adaptation

In ML, learning a classifier consists on estimating $h : \mathcal{X} \to \mathcal{Y}$, among a set of functions $\mathcal{H}$, where $\mathcal{X}$ (e.g., $\mathbb{R}^d$) is the *feature space* and $\mathcal{Y}$ (e.g., $\{1, \cdots, n_c\}$) is the *label space*. This estimation is done via *risk minimization* [36],

$$h^\star = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \mathcal{R}_Q(h) = \underset{\mathbf{x} \sim Q}{\mathbb{E}}[\mathcal{L}(h(\mathbf{x}), h_0(\mathbf{x}))], \tag{1}$$

for a loss function $\mathcal{L}$, a distribution Q, and a ground-truth labeling function h_0. $\mathcal{R}_Q$ is known as *true risk*. Since Q and h_0 are seldom known *a priori*, it is unfeasible to directly minimize equation 1. In practice, one acquires a *dataset*, $\{(\mathbf{x}_i^{(Q)}, y_i^{(Q)})\}_{i=1}^n$, with $\mathbf{x}_i^{(Q)} \overset{\text{i.i.d.}}{\sim} Q$ and $y_i^{(Q)} = h_0(\mathbf{x}_i^{(Q)})$ and minimizes the *empirical risk*,

$$\hat{h} = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \hat{\mathcal{R}}_Q(h) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(h(\mathbf{x}_i^{(Q)}), y_i^{(Q)}).$$

Henceforth $\mathbf{x}_i^{(Q)}$ denotes a feature vector sampled from the marginal $Q(X)$. Likewise, $y_i^{(Q)}$ denotes its corresponding label. We denote its corresponding one-hot encoding (hard-labels) or probability vector (soft-labels) by $\mathbf{y}_i^{(Q)}$.

As discussed in [27], if training and test data are i.i.d. from Q, $\mathcal{R}_Q \to \hat{\mathcal{R}}_Q$ as $n \to \infty$. Nonetheless this assumption is restrictive, as it disregards the distributional heterogeneity within training data, and between train and test data, which motivates DA [21]. Following [21], a domain $\mathcal{D} = (\mathcal{X}, Q(X))$ is a pair of a feature space, and a feature distribution. In DA, one has different domains, i.e., a labeled source $\mathcal{D}_S$ with samples $\{(\mathbf{x}_i^{(Q_S)}, y_i^{(Q_S)})\}_{i=1}^{n_{Q_S}}$ and a target $\mathcal{D}_T$ with samples $\{\mathbf{x}_j^{(Q_T)}\}_{j=1}^{n_{Q_T}}$. In practice, one assumes a shared feature space (e.g., $\mathbb{R}^d$), so that domains differ in their distribution, $Q_S(X) \neq Q_T(X)$. This is known in the literature as *distributional shift*. The goal of DA is improving performance on the target, given knowledge from the source domain. We investigate MSDA, that is, DA between labeled sources $\{\mathcal{D}_{S_\ell}\}_{\ell=1}^{N_S}$ and an unlabeled target $\mathcal{D}_T$.

3.2 Optimal Transport

OT is a field of mathematics widely used in DA and ML. Henceforth we focus on computational OT. We refer readers to [19] and [24] for further background on this topic. Let $\mathbf{x}_i^{(P)} \overset{i.i.d.}{\sim} P$ (resp. $\mathbf{x}_j^{(Q)} \overset{i.i.d.}{\sim} Q$). We P and Q empirically using mixtures of Dirac deltas,

$$\hat{P}(\mathbf{x}) = \frac{1}{n_P} \sum_{i=1}^{n_P} \delta(\mathbf{x} - \mathbf{x}_i^{(P)}). \tag{2}$$

We refer to $\hat{P}$ as a point cloud, and $\mathbf{X}^{(P)} = [\mathbf{x}_1^{(P)}, \cdots, \mathbf{x}_{n_P}^{(P)}] \in \mathbb{R}^{n_P \times d}$ to its *support*. The Kantorovich formulation of OT seeks an OT plan, $\pi \in \mathbb{R}^{n_P \times n_Q}$ that *preserves mass*,

$$\Pi(\hat{P}, \hat{Q}) := \{\pi : \sum_i \pi_{i,j} = 1/n_Q; \sum_j \pi_{i,j} = 1/n_P\}.$$

where $\pi_{i,j}$ denotes how much mass $\mathbf{x}_i^{(P)}$ sends to $\mathbf{x}_j^{(Q)}$. In this sense, the OT problem between $\hat{P}$ and $\hat{Q}$ is,

$$\pi^\star = \text{OT}(\mathbf{X}^{(P)}, \mathbf{X}^{(Q)}) = \underset{\pi \in \Pi(\hat{P}, \hat{Q})}{\operatorname{argmin}} \langle \mathbf{C}, \pi \rangle_F, \tag{3}$$

where $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product and $C_{i,j} = c(\mathbf{x}_i^{(P)}, \mathbf{x}_j^{(Q)})$ is called *ground-cost matrix*. This is a linear program on the variables $\pi_{i,j}$, which has computational complexity $\mathcal{O}(n^3 \log n)$. Given π, one often wants to map samples from P into Q, which can be done through the *barycentric projection* [6],

$$T_\pi(\mathbf{x}_i^{(P)}) = \underset{\mathbf{x} \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{j=1}^{n_Q} \pi_{i,j} c(\mathbf{x}, \mathbf{x}_j^{(Q)}).$$

When c is the Euclidean distance, the barycentric projection has closed form,

$$T_\pi(\mathbf{x}_i^{(P)}) = n_P \sum_{j=1}^{n_Q} \pi_{i,j} \mathbf{x}_j^{(Q)}, \tag{4}$$

or $T_\pi(\mathbf{X}^{(P)}) = n_P \pi \mathbf{X}^{(Q)}$ in short.

Optimal Transport for Domain Adaptation. In the semminal works of [6], the authors proposed using OT for DA, under the assumption that there is $T : \mathbb{R}^d \to \mathbb{R}^d$ such that,

$$T_\sharp Q_S = Q_T \text{ and } Q_S(Y|X) = Q_T(Y|T(X)), \tag{5}$$

where $T_\sharp$ is the push-forward operator (see e.g., [31]). [6] propose estimating T through T_π in equation 4, which allows mapping samples from Q_S to Q_T.

Wasserstein Barycenters. When the ground-cost is a distance, OT defines a distance between distributions, $W_c(\hat{P}, \hat{Q}) = \langle \mathbf{C}, \pi^\star \rangle_F$, called Wasserstein distance. As such, OT defines barycenters of probability distributions [2]. Henceforth we denote the $K-$simplex as $\Delta_K = \{\mathbf{a} \in \mathbb{R}_+^K : \sum_k a_k = 1\}$.

Definition 1 *For distributions $\mathcal{P} = \{P_k\}_{k=1}^K$ and weights $\alpha \in \Delta_K$, the Wasserstein barycenter is a solution to,*

$$B^\star = \mathcal{B}(\alpha; \mathcal{P}) = \underset{B}{inf} \sum_{k=1}^{K} \alpha_k W_c(P_k, B). \tag{6}$$

Henceforth we call $\mathcal{B}(\cdot; \mathcal{P})$ barycentric operator. In this context, the Wasserstein hull of distributions $\mathcal{P}$ is,

$$\mathcal{M}(\mathcal{P}) = \{\mathcal{B}(\alpha; \mathcal{P}) : \alpha \in \Delta_K\} \tag{7}$$

When the distributions in $\mathcal{P}$ are empirical, solving equation 6 corresponds to estimating the support $\mathbf{X}^{(B)}$ of B. In this context, [8] proposed an algorithm known as *free-support Wasserstein barycenter* for calculating $\hat{B}$. Let $\mathbf{x}_i^{(B)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ be an initialization for the barycenter's support. One updates the support of $\hat{B}$ with,

$$\begin{aligned} \pi^{(k,it)} &= \text{OT}(\mathbf{X}^{(P_k)}, \mathbf{X}_{it}^{(B)}) \\ \mathbf{X}_{it+1}^{(B)} &\leftarrow \theta \mathbf{X}_{it}^{(B)} + (1-\theta) \sum_{k=1}^{K} \alpha_k T_{\pi^{(k,it)}}(\mathbf{X}_{it}^{(B)}) \end{aligned} \tag{8}$$

where θ is found at each iteration via line search. In the context of MSDA, [17] previously defined a barycenter of labeled distributions by penalizing transport plans $\pi^{(k,it)}$ that mix classes.

Mini-batch OT. For large scale datasets, computing OT is likely unfeasible, due its cubic complexity. A workaround, coming from ML, consists on using mini-batches [10]. For M batches of size n_b, this approach decreases the time complexity to $\mathcal{O}(Mn_b^3 \log n_b)$.

Remark on Notation. While $W_2(\hat{P}, \hat{Q})$ is defined between *empirical distributions*, in practice it is a function of $(\mathbf{X}^{(P)}, \mathbf{X}^{(Q)})$. With an abuse of notation, the mini-batch Wasserstein distance between given random samples of size n_b from P and Q is still noted as $W_2(\hat{P}, \hat{Q})$, with the support matrices restricted to a mini-batch.

3.3 Dictionary Learning

DiL is a representation learning technique that expresses a collection of vectors $\{\mathbf{x}_\ell\}_{\ell=1}^N$, $\mathbf{x}_\ell \in \mathbb{R}^d$ through a set of atoms $\mathcal{P} = \{\mathbf{p}_k\}_{k=1}^K$, $\mathbf{p}_k \in \mathbb{R}^d$ and weights $\mathcal{A} = \{\alpha_\ell\}_{\ell=1}^N$, $\alpha_\ell \in \mathbb{R}^K$. Mathematically,

$$\underset{\mathcal{P},\mathcal{A}}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(\mathbf{x}_\ell, \mathcal{P}^T\alpha_\ell) + \lambda_A \Omega_A(\mathcal{A}) + \lambda_P \Omega_P(\mathcal{P}),$$

where $\mathcal{L}$ is a suitable loss, Ω_A and Ω_P are regularizing terms on $\mathcal{A}$ and $\mathcal{P}$ respectively. In this sense, OT has previously contributed to DiL either by defining a meaningful loss function, or novel ways to aggregating atoms. For instance, [29] proposed using the Sinkhorn divergence of [7] as a loss function, while [32] proposed using Wasserstein barycenters for aggregating atoms. These works assume data in the form of histograms, i.e., $\mathbf{x}_\ell \in \Delta_d$. As consequence, $\mathbf{p}_k \in \Delta_d$ and $\alpha_\ell \in \Delta_K$.

4 Proposed Framework

In this section, we present our novel framework for MSDA, called Dataset Dictionary Learning (DaDiL). As our discussion relies on analogies with DiL theory, we provide in Table 1 a comparison of DiL concepts in different frameworks. In what follows, section 4.1 presents a novel algorithm for computing Wasserstein barycenters of labeled distributions, and section 4.2 presents our framework.

Table 1: Overview of analogies between different frameworks of DiL.

Concept	Symbol	Classic DiL	WDL [32]	DaDiL (ours)
Data	$\mathbf{x}_\ell$, or $\hat{Q}_\ell$	Vectors	Histograms	Point Clouds
Atom	$\mathcal{P}$	Vectors	Histograms	Point Clouds
Representation	$\mathcal{A}$	Vectors	Barycentric Coordinates	Barycentric Coordinates
Reconstruction	$\mathcal{B}$	Vectors	Histograms	Point Clouds

4.1 Wasserstein Barycenters of Labeled Distributions

We propose a novel algorithm for calculating differentiable Wasserstein barycenters of labeled empirical distributions. This algorithm is at the core of DaDiL (section 4), since we later represent datasets as barycenters of learned atoms.

In OT, there are at least 2 ways of integrating labels, either by penalizing OT plans that transport mass between different classes [6, 17], or by defining a metric in the label space [3]. We choose to integrate labels in the ground-cost,

$$C_{i,j} = \|\mathbf{x}_i^{(P)} - \mathbf{x}_j^{(Q)}\|_2^2 + \beta\|\mathbf{y}_i^{(P)} - \mathbf{y}_j^{(Q)}\|_2^2, \tag{9}$$

where $\mathbf{y}$ denotes labels one-hot encoding, and $\beta > 0$ controls the importance of label discrepancy. While simple, this choice allows us to motivate the barycentric projection of [6], and the label propagation of [26] as first-order optimality conditions of $W_c(\hat{P}, \hat{Q})$,

$$\begin{cases} \hat{\mathbf{x}}_i^{(P)} = T_\pi(\mathbf{x}_i^{(P)}) = n_P \sum_{j=1}^{n_Q} \pi_{i,j}\mathbf{x}_j^{(Q)}, \\ \hat{\mathbf{y}}_i^{(P)} = T_\pi(\mathbf{y}_i^{(P)}) = n_P \sum_{j=1}^{n_Q} \pi_{i,j}\mathbf{y}_j^{(Q)}. \end{cases} \tag{10}$$

Henceforth we denote $\pi = \text{OT}\Big((\mathbf{X}^{(P)}, \mathbf{Y}^{(P)}); (\mathbf{X}^{(Q)}, \mathbf{Y}^{(Q)})\Big)$. As a consequence, we can interpolate between two point clouds, since $\hat{\mathbf{y}}_i^{(P)}$ corresponds to a soft-label (i.e., probabilities). We use equations 9 and 10 for proposing a new barycenter strategy between labeled point clouds, shown in algorithm 1.

Algorithm 1 Free-Support Wasserstein Barycenter of Labeled Distributions

Require: $\{\mathbf{X}^{(P_k)}, \mathbf{Y}^{(P_k)}\}_{k=1}^K$, $\alpha \in \Delta_K$, $\tau > 0$, N_{itb}.
1: **for** $i = 1, \cdots, n_B$ **do**
2: $\quad \mathbf{x}_i^{(B)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, $y_i^{(B)} = \text{randint}(n_c)$
3: **end for**
4: **while** $|J_{it} - J_{it-1}| \geq \tau$ and $it \leq N_{itb}$ **do**
5: $\quad$ **for** $k = 1, \cdots K$ **do**
6: $\quad\quad \pi^{(k,it)} = \text{OT}\Big((\mathbf{X}^{(P_k)}, \mathbf{Y}^{(P_k)}); (\mathbf{X}_{it}^{(B)}, \mathbf{Y}_{it}^{(B)})\Big)$
7: $\quad$ **end for**
8: $\quad J_{it} = \sum_{k=1}^K \alpha_k \langle \pi^{(k,it)}, \mathbf{C}^{(k)} \rangle_F$
9: $\quad \mathbf{X}_{it+1}^{(B)} = \sum_{k=1}^K \alpha_k T_{\pi^{(k,it)}}(\mathbf{X}_{it}^{(B)})$
10: $\quad \mathbf{Y}_{it+1}^{(B)} = \sum_{k=1}^K \alpha_k T_{\pi^{(k,it)}}(\mathbf{Y}_{it}^{(B)})$
11: **end while**
Ensure: Labeled barycenter support $(\mathbf{X}^{(B)}, \mathbf{Y}^{(B)})$.

Differentiation. For calculating derivatives of $\mathbf{x}_i^{(B)}$ and $\mathbf{y}_i^{(B)}$ w.r.t. $\mathbf{x}_l^{(P_k)}$, $\mathbf{y}_l^{(P_k)}$, and α, we use the Envelope theorem of [1]. In other words, we do not propagate derivatives through the iterations of algorithm 1. We provide further details in our appendix.

Computational Complexity. Let $\hat{P}_k$ have n points in its support, for $k = 1, \cdots, K$. The complexity of algorithm 1 is dominated by line 6, which has complexity $\mathcal{O}(n^3 \log n)$. Hence, the overall computational complexity is $\mathcal{O}(N_{itb} K n^3 \log n)$.

4.2 Dataset Dictionary Learning for MSDA

In this section, we introduce our novel framework, called DaDiL, and explore how to use it for MSDA. Let $\mathcal{Q} = \{\hat{Q}_{S_\ell}\}_{\ell=1}^{N_S} \cup \{\hat{Q}_T\}$ correspond to N_S labeled sources and an unlabeled target. Let $\mathcal{A} = [\alpha_1, \cdots, \alpha_{N_S}, \alpha_{N_S+1}]$, and $\mathcal{P} = \{\hat{P}_k\}_{k=1}^K$. The $\hat{P}_k$'s are an empirical approximation of the point clouds that interpolate distributional shift. Following our notation, $\alpha_T := \alpha_{N_S+1}$. For $N = N_S + 1$, DaDiL consists on minimizing,

$$(\mathcal{P}^\star, \mathcal{A}^\star) = \underset{\mathcal{P},\mathcal{A}\in(\Delta_K)^N}{\operatorname{argmin}} \frac{1}{N}\sum_{\ell=1}^{N} \mathcal{L}(\hat{Q}_\ell, \mathcal{B}(\alpha_\ell; \mathcal{P})), \tag{11}$$

where $\mathcal{L}$ is a loss between distributions. Since the target domain is not labeled, we define,

$$\mathcal{L}(\hat{Q}_\ell, \hat{B}_\ell) = \begin{cases} W_c(\hat{Q}_\ell, \hat{B}_\ell), & \text{if } \hat{Q}_\ell \text{ is labeled,} \\ W_2(\hat{Q}_\ell, \hat{B}_\ell), & \text{otherwise,} \end{cases}$$

i.e., when no labels in $\hat{Q}_\ell$ are available, we minimize the standard 2-Wasserstein distance. Optimizing 11 over entire datasets might be

intractable due the complexity of OT. We thus employ mini-batch OT [10]. In addition, we need to enforce the constraints $\mathbf{y}_l^{(P_k)} \in \Delta_{n_c}$ and $\alpha_\ell \in \Delta_K$. In the first case we do a change of variables, and optimize the logits $\mathbf{p} \in \mathbb{R}^{n_c}$ s.t. $\mathbf{y} = \text{softmax}(\mathbf{p})$. In the second case, we project α_ℓ into the simplex orthogonally,

$$\text{proj}_{\Delta_K}(\alpha_\ell) = \underset{\alpha \in \Delta_K}{\text{argmin}} \|\alpha - \alpha_\ell\|_2.$$

The overall optimization algorithm is shown in algorithm 2.

Algorithm 2 DaDiL learning loop.

Require: $\mathcal{Q} = \{\hat{Q}_\ell\}_{\ell=1}^N$, number of iterations N_{iter}, of atoms K, of batches M, batch size n_b, learning rate η.

1: Initialize $\mathbf{x}_j^{(P_k)} \sim \mathcal{N}(0, \mathbf{I}_d)$, $\mathbf{a}_\ell \sim \mathcal{N}(0, \mathbf{I}_K)$.
2: **for** $it = 1 \cdots, N_{iter}$ **do**
3: **for** $batch = 1, \cdots, M$ **do**
4: **for** $\ell = 1, \cdots, (N_S + 1)$ **do**
5: Sample $\{\mathbf{x}_1^{(Q_\ell)}, \cdots, \mathbf{x}_{n_b}^{(Q_\ell)}\}$.
6: **if** $\hat{Q}_\ell$ is labeled **then**
7: Sample $\{\mathbf{y}_1^{(Q_\ell)}, \cdots, \mathbf{y}_{n_b}^{(Q_\ell)}\}$.
8: **end if**
9: **for** $k = 1, \cdots, K$ **do**
10: sample $\{(\mathbf{x}_1^{(P_k)}, \mathbf{p}_1^{(P_k)}), \cdots, (\mathbf{x}_{n_b}^{(P_k)}, \mathbf{p}_{n_b}^{(P_k)})\}$,
11: change variables $\mathbf{y}_j^{(P_k)} = \text{softmax}(\mathbf{p}_j^{(P_k)})$
12: **end for**
13: calculate $\mathbf{X}^{(B_\ell)}, \mathbf{Y}^{(B_\ell)} = \mathcal{B}(\alpha_\ell; \mathcal{P})$
14: **end for**
15: $L = (1/N) \sum_{\ell=1}^N \mathcal{L}(\hat{Q}_\ell, \hat{B}_\ell)$
16: $\mathbf{x}_j^{(P_k)} \leftarrow \mathbf{x}_j^{(P_k)} - \eta \partial L / \partial \mathbf{x}_j^{(P_k)}$
17: $\mathbf{p}_j^{(P_k)} \leftarrow \mathbf{p}_j^{(P_k)} - \eta \partial L / \partial \mathbf{p}_j^{(P_k)}$
18: $\alpha_\ell \leftarrow \text{proj}_{\Delta_K}(\alpha_\ell - \eta \partial L / \partial \alpha_\ell)$.
19: **end for**
20: **end for**

Ensure: Dictionary $\mathcal{P}^\star$ and weights $\mathcal{A}^\star$.

Intuition. We learn how to express each distribution $\hat{Q}_\ell \in \mathcal{Q}$ as a barycenter of free distributions $\mathcal{P} = \{\hat{P}_k\}_{k=1}^K$, parametrized by their support i.e., $(\mathbf{X}^{(P_k)}, \mathbf{Y}^{(P_k)})$. In other words, we learn $\mathcal{P}$ s.t. $\mathcal{Q}$ is contained in the *Wasserstein hull* of atoms, $\mathcal{M}(\mathcal{P})$.

Implementation. We implement algorithms 1 and 2 using Pytorch [22] and Python Optimal Transport (POT) [12], for automatic differentiation and OT details respectively. As previous works [17, 35], DaDiL is applied to the latent space of an encoder, pre-trained on source domain data, as shown in figure 1.

Computational Complexity. In algorithm 2, the complexity of line 13 dominates over other lines. As we discussed in section 4.1, the complexity of calculating $\mathcal{B}(\alpha_\ell; \mathcal{P})$ depends on the size of distributions support. Since we do computations using mini-batches, this corresponds to $\mathcal{O}(N_{itb} n_b^3 \log n_b)$. This is repeated for $N_{iter} \times M \times (N_S + 1)$, which implies a complexity of $\mathcal{O}(N_{iter} M N_S N_{itb} n_b^3 \log n_b)$.

Multi-Source Domain Adaptation. We recast the hypothesis in eq. 5 for MSDA. We assume the existence of $K > 1$ unknown distributions, $P_1, \cdots, P_K$ for which Q_ℓ can be approximated as their interpolation in Wasserstein space, i.e. $Q_\ell = T_\sharp B_\ell$, and $Q_\ell(Y|X) = B_\ell(Y|T(X))$, for $B_\ell = \mathcal{B}(\alpha_\ell; \mathcal{P})$ and a possibly non-linear transformation T. If $W_c(Q_\ell, B_\ell) \approx 0$ we can assume $T(\mathbf{x}) = \mathbf{x}$.

We start by learning $(\mathcal{P}, \mathcal{A})$, as illustrated in figure 2. Then, we propose 2 ways of using our dictionary for MSDA. Our first strategy, called DaDiL-R, consists on computing $\hat{B}_T = \mathcal{B}(\alpha_T; \mathcal{P})$, i.e.,

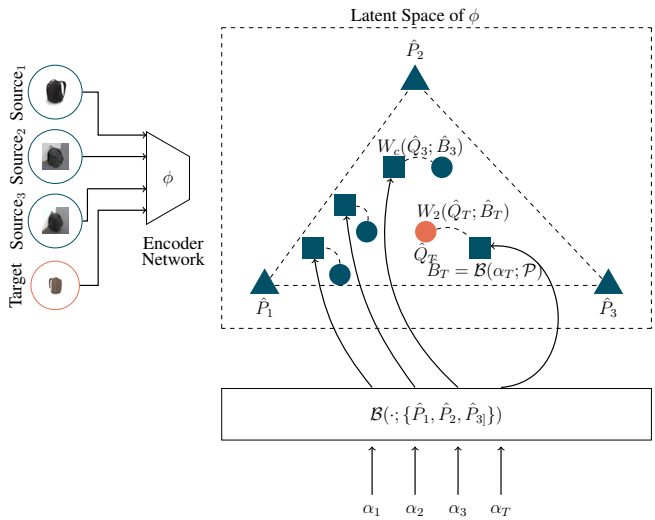

Figure 1: Conceptual illustration of DaDiL. Each domain is denoted by a blue or orange circle, corresponding to whether it is labeled or not. DaDiL *reconstructs* domains as Wasserstein barycenters, denoted by squares, of atoms, denoted by triangles. The target domain (orange circle) is unlabeled, but we are able to represent it through a labeled distribution through a Wasserstein barycenter.

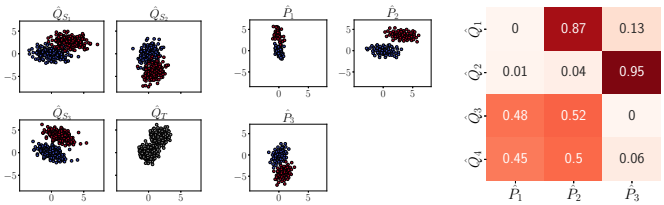

Figure 2: From left to right: set of datasets $\mathcal{Q} = \{\hat{Q}_{S_1}, \cdots, \hat{Q}_{S_3}, \hat{Q}_T\}$, where $\hat{Q}_T$ is the unlabeled target domain; atoms $\mathcal{P} = \{\hat{P}_1, \hat{P}_2, \hat{P}_3\}$; barycentric weights $\mathcal{A}$.

the distribution in $\mathcal{M}(\mathcal{P})$ closest to $\hat{Q}_T$. Since each $\hat{P}_k$ has a labeled support, algorithm 1 yields matrices $\mathbf{X}^{(B_T)}$ and $\mathbf{Y}^{(B_T)}$ corresponding to the support of $\hat{B}_T$. Then,

$$\hat{h}_R = \underset{h \in \mathcal{H}}{\text{argmin}}\, \hat{\mathcal{R}}_{B_T}(h) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}(h(\mathbf{x}_i^{(B_T)}), y_i^{(B_T)})$$

We theoretically justify it using Theorem 2 of [25],

Theorem 1 *(Due to [25]) Let $\mathbf{X}^{(P)} \in \mathbb{R}^{n_P \times d}$ and $\mathbf{X}^{(Q)} \in \mathbb{R}^{n_Q \times d}$ be i.i.d. samples from P and Q. Then, for any $d' > d$ and $\xi' < \sqrt{2}$ there exists some constant n_0 depending on d' s.t. for $\delta \in (0, 1)$ and $min(n_P, n_Q) \geq n_0 max(\delta^{-(d+2)}, 1)$ with probability at least $1 - \delta$ for all h,*

$$\mathcal{R}_Q(h) \leq \mathcal{R}_P(h) + W_2(\hat{P}, \hat{Q}) + \zeta + \lambda,$$

where,

$$\zeta = \sqrt{2^{(\log 1/\delta)/\xi'}} \left(\sqrt{1/n_P} + \sqrt{1/n_Q} \right),$$

$$\lambda = min_{h \in \mathcal{H}} \mathcal{R}_Q(h) + \mathcal{R}_P(h).$$

Additional discussion on this result is provided in our appendix. We apply this result for the residual shift $W_2(\hat{Q}_T, \hat{B}_T)$,

$$\mathcal{R}_{Q_T}(h) \leq \mathcal{R}_{B_T}(h) + W_2(\hat{Q}_T, \hat{B}_T) + \zeta + \lambda. \quad (12)$$

As discussed in [25], 3 factors play a role in the success of DA, namely, $W_2(\hat{P}, \hat{Q})$, $\mathcal{R}_{B_T}(h)$, and λ. The first term is the reconstruction error, and is directly minimized in algorithm 2. The second term is the risk of h in B_T, which is minimized when learning the classifier $\hat{h}_R = \text{argmin}\, \hat{\mathcal{R}}_{B_T}(h)$. This term depends on the separability of classes in $\hat{B}_T$, which is enforced by considering labels in the ground-cost (eqn. 9). The last term is the joint risk λ of a classifier learned with data from Q_T and B_T. This term is difficult to bound, as no labels in $\hat{Q}_T$ are available, but, under the hypothesis $Q_T(Y|X) = B_T(Y|T(X))$, this term is low. This was similarly assumed by [6, 25]. DaDiL-R is illustrated in figure 3.

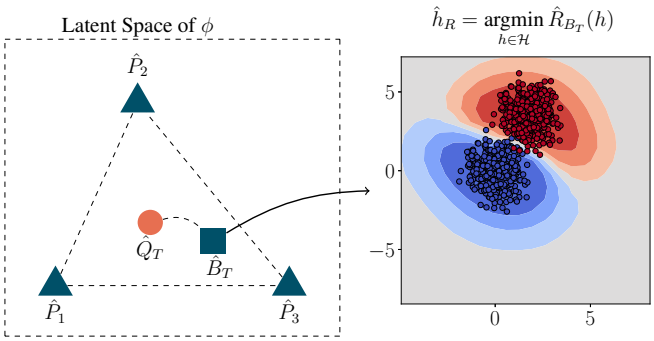

Figure 3: Conceptual outline of DaDiL-Reconstruction. Labeled samples in the target domain are acquired through a Wasserstein barycenter $\hat{B}_T = \mathcal{B}(\alpha_T; \mathcal{P})$, which is close to $\hat{Q}_T$ in Wasserstein sense.

Our second strategy, called DaDiL-E, is based on ensembling. Since each of our atoms is labeled, i.e., each $\mathbf{x}_i^{(P_k)}$ has an associated $\mathbf{y}_i^{(P_k)}$, we may learn a set of K classifiers, $\hat{h}_k = \text{argmin}_{h \in \mathcal{H}} \hat{\mathcal{R}}_{P_k}(h)$, one for each atom. Naturally, one may use $\alpha \in \Delta_K$ for weighting predictions of atom classifiers. We weight the $\hat{h}_k$'s using α_T, which is theoretically justified in theorem 2,

$$\hat{h}_E(\mathbf{x}_j^{(Q_T)}) = \sum_{k=1}^{K} \alpha_{T,k} \hat{h}_k(\mathbf{x}_j^{(Q_T)}),$$

Theorem 2 *Let* $\{\mathbf{X}^{(P_k)}\}_{k=1}^K$, $\mathbf{X}^{(P_k)} \in \mathbb{R}^{n_k \times d}$ *and* $\mathbf{X}^{(Q_T)} \in \mathbb{R}^{n_T \times d}$ *be i.i.d. samples from* P_k *and* Q_T. *Let* $\hat{h}_k$ *be the minimizer of* $\mathcal{R}_{P_k}$ *and* $\mathcal{R}_\alpha(h) = \sum_{k=1}^K \alpha_k \mathcal{R}_{P_k}(h)$. *Under the same conditions of theorem 1, and for* $\delta \in (0, 1)$, *with probability at least* $1 - \delta$, *the following holds,*

$$\mathcal{R}_{Q_T}(\hat{h}_\alpha) \leq \mathcal{R}_\alpha(\hat{h}_\alpha) + W_2(\mathcal{B}(\alpha; \mathcal{P}), \hat{Q}_T) + \gamma + \lambda + \zeta,$$

$$\gamma = \sum_{k=1}^{K} \alpha_k W_2(\hat{P}_k, \mathcal{B}(\alpha; \mathcal{P})),$$

$$\zeta = \sum_{k=1}^{K} \alpha_k \sqrt{2 \log 1/\delta / \xi'} \left(\sqrt{1/n_k} + \sqrt{1/n_T} \right),$$

$$\lambda = \sum_{k=1}^{K} \alpha_k \left(min_{h \in \mathcal{H}} \mathcal{R}_{P_k}(h) + \mathcal{R}_{Q_T}(h) \right).$$

We provide the proof of this result and additional discussion in our appendix. This bound depends on different terms. First, γ is, for a given α, minimal, as $\mathcal{B}(\alpha; \mathcal{P})$ is the minimizer of $\hat{B} \mapsto \sum_k \alpha_k W_2(\hat{P}_k, \hat{B})$. λ corresponds to the complexity of domain adaptation, and in general cannot be directly controlled due the unavailability of labels in $\hat{Q}_T$. Finally, ξ corresponds to the sample complexity of estimating $W_2(P_k, Q_T)$ via finite samples. Note that α_T minimizes the terms in the r.h.s., as, by design, it minimizes the term $\alpha \mapsto W_2(\mathcal{B}(\alpha; \mathcal{P}), \hat{Q}_T)$. DaDiL-E is illustrated in figure 4.

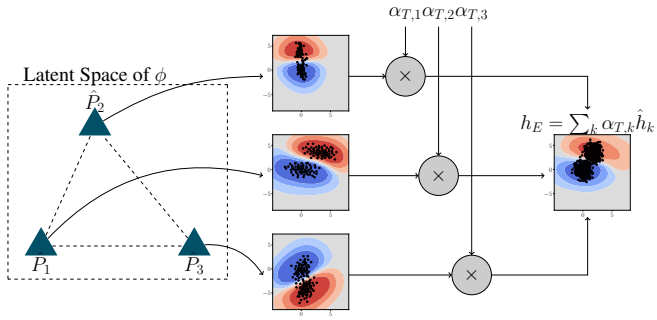

Figure 4: Conceptual outline of our DaDiL-Ensembling technique, where a classifier is defined on target domain data by ensembling classifiers learned on atom elements.

5 Experiments

5.1 Multi-Source Domain Adaptation

Experimental Setup. All experiments were run on a computer with a Ubuntu 22.04 OS, a 12th Gen Intel(R) Core™ i9-12900H CPU with 64 GB of RAM, and with a NVIDIA RTX A100 GPU with 4GB of VRAM. We explore the following hyper-parameters,

- Number of samples n is searched among $\{50, 100, 200\} \times n_c$.
- Number of atoms K is searched among $\{3, 4, \cdots, 8\}$.
- Batch size n_b is searched among $\{5, 10, 20\} \times n_c$. We further sample balanced batches from the sources.

The complexity of our model is controlled through n and K. We provide further analysis on the robustness w.r.t. hyper-parameter choice, as well as the full set of chosen hyper-parameters in our appendix. For other algorithms from the State-of-the-Art (SOTA), we use the best hyper-parameter settings reported by their respective authors.

Caltech-Office 10 is a benchmark consisting on the intersection of the Caltech 256 dataset of [14] and the Office 31 dataset of [30]. It has 4 domains: Amazon (A), dSLR (D), Webcam (W) and Caltech (C). In this benchmark we compare DaDiL with other shallow DA algorithms, such as: (i) Subspace Alignment (SA) of [11]; (ii) Transfer Component Analysis (TCA) of [20]; (iii) Optimal Transport Domain Adaptation (OTDA) of [6]; (iv) Wasserstein Barycenter Transport (WBT) of [17, 18]; (v) Weighted JDOT (WJDOT) of [35]. (i) and (ii) are standard algorithms in DA, (iii) is the single-source OT baseline, and (iv, v) are the SOTA for shallow MSDA. The baseline corresponds to training a single-layer Perceptron with the concatenation of source domain data.

Our results is presented in table 2. DaDiL improve over previous OT-based MSDA baselines, i.e. WJDOT and WBT, being especially better on the Webcam and Caltech domains. Overall, we improve previous SOTA by 3.15 in terms of average DA performance.

Ablation Study. We investigate the effectiveness of DiL in comparison with other barycenter-based approaches. As follows, we compare the performance on the Caltech-Office 10 benchmark of 4 methods: (i) Wasserstein Barycenter (WB); (ii) WBT; (iii) Wasserstein Barycentric Coordinates Regression (WBR)-R and WBR-E, which can be understood as the adaptation of the framework of [4] for point clouds. The R and E methods are analogous to DaDiL when the

Table 2: Classification accuracy (in %) of DA methods. Each column represents a target domain for which we report mean ± standard deviation over 5 folds. * and † denote results from [17] and [35].

Method	A	D	W	C	Avg
Baseline	90.55 ± 1.36	96.83 ± 1.33	88.36 ± 1.33	82.95 ± 1.26	89.67
SA	88.61 ± 1.72	92.08 ± 3.82	79.33 ± 3.67	73.00 ± 2.31	83.26
TCA*	86.83 ± 4.71	89.32 ± 1.33	97.51 ± 1.18	80.79 ± 2.65	88.61
OTDA	88.26 ± 1.36	90.41 ± 3.86	88.09 ± 3.80	83.02 ± 1.67	87.44
WJDOT†	**94.23 ± 0.90**	**100.00 ± 0.00**	89.33 ± 2.91	85.93 ± 2.07	92.37
$\text{WBT}^{\star}_{reg}$	92.74 ± 0.45	95.87 ± 1.43	96.57 ± 1.76	85.01 ± 0.84	92.55
DaDiL-R	94.06 ± 1.82	98.75 ± 1.71	98.98 ± 1.51	88.97 ± 1.06	95.19
DaDiL-E	94.16 ± 1.58	**100.00 ± 0.00**	**99.32 ± 0.93**	**89.15 ± 1.68**	**95.66**

atoms are initialized and fixed as the source domains. We provide further details of this adaptation in our appendix.

Table 3: Classification accuracy (in %) of DA methods. $\mathcal{P}$ and $\mathcal{A}$ indicate learning atom distributions and barycentric coefficients respectively. T indicates an additional transport step towards Q_T.

Method	$\mathcal{P}$	$\mathcal{A}$	T	A	D	W	C	Avg.
WB				88.54 ± 1.16	90.62 ± 8.38	93.89 ± 3.30	83.73 ± 1.49	89.19
WBT_{reg}			✔	92.74 ± 0.45	95.87 ± 1.43	96.57 ± 1.76	85.01 ± 0.84	92.55
WBR-R		✔		91.35 ± 1.19	91.87 ± 9.47	81.69 ± 3.26	86.31 ± 1.73	86.09
WBR-E		✔		91.97 ± 2.40	91.87 ± 2.79	83.73 ± 2.57	86.13 ± 1.84	88.42
DaDiL-R	✔	✔		94.06 ± 1.82	98.75 ± 1.71	98.98 ± 1.51	88.97 ± 1.06	95.19
DaDiL-E	✔	✔		**94.16 ± 1.58**	**100.00 ± 0.00**	**99.32 ± 0.93**	**89.15 ± 1.68**	**95.66**

We report our findings in table 3. Overall, WB and WBR have sub-optimal performance. On the one hand, this implies that $\hat{Q}_T \notin \mathcal{M}(\mathcal{Q}_S)$. On the other hand, this implies that DiL is key for MSDA. Indeed, since $\hat{P}_k \in \mathcal{P}$ are free, DaDiL learns $\mathcal{P}$ s.t. $\hat{Q}_T \in \mathcal{M}(\mathcal{P})$. WBT_{reg} compensates this fact by transporting the $\hat{B}$ towards $\hat{Q}_T$, thus minimizing the *residual shift* $W_2(\hat{B}, \hat{P}_T)$.

Refurbished Office 31. In this experiment, we use the Office 31 benchmark of [30], with the improvements proposed by [28]. This benchmark has 3 domains: Amazon (A), dSLR (D) and Webcam (W). Our goal is to establish a comparison with deep DA methods. As follows, we consider: (i) Domain Adversarial Neural Network (DANN) of [13], (ii) Wasserstein Distance Guided Representation Learning (WDGRL) of [33], (iii) Deep-JDOT of [9], (iv) Moment Matching for MSDA (M3SDA) of [23], (v) WJDOT and (vi) WBT. While (i) - (iii) are single source baselines, (iv) is a standard method for MSDA. We use a ResNet-50 [15] as backbone.

Table 4: Classification accuracy (in %) of DA methods on the Office 31 benchmark. Each column represents a target domain for which we report mean ± standard deviation over 5 folds.

Method	A	D	W	Avg
Baseline	70.57	97.00	95.47	87.68
DANN	78.19	97.00	93.08	89.42
WDGRL	76.06	97.00	93.71	88.92
DeepJDOT	80.85	94.00	93.38	89.61
M3SDA	64.89	**98.00**	96.85	86.58
WBT_{reg}	77.48	96.00	95.59	89.69
WJDOT	70.21	97.00	94.96	87.39
DaDiL-R	**85.46**	93.00	**97.48**	**91.98**
DaDiL-E	83.51	94.00	94.34	90.61

A summary of our results is shown in table 4. Overall, DaDiL-R and E are especially better than previous algorithms in the Amazon domain. As a consequence, in terms of average domain performance, DaDiL-R and E improve over the second-best method (WBT_{reg}) by a margin of 2.29% and 1.37% respectively.

CWRU. In this benchmark, we explore DaDiL for cross-domain fault diagnosis. The goal is to classify which type of fault has occurred, based on sensor readings. Hence, we extract 2048 Fourier coefficients from a sub-set of 4096 time-steps extracted from the raw signals (see [38], or our appendix for more details). As feature extractor, we use a 3-layer fully connected encoder[1]. We compare 3 single, and 5 multi-source DA algorithms to DaDiL, namely, DANN, OTDA, TCA, M3SDA, LTC-MSDA of [37], JCPOT of [26], WBT_{reg} and WJDOT.

Table 5: Classification accuracy (in %) of DA methods on the CWRU benchmark. Each column represents a target domain for which we report mean ± standard deviation over 5 folds.

Method	1772rpm	1750rpm	1730rpm	Avg
Baseline	70.90 ± 0.40	79.76 ± 0.11	72.26 ± 0.23	74.31
DANN	67.96 ± 8.52	64.38 ± 5.03	57.75 ± 17.06	63.37
OTDA	70.48 ± 2.25	79.61 ± 0.25	74.98 ± 1.26	75.02
TCA	87.17 ± 4.25	84.11 ± 4.77	92.74 ± 4.12	88.01
M3SDA	56.86 ± 7.31	69.81 ± 0.36	61.06 ± 6.35	62.57
WJDOT	65.01 ± 0.27	69.81 ± 0.07	57.40 ± 1.18	64.07
M3SDA_{β}	60.15 ± 8.38	70.00 ± 0.00	64.00 ± 5.47	64.72
LTC-MSDA	82.21 ± 8.03	75.33 ± 5.91	81.04 ± 5.45	79.52
JCPOT	77.48 ± 0.86	96.00 ± 0.10	95.59 ± 0.56	91.74
WBT_{reg}	99.28 ± 0.18	79.91 ± 0.04	97.71 ± 0.76	92.30
DaDiL-R	**99.86 ± 0.21**	**99.85 ± 0.08**	**100.00 ± 0.00**	**99.90**
DaDiL-E	93.71 ± 6.50	83.63 ± 4.98	99.97 ± 0.05	92.33

We present a summary of our results in table 5. Overall, WBT_{reg} and DaDiL are the best performing methods, demonstrating the power of Wasserstein barycenters for DA. Our method outperforms WBT_{reg} by 7.71%, in terms of average domain performance. Furthermore, our methods surpass other deep learning baselines, such as M3SDA [23] and LTC-MSDA [37], by a margin of 19.90%.

5.2 *Domain Adaptation using Atom Interpolations*

Besides performing MSDA with optimal barycentric coordinates $\alpha_T \in \Delta_K$, in this section we explore the question *how well do* $\alpha \in \Delta_K$ *perform?* We explore these questions in terms of Wasserstein distance $W_2(\mathcal{B}(\alpha; \mathcal{P}), \hat{Q}_T)$, and classification accuracy of using α in DaDiL-R and E, as shown in figure 5.

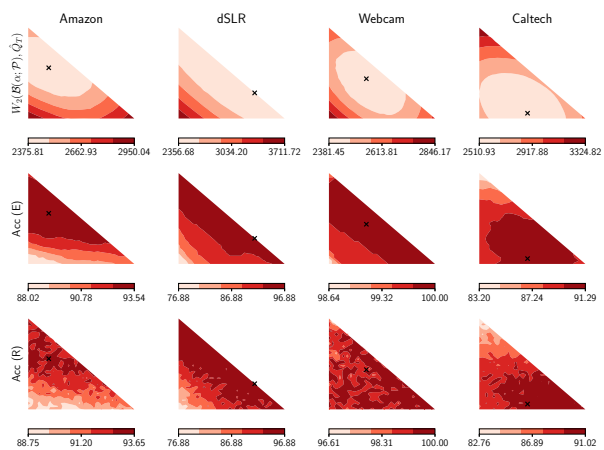

Figure 5: Analysis of DA on Caltech-Office with interpolations of dictionary atoms. The black cross represents the α found by DaDiL.

[1] i.e., 2048 → 1024 → 512 → 256 with ReLU activations.

In figure 5 we construct an uniform grid over Δ_3. For each α in such grid, we reconstruct $\mathcal{B}(\alpha; \mathcal{P})$, then we evaluate: (i) the *reconstruction loss* $W_2(\mathcal{B}(\alpha; \mathcal{P}), \hat{Q}_T)$; (ii) the classification accuracy of DaDiL-E, with α, on $\hat{Q}_T$; (iii) the classification accuracy of DaDiL-R with α, on $\hat{Q}_T$. These correspond to the 3 rows in figure 5. As shown, the weights found by DaDiL are optimal w.r.t. other choices $\alpha \in \Delta_3$. Nonetheless, a wide region of the simplex yield *equally good* reconstructions, either w.r.t. reconstruction loss, or w.r.t. DA performance. We conclude that DaDiL is able to learn distribution whose interpolations generalize well to the target domain.

Furthermore, in figure 6 we analyze the correlation between DA performance and reconstruction loss, for $\alpha \in \Delta_3$. Our analysis shows that these 2 terms are negatively correlated, for both DaDiL-R and E. Indeed, based on our theoretical analysis (theorems 1 and 2), classification risk is bounded by the reconstruction loss. Since DA performance is inversely proportional to the classifier risk in a given domain, our analysis agrees with both theorems.

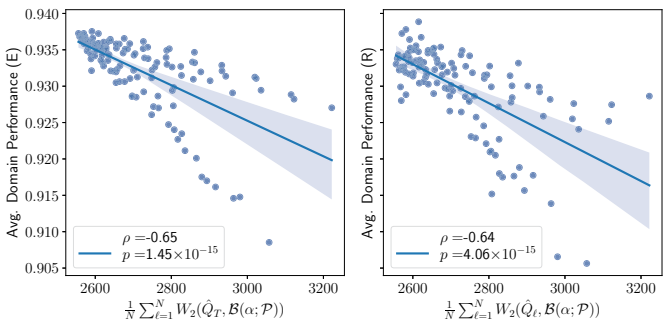

Figure 6: Correlation between DiL loss and the performance of DaDiL-R and E.

Finally, we analyze the performance of DaDiL-R and E for α taken uniformly from Δ_K, for $K \in \{3, \cdots, 8\}$. We report our findings in figure 7, and compare the performance w.r.t. DaDiL performance in table 2, for $\alpha := \alpha_T$. As shown in Figure 7 α_T is above average for most domains and number of atoms K.

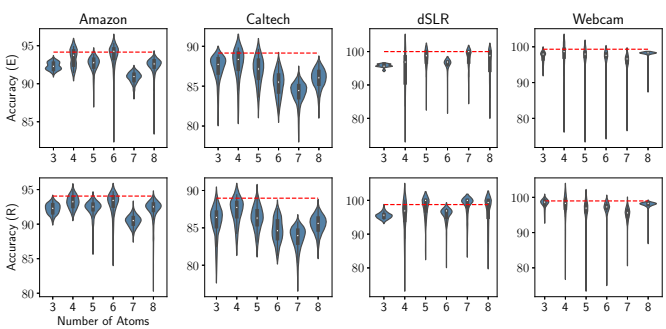

Figure 7: Performance analysis of latent space interpolations on the Caltech-Office 10 benchmark. The red dotted line corresponds to the results reported in Table 2 for DaDiL.

Overall, figures 5, 6 and 7 show that DaDiL learns an optimal set of barycentric coordinates for the target domain. Nonetheless, interpolations in the Wasserstein hull $\mathcal{M}(\mathcal{P})$ of atom distributions can be equally interesting for MSDA. These remarks indicate that DaDiL is able to (i) learn common discriminant information about the source domains; (ii) interpolate the distributional shift between the various distributions in $\mathcal{Q} = \{\hat{Q}_{S_\ell}\}_{\ell=1}^{N_S} \cup \{\hat{Q}_T\}$ through the atoms $\mathcal{P}$.

6 Discussion

Benefits of Dictionary Learning. Our proposed framework allows for the learning of new, *virtual distributions*, which can then be used to reconstruct distributions seen during DiL by generating new samples. As such, our algorithm is able to improve over past SOTA, and, as shown in section 5.2, we are able to generate new domains by interpolating the atom distributions in Wasserstein space. Especially, we improve previous SOTA and barycenter-based algorithms by 3.15% in the Caltech-Office 10 benchmark.

Benefits of Wasserstein Barycenters. In our experiments, we established a comparison between DaDiL, WBT [17] and WJDOT [35]. The first two methods rely on Wasserstein barycenters for reconstructing the target domain, while WJDOT aggregates the source domains linearly. Overall we show that Wasserstein barycenters are an important component of MSDA, as they allow to *average probability distributions non-linearly*. On the other hand, the linear average of distributions can be understood as importance weighting on samples. Under the covariate shift hypothesis, re-weighting samples is enough, but under more complicated shifts (i.e. non-linear data transformations), Wasserstein barycenters are more flexible.

Shallow vs. Deep Domain Adaptation. As remarked by [35], the assumption of having a meaningful feature extractor ϕ *before performing DA* is realistic, as in modern practice pre-trained models are widely available. It is noteworthy that a fine-tuning step with source-domain data may be necessary in order to achieve better performance. In addition, doing so allows for the comparison with deep DA methods. In this context, we remark that our method improves over previous deep DA SOTA in the context of the Refurbished Office 31 and CWRU benchmarks. Overall, shallow DA is computationally simpler than deep DA, as one needs to learn a smaller set of parameters (i.e., the classification layer).

7 Conclusion

In this work, we tackle the problem of MSDA through OT-based DiL of probability distributions. We view elements in DiL as empirical distributions. As such we learn a dictionary that is able to interpolate the distributional shift of distributions in DiL. We make 2 methodological contributions to MSDA, through methods called DaDiL-R, based on the reconstruction of labeled samples in the target domain, and DaDiL-E, based on ensembling of atom classifiers. Our methods are theoretically grounded on previous theorems from the literature [25, Theorem 2] and a novel result (theorem 2).

Our proposed methods are compared to 11 methods from the SOTA in MSDA in 3 benchmarks, namely, Caltech-Office 10 [30, 14], Refurbished Office 31 [30, 28] and CWRU. We improve previous performance by 3.15%, 2.29% and 7.71% respectively. Moreover, we show that general interpolations in the Wasserstein hull of our learned dictionary can be equally interesting for MSDA.

Our framework opens an interesting line of research, for *learning* empirical distributions, generating synthetic through Wasserstein barycenters and interpolating distributional shift in Wasserstein space. It is flexible so as to accommodate other notions of barycenters of distributions, and loss functions between reconstructions and real datasets. In practical terms, future works will focus on parametric formulations of DaDiL. In theoretical terms, we seek to understand the statistical challenges posed by DaDiL.

References

[1] SN Afriat, 'Theory of maxima and the method of lagrange', *SIAM Journal on Applied Mathematics*, **20**(3), 343–357, (1971).

[2] Martial Agueh and Guillaume Carlier, 'Barycenters in the wasserstein space', *SIAM Journal on Mathematical Analysis*, **43**(2), 904–924, (2011).

[3] David Alvarez-Melis and Nicolo Fusi, 'Geometric dataset distances via optimal transport', *Advances in Neural Information Processing Systems*, **33**, 21428–21439, (2020).

[4] Nicolas Bonneel, Gabriel Peyré, and Marco Cuturi, 'Wasserstein barycentric coordinates: histogram regression using optimal transport.', *ACM Trans. Graph.*, **35**(4), 71–1, (2016).

[5] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy, 'Joint distribution optimal transportation for domain adaptation', *Advances in Neural Information Processing Systems*, **30**, (2017).

[6] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy, 'Optimal transport for domain adaptation', *IEEE transactions on pattern analysis and machine intelligence*, **39**(9), 1853–1865, (2016).

[7] Marco Cuturi, 'Sinkhorn distances: Lightspeed computation of optimal transport', *Advances in neural information processing systems*, **26**, (2013).

[8] Marco Cuturi and Arnaud Doucet, 'Fast computation of wasserstein barycenters', in *International conference on machine learning*, pp. 685–693. PMLR, (2014).

[9] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty, 'Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation', in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 447–463, (2018).

[10] Kilian Fatras, Younes Zine, Szymon Majewski, Rémi Flamary, Rémi Gribonval, and Nicolas Courty, 'Minibatch optimal transport distances; analysis and applications', *arXiv preprint arXiv:2101.01792*, (2021).

[11] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars, 'Unsupervised visual domain adaptation using subspace alignment', in *Proceedings of the IEEE international conference on computer vision*, pp. 2960–2967, (2013).

[12] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al., 'Pot: Python optimal transport.', *J. Mach. Learn. Res.*, **22**(78), 1–8, (2021).

[13] 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*, **17**(1), 2096–2030, (2016).

[14] Gregory Griffin, Alex Holub, and Pietro Perona, 'Caltech-256 object category dataset', Technical report, California Institute of Technology, (2007).

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, (2016).

[16] De-An Huang and Yu-Chiang Frank Wang, 'Coupled dictionary and feature space learning with applications to cross-domain image synthesis and recognition', in *Proceedings of the IEEE international conference on computer vision*, pp. 2496–2503, (2013).

[17] Eduardo Fernandes Montesuma and Fred Maurice Ngole Mboula, 'Wasserstein barycenter for multi-source domain adaptation', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16785–16793, (June 2021).

[18] Eduardo Fernandes Montesuma and Fred Maurice Ngole Mboula, 'Wasserstein barycenter transport for acoustic adaptation', in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3405–3409, (May 2021).

[19] Eduardo Fernandes Montesuma, Fred Ngole Mboula, and Antoine Souloumiac, 'Recent advances in optimal transport for machine learning', *arXiv preprint arXiv:2306.16156*, (2023).

[20] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang, 'Domain adaptation via transfer component analysis', *IEEE transactions on neural networks*, **22**(2), 199–210, (2010).

[21] Sinno Jialin Pan and Qiang Yang, 'A survey on transfer learning', *IEEE Transactions on knowledge and data engineering*, **22**(10), 1345–1359, (2009).

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, 'Pytorch: An imperative style, high-performance deep learning library', in *Advances in Neural Information Processing Systems 32*, eds., H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, 8024–8035, Curran Associates, Inc., (2019).

[23] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang, 'Moment matching for multi-source domain adaptation', in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, (2019).

[24] Gabriel Peyré, Marco Cuturi, et al., 'Computational optimal transport: With applications to data science', *Foundations and Trends® in Machine Learning*, **11**(5-6), 355–607, (2019).

[25] Ievgen Redko, Amaury Habrard, and Marc Sebban, 'Theoretical analysis of domain adaptation with optimal transport', in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 737–753. Springer, (2017).

[26] Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younes Bennani, *Advances in domain adaptation theory*, Elsevier, 2019.

[27] Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani, 'A survey on domain adaptation theory: learning bounds and theoretical guarantees', *arXiv preprint arXiv:2004.11829*, (2020).

[28] Tobias Ringwald and Rainer Stiefelhagen, 'Adaptiope: A modern benchmark for unsupervised domain adaptation', in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 101–110, (2021).

[29] Antoine Rolet, Marco Cuturi, and Gabriel Peyré, 'Fast dictionary learning with a smoothed wasserstein loss', in *Artificial Intelligence and Statistics*, pp. 630–638. PMLR, (2016).

[30] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell, 'Adapting visual category models to new domains', in *European conference on computer vision*, pp. 213–226. Springer, (2010).

[31] Filippo Santambrogio, 'Optimal transport for applied mathematicians', *Birkäuser, NY*, **55**(58-63), 94, (2015).

[32] Morgan A Schmitz, Matthieu Heitz, Nicolas Bonneel, Fred Ngole, David Coeurjolly, Marco Cuturi, Gabriel Peyré, and Jean-Luc Starck, 'Wasserstein dictionary learning: Optimal transport-based unsupervised nonlinear dictionary learning', *SIAM Journal on Imaging Sciences*, **11**(1), 643–678, (2018).

[33] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu, 'Wasserstein distance guided representation learning for domain adaptation', in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, (2018).

[34] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller, 'Covariate shift adaptation by importance weighted cross validation.', *Journal of Machine Learning Research*, **8**(5), (2007).

[35] Rosanna Turrisi, Rémi Flamary, Alain Rakotomamonjy, et al., 'Multi-source domain adaptation via weighted joint distributions optimal transport', in *The 38th Conference on Uncertainty in Artificial Intelligence*, (2022).

[36] Vladimir Vapnik, 'Principles of risk minimization for learning theory', *Advances in neural information processing systems*, **4**, (1991).

[37] Hang Wang, Minghao Xu, Bingbing Ni, and Wenjun Zhang, 'Learning to combine: Knowledge aggregation for multi-source domain adaptation', in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pp. 727–744. Springer, (2020).

[38] Bo Zhang, Wei Li, Xiao-Li Li, and See-Kiong Ng, 'Intelligent fault diagnosis under varying working conditions based on domain adaptive convolutional neural networks', *Ieee Access*, **6**, 66367–66384, (2018).