LEARNING DICTIONARIES OVER DATASETS THROUGH WASSERSTEIN BARYCENTERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Dictionary learning consists of trying to represent objects in terms of basic elements (atoms) weighted by an importance factor (representation). Non-linear dictionary learning using optimal transport as a metric has been previously studied for normalized non-negative data on a fixed grid. We propose a new framework by using Wasserstein Dictionary Learning on datasets understood as empirical distributions. We leverage Wasserstein barycenters for learning a dictionary of virtual datasets and embeddings in a simplex. We apply our method for unsupervised domain adaptation, improving the state-of-the-art over 1.96% and 2.70%, respectively, and manifold learning of Gaussian distributions and color histograms.

1 INTRODUCTION

The idea of a barycenter is an ancient concept in geometry. From a physical perspective, it is defined as the center of mass of a group of objects. This perspective reflects its etymology, as the word *barycenter* comes from Greek and can be roughly translated as *heavy center*. Mathematically, barycenters are associated with the notion of a mean or central point. This view is advocated by Fréchet (1948), who studied this problem in the case of metric spaces, and more generally by Karcher (1977), in the context of Riemannian manifolds.

The mathematical formulation of a barycenter allows for the definition of centroids of abstract objects, such as probability distributions. A principled way of doing this is through Optimal Transport (OT). This mathematical framework is used in data science and machine learning (Peyré et al. (2019)) for comparing and matching probability distributions. The recent interest in OT is due to its solid theoretical formulation and state-of-the-art results in problems such as generative modeling (Arjovsky et al. (2017)) and transfer learning (Courty et al. (2016)).

Optimal transport defines a metric known as the Wasserstein distance (Santambrogio (2015)), which metrizes the space of probability distributions. Hence, OT allows for the extension of geometric concepts to this abstract space, such as geodesics (McCann (1997)) and barycenters (Agueh & Carlier (2011)). Especially, Wasserstein barycenters have a wide range of applications in data science (Rabin et al. (2011); Cuturi & Doucet (2014)); generative modeling (Fan et al. (2021)); reinforcement learning (Metelli et al. (2019)); domain adaptation (Montesuma & Mboula (2021b)); natural language processing (Huynh et al. (2020)); dictionary learning (Schmitz et al. (2018)); graph learning (Brogat-Motte et al. (2022)).

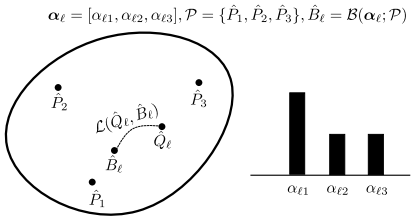


Figure 1: Our framework learns a set of atoms $\mathcal{P}$ and weights $\mathcal{A}$ by minimizing the reconstruction loss between barycenters $\hat{B}_\ell$ and datasets $\hat{Q}_\ell$.

Contributions: We propose a new dictionary learning problem over datasets, using Wasserstein barycenters. As such, Our method relies on previous work (Bonneel et al. (2016); Schmitz et al. (2018)). An illustration is presented in Figure 1. We explore 2 types of applications. First, our dictionary embeds datasets in the K probability simplex Δ_K. As such, our framework can be applied for manifold learning on the space of distributions. Second, our dictionary is used for domain adaptation, where we rely on the learned atoms. We believe our work opens new research

directions, especially concerning the non-linear factorization of complex datasets in terms of simpler components.

This paper is structured as follows. Section 2 covers the necessary theory behind our method. Section 3 introduces our view on dictionary learning. Section 4 covers experiments in manifold learning of distributions and domain adaptation. Section 5 presents our conclusions.

2 PRELIMINARIES

2.1 DICTIONARY LEARNING

Let $\Delta_K = \{\mathbf{a} \in \mathbb{R}_+^K : \sum_{k=1}^K a_k = 1\}$ denote the K probability simplex. In representation learning, Dictionary Learning (DiL) is a technique that tries to express a collection of N vectors $\mathbf{X} \in \mathbb{R}^{N\times d}$ in terms of a dictionary composed by a set of K atoms $\mathcal{P} \in \mathbb{R}^{K\times d}$ and a set of *weights* $\mathcal{A} \in \mathbb{R}^{N\times K}$. Therefore DiL poses the following optimization problem,

$$(\mathcal{P}^\star, \mathcal{A}^\star) = \underset{\mathcal{P},\mathcal{A}}{\text{argmin}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathbf{x}_\ell, \mathcal{P}^T \alpha_\ell), \tag{1}$$

where $\mathcal{L}$ is a suitable loss function. When the $\mathbf{x}_\ell$ are histograms (i.e. $\mathbf{X} \in (\Delta_d)^N$), problem 1 is known as Nonnegative Matrix Factorization (NMF) (Paatero & Tapper (1994)). In this case $\mathcal{P} \in (\Delta_d)^K$ and $\mathcal{A} \in (\Delta_K)^N$. NMF in natural language processing (Xu et al. (2003)) and computer vision (Hoyer (2004)). In this case, the data points $\mathbf{x}_\ell$ can be understood as probability distributions over a fixed grid. As consequence, $\mathcal{L}$ is commonly taken as the Kullback-Leibler (KL) divergence, or the Wasserstein distance (Rolet et al. (2016)). A further contribution of OT consists of substituting the linear aggregation of atoms with a Wasserstein barycenter (Schmitz et al. (2018)).

2.2 STATISTICAL LEARNING AND DOMAIN ADAPTATION

In machine learning and statistics, Statistical Learning Theory (SLT) (Vapnik (2013)) formalizes the problem of learning a function $h : \mathbb{R}^d \to \{1, \cdots, n_c\}$ among a family $\mathcal{H}$ that maps a feature space $\mathbb{R}^d$ into a set of n_c classes. Let $\mathbf{x} \in \mathbb{R}^d, \mathbf{x} \sim P$; SLT is based on the principle of risk minimization, where the risk $\mathcal{R}_P$ between classifiers h and h' is,

$$\mathcal{R}_P(h, h') = \mathbb{E}_{\mathbf{x}\sim P}[l(h(\mathbf{x}), h'(\mathbf{x}))]. \tag{2}$$

Calculating $\mathcal{R}_P$ is impractical because P is *unknown a priori*. One makes two assumptions: (i) data, $\{\mathbf{x}_i^{(P)}\}_{i=1}^n$, are Independent and Identically Distributed (i.i.d.) according to P; (ii) features are labeled from an *unknown* ground-truth labeling function $h_0 : \mathbb{R}^d \to \{1, \cdots, n_c\}$. Henceforth, we adopt the following abbreviation: $\mathcal{R}_P(h) = \mathcal{R}_P(h, h_0)$. This leads to the celebrated Empirical Risk Minimization (ERM) principle of Vapnik (2013):

$$\hat{h}^\star = \underset{h\in\mathcal{H}}{\text{argmin}}\, \hat{\mathcal{R}}_P(h) = \frac{1}{n} \sum_{i=1}^{n} l(h(\mathbf{x}_i^{(P)}), y_i^{(P)}),$$

where $y_i^{(P)} = h_0(\mathbf{x}_i^{(P)})$. Nonetheless, the i.i.d. assumption is restrictive since real-world data may be acquired under different regimes. In this case, train and test data may follow different distributions (Quinonero-Candela et al. (2008)). Examples of this occur in image processing (Patel et al. (2015)), natural language processing (Ramponi & Plank (2020)) and fault diagnosis (Montesuma et al. (2022)).

Transfer Learning (TL) (Pan & Yang (2009)) is a framework for such non-standard learning scenario. Let a domain $\mathcal{D} = (\mathcal{X}, P(X))$ be a pair of a feature space (e.g., $\mathcal{X} = \mathbb{R}^d$) and a probability distribution, and a task $\mathcal{T} = (\mathcal{Y}, P(Y|X))$ be a pair of a label space (e.g., $\mathcal{Y} = \{1, \cdots, n_c\}$) and a predictive distribution $P(Y|X)$. TL seeks to improve performance on the target domain/task given knowledge from a source domain/task. In this context, unsupervised Domain Adaptation (DA) is a sub-problem within TL where one assumes different domains $\mathcal{D}_S$ and $\mathcal{D}_T$, characterized by a shift in their distributions $P_S \neq P_T$. Furthermore, during training, one has access to labeled data from the source and only features from the target.

Methods for DA differ in their assumptions and setting. A first distinction is between shallow (e.g., Pan et al. (2010)) and deep DA (e.g., Ganin et al. (2016)). The first case relies on feature extraction (e.g., a pre-trained deep Neural Network (NN)) and usually performs adaptation by transforming these features. The second focuses on using target domain features for forcing the feature extraction part of a deep NN to learn invariant representations. The second distinction is the setting for performing DA. Most methods are *single source*, as they consider that adaptation occurs between a single source and target. When multiple sources are available, one has Multi-Source Domain Adaptation (MSDA).

Within DA, OT is a statistical framework for novel contributions. Indeed, new algorithms were developed for shallow (Courty et al. (2016; 2017)), deep (Damodaran et al. (2018)), and MSDA (Montesuma & Mboula (2021a) and Turrisi et al. (2022)). In addition, the validity of this framework has been established by Redko et al. (2017), where the authors provided an analysis of how the risk on target $\mathcal{R}_{P_T}$ is bounded to the risk on source $\mathcal{R}_{P_S}$, similarly to Ben-David et al. (2010). When proposing new algorithms for DA, a common practice is establishing theoretical results bounding these two terms.

2.3 Optimal Transport

In what follows, we model datasets through empirical distributions. Let P be an unknown distribution, from which we assume an i.i.d. sample of size n, $\mathbf{X}^{(P)} = [\mathbf{x}_1^{(P)}, \cdots, \mathbf{x}_n^{(P)}]$. An empirical approximation of P is,

$$\hat{P}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \delta(\mathbf{x} - \mathbf{x}_i^{(P)}), \tag{3}$$

where $\mathbf{X}^{(P)}$ is also called the support $\hat{P}$. Formally, an arbitrary empirical distribution is *parametrized* by its support. Henceforth we consider the subset $\mathcal{M} = \{\hat{P} : \mathbf{X}^{(P)} \in \mathbb{R}^{n \times d}\}$ of the space of distributions over $\mathbb{R}^d$. We are interested in how OT defines a geometry on $\mathcal{M}$ through the formulation of Kantorovich (1942).

For $\hat{P}, \hat{Q}$, one seeks a transportation plan $\pi \in \mathbb{R}^{n \times m}$, where $\pi_{i,j}$ corresponds to the amount of mass transported from $\mathbf{x}_i^{(P)}$ to $\mathbf{x}_j^{(Q)}$. Additionally, π is required to preserve mass, that is, $\sum_i \pi_{i,j} = 1/m$ and $\sum_j \pi_{i,j} = 1/n$, or $\pi \in U(\mathbf{u}_n, \mathbf{u}_m)$, where $\mathbf{u}_n$ is the vector with uniform entries $1/n$ and $U(\mathbf{a}, \mathbf{b})$ is the set of bi-stochastic matrices with marginals $\mathbf{a}$ and $\mathbf{b}$. Therefore,

$$\pi^\star = \text{OT}(\mathbf{u}_n, \mathbf{u}_m, \mathbf{C}) = \underset{\pi \in U(\mathbf{u}_n, \mathbf{u}_m)}{\text{argmin}} \sum_{i=1}^{n} \sum_{j=1}^{m} C_{i,j} \pi_{i,j} = \langle \mathbf{C}, \pi \rangle_F, \tag{4}$$

is called the OT problem between $\hat{P}$ and $\hat{Q}$. In Equation 4, $C_{i,j} = c(\mathbf{x}_i^{(P)}, \mathbf{x}_j^{(Q)})$ is called *ground-cost matrix*, and reflects the effort of transporting $\mathbf{x}_i^{(P)}$ into $\mathbf{x}_j^{(Q)}$. When c is a distance, OT defines a distance between distributions ((Peyré et al., 2019, Proposition 2.2)) called Wasserstein distance, $W_c(\hat{P}, \hat{Q}) = \langle \mathbf{C}, \pi^\star \rangle_F$. For $\mathcal{P} = \{\hat{P}_k\}_{k=1}^K$ and $\boldsymbol{\alpha} \in \Delta_K$, the Wasserstein barycenter (Agueh & Carlier (2011)) is a solution to

$$B^\star = \mathcal{B}(\boldsymbol{\alpha}; \mathcal{P}) = \inf_B \sum_{k=1}^{K} \alpha_k W_c(P_k, B). \tag{5}$$

The mapping $\mathcal{B}(\cdot; \mathcal{P}) : \Delta_K \to \mathcal{M}$ is called *barycentric operator*. For empirical $\hat{P}_k$, calculating $\hat{B}^\star$ corresponds to estimating $\mathbf{X}^{(B)}$. We propose using the fixed-point iteration of Álvarez-Esteban et al. (2016),

$$(\mathbf{C}_k)_{i,j} = \|\mathbf{x}_i^{(P_k)} - \mathbf{x}_j^{(B)}\|_2^2, \qquad \pi_k = \text{OT}(\mathbf{u}_{n_k}, \mathbf{u}_n, \mathbf{C}_k), \qquad \mathbf{X}^{(B)} = n \sum_{k=1}^{K} \alpha_k \pi_k^T \mathbf{X}^{(P_k)}, \tag{6}$$

until convergence. Further discussion is available in appendix A. When the distributions under analysis are labeled, one needs to define c between pairs $(\mathbf{x}_i^{(P)}, y_i^{(P)})$ and $(\mathbf{x}_i^{(Q)}, y_i^{(Q)})$ (e.g., Alvarez-Melis & Fusi (2020)). We propose using the following ground-cost,

$$C_{i,j} = \|\mathbf{x}_i^{(P)} - \mathbf{x}_j^{(Q)}\|_2^2 + \beta \|\mathbf{Y}_i^{(P)} - \mathbf{Y}_j^{(Q)}\|_2^2, \tag{7}$$

where $\mathbf{Y}_i^{(P)}$ is the one-hot encoding of $y_i^{(P)}$. In particular we adopt $\beta = \max_{i,j} \|\mathbf{x}_i^{(P)} - \mathbf{x}_j^{(Q)}\|_2^2$ so that π is class sparse (i.e. $\pi_{i,j} > 0 \iff y_i^{(P)} = y_j^{(Q)}$).

In previous work, Courty et al. (2016) and Montesuma & Mboula (2021a) proposed a ground-cost similar to Equation 7. We use the Euclidean distance as it allows us to label the barycenter support using the label propagation procedure of Redko et al. (2019),

$$\mathbf{Y}^{(B)} = n \sum_{k=1}^{K} \alpha_k \pi_k^T \mathbf{Y}^{(P_k)}. \tag{8}$$

We propose viewing $\mathcal{B}(\cdot; \mathcal{P})$ as a generative model, by using iterations 6 and Equation 8 for generating $\mathbf{X}^{(B)}$ and $\mathbf{Y}^{(B)}$. This generative view is illustrated in Figure 2, which shows the interpolation for $\mathcal{P} = \{\hat{P}_1, \hat{P}_2\}$ and $\boldsymbol{\alpha} = [t, 1-t]$, $0 \leq t \leq 1$. In this sense, $\boldsymbol{\alpha}$ is used to control the location over the set $\mathcal{M}_{\mathcal{P}} = \{\mathcal{B}(\boldsymbol{\alpha}; \mathcal{P}) : \boldsymbol{\alpha} \in \Delta_K\}$.

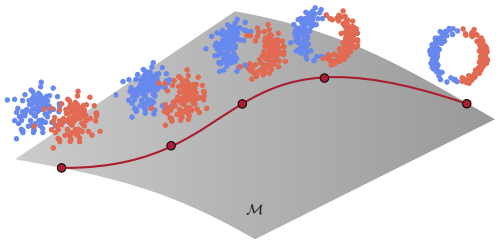


Figure 2: Interpolation under the Wasserstein metric between joint distributions in a statistical manifold through Wasserstein barycenters.

3 Dataset Dictionary Learning

We propose learning dictionaries over datasets understood as empirical distributions, rather than fix grid histograms, as featured in previous works. We name our framework Dataset Dictionary Learning (DaDiL). We take inspiration from the work of Poincaré (1908)[1] in believing that *this remains a dictionary learning problem*. For N datasets $\mathcal{Q} = \{\hat{Q}_\ell\}_{\ell=1}^N$,

$$(\mathcal{P}^\star, \mathcal{A}^\star) = \underset{\mathcal{P},\mathcal{A}}{\text{argmin}}\, L(\mathcal{P}, \mathcal{A}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\hat{Q}_\ell, \mathcal{B}(\boldsymbol{\alpha}_\ell; \mathcal{P})), \tag{9}$$

for $\mathcal{P} = \{\hat{P}_k\}_{k=1}^K$ and $\mathcal{A} \in (\Delta_K)^N$. Each distribution $\hat{P}_k$ is parametrized by its support $\mathbf{X}^{(P_k)} \in \mathbb{R}^{n \times d}$. Equation 9 is quite flexible, as one can consider $\hat{P}_k$ as an empirical joint distribution over features and labels, in which case we represent the support by matrices $\mathbf{X}^{(P_k)}$ and $\mathbf{Y}^{(P_k)}$. Therefore $\hat{B}_\ell$ is an approximation for the marginal $Q_\ell(X)$ or joint $Q_\ell(X, Y)$. Different choices of loss function can be made (e.g., the maximum mean discrepancy of Gretton et al. (2007)). In our experiments, we use the Wasserstein distance W_c (Equation 4), for $C_{i,j}$ defined in Equation 7. Whenever $\hat{P}$ or $\hat{Q}$ in Equation 4 is unlabeled, we set $\lambda = 0$ in Equation 7.

Algorithm 1 DaDiL learning loop.

Require: $\mathcal{Q} = \{\hat{Q}_\ell\}_{\ell=1}^N$, number of iterations N_{iter}, number of atoms K, loss $\mathcal{L}$, batch size n_b

1: Initialize $\mathbf{x}_i^{(P_k)} \sim \mathcal{N}(0, \mathbf{I}_d)$, $\mathbf{a}_\ell \sim \mathcal{N}(0, \mathbf{I}_K)$.
2: **for** $it = 1 \cdots, N_{iter}$ **do**
3: $\forall k$ sample $\mathbf{X}^{(P_k)} = \{\mathbf{x}_1^{(P_k)}, \cdots, \mathbf{x}_{n_b}^{(P_k)}\}$,
4: $\forall \ell$ sample $\mathbf{X}^{(Q_\ell)} = \{\mathbf{x}_1^{(Q_\ell)}, \cdots, \mathbf{x}_{n_b}^{(Q_\ell)}\}$,
5: $\forall \ell$ change variables $\alpha_\ell = \text{softmax}(\mathbf{a}_\ell)$
6: $\forall \ell$ calculate $\mathbf{X}^{(B_\ell)} = \mathcal{B}(\alpha_\ell; \mathcal{P})$
7: Calculate $L = 1/N \sum_{\ell=1}^N \mathcal{L}(\mathbf{X}^{(B_\ell)}, \mathbf{X}^{(Q_\ell)})$
8: $\forall i, k$, update $\mathbf{x}_i^{(P_k)}$ using $\partial L / \partial \mathbf{x}_i^{(P_k)}$,
9: $\forall \ell$, update $\mathbf{a}_\ell$ using $\partial L / \partial \mathbf{a}_\ell$.
10: **end for**

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

We propose to optimize equation 9 through mini-batch OT (Fatras et al. (2020)). As such, we sample mini-batches from $\hat{P}_k$, then we calculate $\hat{B}_\ell = \mathcal{B}(\boldsymbol{\alpha}_\ell; \mathcal{P})$. We remark that $\mathcal{B}$ is differentiable w.r.t. α_ℓ and

[1] Mathematicians do not study objects, but the relations between objects; to them it is a matter of indifference if these objects are replaced by others, provided that the relations do not change. Matter does not engage their attention, they are interested by form alone.

the atom's support points $\mathbf{x}_i^{(P_k)}$ (see Appendix A). The loss is then calculated using $\hat{B}_\ell$ and mini-batches from $\hat{Q}_\ell$. Similarly to Bonneel et al. (2016) and Schmitz et al. (2018), we optimize w.r.t $\mathbf{A} = \{a_{\ell,k}\} \in \mathbb{R}^{N\times K}$, then perform a change of variables $\boldsymbol{\alpha}_\ell = \text{softmax}(\mathbf{a}_\ell)$. The overall procedure is shown in algorithm 1, which is implemented using Pytorch (automatic differentiation) of Paszke et al. (2019) and Python Optimal Transport (OT-related details) of Flamary et al. (2021).

3.1 A PROXY FOR THE EMPIRICAL WASSERSTEIN DISTANCE

Our learned dictionary approximates $\mathcal{M}$ through $\mathcal{M}_\mathcal{P} = \{\mathcal{B}(\boldsymbol{\alpha};\mathcal{P}) : \boldsymbol{\alpha} \in \Delta_K\}$. We propose using $\mathcal{P}$ and $\mathcal{A}$ for defining a proxy for the empirical Wasserstein distance W_2. We call this proxy Wasserstein Embedding Distance (WED),

$$\text{WED}(\hat{Q}_1, \hat{Q}_2) = \underset{\pi\in U(\alpha_1,\alpha_2)}{\text{argmin}} \sum_{k_1=1}^{K}\sum_{k_2=1}^{K} \pi_{k_1,k_2} W_c(\hat{P}_{k_1}, \hat{P}_{k_2}), \tag{10}$$

The WED relies on the notion of barycentric coordinates (Bonneel et al. (2016)), defined as,

$$\boldsymbol{\alpha}^\star = \underset{\boldsymbol{\alpha}\in\Delta_K}{\text{argmin}} \sum_{k=1}^{K} \alpha_k W_c(\hat{P}_k, \hat{Q}). \tag{11}$$

Note that, since for $\hat{Q}_1 \neq \hat{Q}_2$, one may have the same minimizer $\boldsymbol{\alpha}^\star$, the WED is a semi-metric rather than a proper metric (i.e., $\text{WED}(\hat{Q}_1, \hat{Q}_2) = 0$ for $\hat{Q}_1 \neq \hat{Q}_2$). This is explored in the next theorem, which is proven in appendix A.3,

Theorem 3.1. Let $\hat{Q}_1$ and $\hat{Q}_2$ be 2 empirical distributions, and $\mathcal{P} = \{\hat{P}_k\}_{k=1}^K$ be a set of atoms learned by minimizing equation 9. In this case, the WED is a pseudo-metric in $\mathcal{M}$.

Note that calculating the WED is simpler than W_c since one needs to: (i) calculate and store the pairwise $W_c(\hat{P}_{k_1}, \hat{P}_{k_2})$; (ii) solve a small $K \times K$ OT problem. Especially if the distributions $\hat{P}_k$ have much fewer samples than $\hat{Q}_\ell$, all $W_2(\hat{P}_{k_1}, \hat{P}_{k_2})$ is cheaper than $W_c(\hat{Q}_1, \hat{Q}_2)$. We present a theoretical result bounding the WED by the Wasserstein distance of different terms in our dictionary learning problem,

Theorem 3.2. Let $\hat{Q}_1$ and $\hat{Q}_2$ be 2 empirical distributions, and $\mathcal{P} = \{\hat{P}_k\}_{k=1}^K$ be a set of learned atoms. For $\boldsymbol{\alpha}_\ell$, the solution of equation 11 for $\hat{Q} = \hat{Q}_\ell$, and $\hat{B}_\ell = \mathcal{B}(\boldsymbol{\alpha}_\ell;\mathcal{P})$ the WED is bounded as follows,

$$\text{WED}(\hat{Q}_1, \hat{Q}_2) \leq \sum_{k=1}^{K} \alpha_{1,k} W_c(\hat{P}_k, \hat{B}_1) + \sum_{k=1}^{K} \alpha_{2,k} W_c(\hat{P}_k, \hat{B}_2) + W_c(\hat{Q}_1, \hat{Q}_2). \tag{12}$$

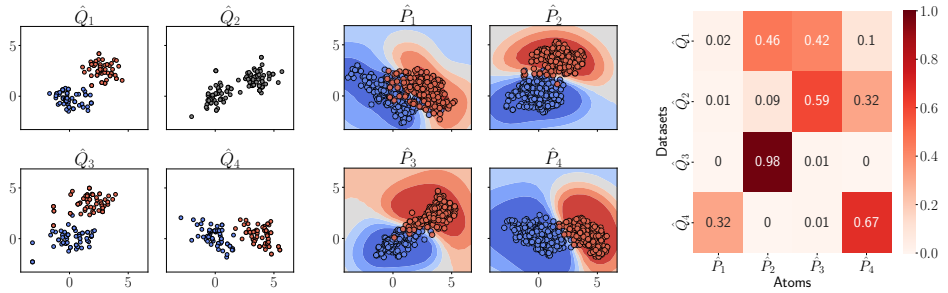
Equation 12 bounds our proposed WED by the Wasserstein distance $W_2(\hat{Q}_1, \hat{Q}_2)$, plus terms linked to the geometry of $\mathcal{P}$. These are constants that do not depend on $\hat{Q}_\ell$.

3.2 DOMAIN ADAPTATION

The second application of DaDiL is MSDA. As follows, we suppose N source labeled distributions $\{\hat{Q}_{S_\ell}(X,Y)\}_{\ell=1}^N$, and a single unlabeled target distribution $\hat{Q}_T(X)$. We want to learn a dictionary $\{\hat{P}_k(X,Y)\}_{k=1}^K$ of labeled distributions, using W_c, with c as in Equation 7. An example of a dictionary of datasets is shown in Figure 3. We propose two strategies for MSDA.

The first strategy, DaDiL-R, is based on the reconstruction $\hat{B}_T = \mathcal{B}(\boldsymbol{\alpha}_T;\mathcal{P})$ of $\hat{Q}_T$. To justify this approach, we use (Redko et al., 2017, Theorem 2) which implies,

$$\mathcal{R}_{Q_T}(h) \leq \mathcal{R}_{B_T}(h) + W_1(\hat{Q}_T, \hat{B}_T) + \sqrt{2\frac{\log(\frac{1}{\delta})}{\zeta}}\left(\sqrt{\frac{1}{n_{Q_T}}} + \sqrt{\frac{1}{n}}\right) + \lambda. \tag{13}$$



(a) Set of datasets. (b) Learned atoms using DaDiL. (c) Learned atoms using DaDiL.

Figure 3: Illustration of the dictionary learned by DaDiL in DA. We want to perform dictionary learning over a set of datasets (a), some of which may be unlabeled (e.g., $\hat{Q}_2$). The proposed dictionary comprises labeled distributions, as shown in (b) and (c).

Therefore, $\mathcal{R}_{Q_T}(h)$ is bounded by $\mathcal{R}_{B_T}(h)$ plus the term $W_1(\hat{Q}_T, \hat{B}_T)$, which is minimized in Equation 9. The other terms are related to the sample size and the optimal joint error $\lambda = \text{argmin}_{h\in\mathcal{H}} \mathcal{R}_{Q_T}(h) + \mathcal{R}_{B_T}(h)$. Since $\hat{Q}_T$ is unlabeled, λ can not be controlled directly. However, similarly to Redko et al. (2017), we hypothesize that the inclusion of labels in the ground-cost and the presence of labels in the other sources help control λ. Further discussion is presented in Appendix A.4. The second strategy, DaDiL-E, is based on ensembling classifiers $\hat{h}_k = \text{argmin}_{h\in\mathcal{H}} \hat{\mathcal{R}}_{P_k}(h)$ learned with atom samples. The resulting classifier is expressed by $\hat{h}_{\boldsymbol{\alpha}_T} = \sum_{k=1}^{K} \alpha_{T,k} \hat{h}_k$. We provide the following theoretical result in favor of this strategy,

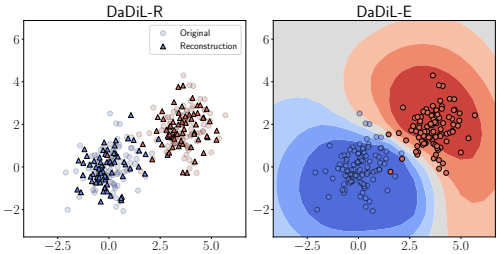


Figure 4: On the left, DaDiL-R reconstructs samples in $\hat{Q}_T$. On the right, DaDiL-E uses $\boldsymbol{\alpha}_T$ for weighting classifiers learned on atoms.

Theorem 3.3. Let $\{\mathbf{X}^{(P_k)}\}_{k=1}^{K}$ and $\mathbf{X}^{(Q_T)}$ be samples of size n and n_{Q_T} from P_k and Q_T. Let $\{\hat{P}_k\}_{k=1}^{K}$ and $\hat{Q}_T$ be the associated empirical measures. Let h_k be the minimizer of $\mathcal{R}_{P_k}$. Then, for any $d' > d$ and $\zeta > \sqrt{2}$ there exists some constant N_0, depending on d' such that for any $\delta > 0$ and $\min(n, n_{Q_{N+1}}) > N_0 \max(\delta^{-d'+2}, 1)$ and $\boldsymbol{\alpha} \in \Delta_K$, with probability at least $1 - \delta$,

$$\mathcal{R}_{Q_T}(h_{\boldsymbol{\alpha}}) \leq \mathcal{R}_{\boldsymbol{\alpha}}(h_{\boldsymbol{\alpha}}) + \sum_{k=1}^{K} \alpha_k (W_1(\hat{P}_k, \hat{Q}_T) + \lambda_k + c), \tag{14}$$

where $h_{\boldsymbol{\alpha}} = \sum_{k=1}^{K} \alpha_k h_k$, $\lambda_k = \underset{h\in\mathcal{H}}{\text{minimize}}\ \mathcal{R}_{P_k}(h) + \mathcal{R}_{Q_T}(h)$,

$$\mathcal{R}_{\boldsymbol{\alpha}}(h) = \sum_{k=1}^{K} \alpha_k \mathcal{R}_{P_k}(h) \text{ and } c = \sqrt{\frac{2\log(1/\delta)}{\zeta}} \left(\sqrt{\frac{1}{n}} + \sqrt{\frac{1}{n_{Q_T}}} \right).$$

The bound in Equation 14 is very similar to that of equation 13, but it takes into account how the atom classifiers are weighted to form $h_{\boldsymbol{\alpha}}$. Note that for $\boldsymbol{\alpha}_T$, the r.h.s. of Equation 14 is minimized. We make a similar assumption to DaDiL-R and Redko et al. (2017), in considering that using Equation 7 induces a low λ_k. An illustration of our strategies is shown in Figure 4.

4 Experiments

4.1 Toy Example: Learning the Manifold of Gaussian Distributions

We explore how we can learn a dictionary over the Gaussian distributions with unit variance. We thus consider $Q_\ell = \mathcal{N}(\mu_\ell, \mathbf{I})$, where $\mu_\ell \in [0,8]^2 \subset \mathbb{R}^2$ is the mean vector. We discretize $[0,8]^2$ using 25 uniformly spaced points. For each Q_ℓ, we sample $n = 512$ points, generating datasets $\hat{Q}_\ell$, as shown on the left of Figure 5. We learn eight dictionaries with an increasing number of atoms. The training curves are shown on the right of Figure 5. Except for $K = 2$, all dictionaries provide good reconstructions for the manifold. Note that since $\mu = [\mu_1, \mu_2]$ is a coordinate system for $\mathcal{Q}$, one needs at least three atoms to recover faithful reconstructions.

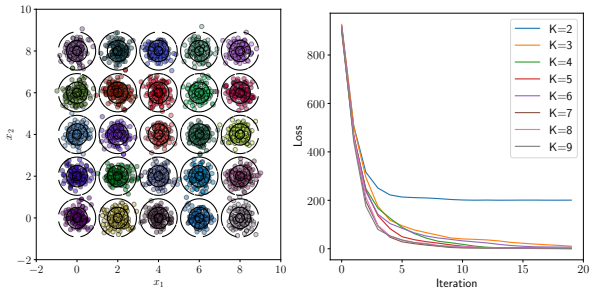


Figure 5: (Left) Manifold of unit variance Gaussian distributions. (Right) Learning curves of dictionaries.

Now, let us focus on $K = 3$. We can represent the latent space using $\{(\alpha_1, \alpha_2) : \alpha_1 + \alpha_2 \leq 1\}$, since $\alpha_3 = 1 - \alpha_2 - \alpha_1$. Let us fix $\alpha_0 = \mathbf{u}_3$ and calculate $\text{WED}(\boldsymbol{\alpha}_0, \boldsymbol{\alpha})$ and $W_2(\hat{B}_0, \hat{B})$ for each $\alpha \in \Delta_3$, $\hat{B} = \mathcal{B}(\boldsymbol{\alpha}; \mathcal{P})$. This is shown in Figure 6. First, since each $Q_\ell \in \mathcal{Q}$ is Gaussian, so is B ((Agueh & Carlier, 2011, Section 6.3)). Therefore $W_2(\hat{B}_0, \hat{B}) = \|\sum_k (\alpha_{0,k} - \alpha_k)\mu_k\|_2^2$, which explains the quadratic contours. Second, the contours of the WED are associated with the positions of the learned atoms and the polytope $U(\boldsymbol{\alpha}_0, \boldsymbol{\alpha})$, since the ground-cost matrix $C_{k_1,k_2} = W_2(\hat{P}_{k_1}, \hat{P}_{k_2})$ does not depend on $\boldsymbol{\alpha}_0$ or $\boldsymbol{\alpha}$. Indeed, for each $\boldsymbol{\alpha}$, the WED is defined by a 3×3 OT problem. Thus one has much less degrees of freedom (w.r.t. $\boldsymbol{\alpha} \in \Delta_3$) than with the Wasserstein distance, which is a 512×512 OT problem.

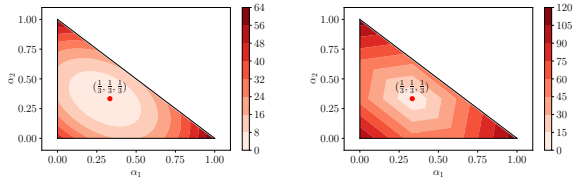


Figure 6: W_2 (left) and WED (right) over points in the latent space Δ_3.

4.2 Multi-Source Domain Adaptation

We compare our strategy with shallow and deep DA methods. We divide our experiments into (i) shallow DA, which transforms features from a fixed feature extractor, and (ii) deep DA, which adapts a deep feature extractor during training. In both cases, the baseline consists of a classifier trained on source domain data and evaluated on target. For shallow DA, we use the Office-Caltech10 dataset (Saenko et al. (2010); Griffin et al. (2007)). This dataset has four domains: Amazon (A), Webcam (W), dSLR (D), and Caltech (C), with 958, 295, 157, and 1123 samples. We use the seventh layer activations of the DeCAF network (Donahue et al. (2014)) as features. For deep DA, we use the Office 31 dataset (Saenko et al. (2010)), which has three domains: Amazon (A), Webcam (W), and dSLR (D), with 2817, 795, and 498 samples, respectively. Additional information is available in Appendix B. We summarize our results in Table 1.

Shallow DA: we evaluate two types of methods: (i) single-source methods, such as Subspace Alignment (SA) (Fernando et al. (2013)), Transfer Component Analysis (TCA) (Pan et al. (2010)) and Optimal Transport Domain Adaptation (OTDA) (Courty et al. (2016)). (ii) MSDA, such as WBT_{reg} (Montesuma & Mboula (2021a)), and Weighted JDOT (WJDOT) (Turrisi et al. (2022)). On the left of Table 1, we show the classification accuracy (in %). * And † indicate results reproduced from Montesuma & Mboula (2021a) and Turrisi et al. (2022), respectively. Our method presents an improvement of 1.96% over previous state-of-the-art (WJDOT and WBT_{reg}), particularly better on Webcam and Caltech domains.

We highlight that WJDOT relies on finding a target domain classifier aligned with a convex combination of source distributions. In their experiments, Turrisi et al. (2022) show that the weights learned by their algorithm are one-hot encoded vectors. Hence, WJDOT chooses one source do-

Table 1: Classification accuracy (in %) of domain adaptation methods using a ResNet-50 backbone. Each column represents a target domain for the Caltech-Office 10 (left) and Office 31 (right) benchmarks.

	Shallow DA				
Method	A	D	W	C	Avg
Baseline	92.36 ± 1.21	98.33 ± 1.56	93.26 ± 1.88	86.58 ± 1.92	92.63
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	$\underline{97.51} \pm \underline{1.18}$	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†	$\mathbf{94.23 \pm 0.90}$	$\mathbf{100.00 \pm 0.00}$	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	93.26 ± 1.53	97.50 ± 2.43	$\mathbf{98.88 \pm 0.71}$	$\underline{86.65} \pm \underline{1.13}$	$\underline{94.07}$
DaDiL-E	$\underline{93.89} \pm \underline{1.81}$	$\underline{98.33} \pm \underline{0.83}$	$\mathbf{98.88 \pm 0.71}$	$\mathbf{87.00 \pm 1.42}$	**94.52**

	Deep DA			
Method	A	D	W	Avg
Baseline	57.45	98.00	92.45	82.63
DANN	58.69	98.00	93.08	83.25
WDGRL	59.75	98.00	94.34	84.03
MCD	62.23	98.00	94.34	84.85
MDD	64.36	$\underline{99.00}$	94.34	85.90
MOST	64.01	97.00	96.23	85.74
M3SDA	50.35	**100.00**	**98.74**	83.03
WBT_{reg}	65.78	91.00	89.94	82.24
DaDiL-R	$\underline{72.52}$	95.00	$\underline{98.11}$	$\underline{88.54}$
DaDiL-E	**73.58**	96.00	96.23	**88.60**

main (optimal w.r.t. domain similarity) for the adaptation. On the other hand, our method extracts information from all sources, which proves beneficial for W and C domains.

Deep DA: we compare DaDiL-R and DaDiL-E with deep domain adaptation methods. In our experiments, we fine-tune a ResNet-50 model (He et al. (2016)) on the concatenation of all source domain data (e.g., W and D) and evaluate it on the target (e.g., A). This setting constitutes the baseline shown on the right of Table 1. For deep learning methods, we compare (i) single source methods, such as Domain Adversarial Neural Network (DANN) (Ganin et al. (2016)); Wasserstein Distance Guided Representation Learning (WDGRL) (Shen et al. (2018)); Maximum Classifier Discrepancy (MCD) (Saito et al. (2018)); Margin Disparity Discrepancy (MDD) (Zhang et al. (2019)); (ii) MSDA methods, such as MOST (Nguyen et al. (2021)), M3SDA (Peng et al. (2019)) and WBT_{reg}.

On average, our method is state-of-the-art, primarily due to its outstanding performance on the *Amazon* domain, surpassing other methods by a margin of 7.80%. On average, it presents an increase of 2.70% in accuracy. Nonetheless, the transfer towards the *dSLR* domain is negative. In this domain, deep methods such as MDD and M3SDA perform better. Overall, the main difference between our method and the deep methods in the state-of-the-art is that we cannot update the parameters of convolutional layers, thus decreasing performance on this domain.

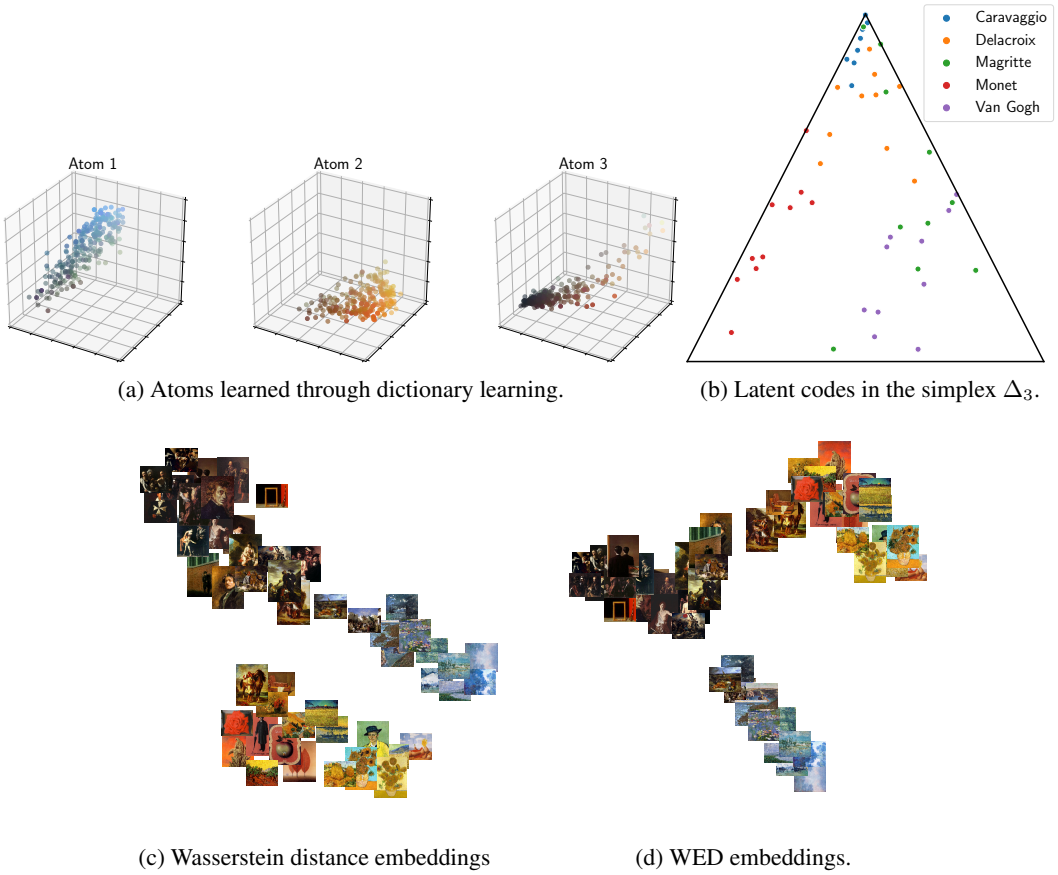
4.3 Learning Dictionaries over Color Histograms

One of the standard applications of OT is the analysis of color histograms (e.g., Ferradans et al. (2014)). This analysis is done by considering RGB images $I_\ell \in \mathbb{R}^{h \times w \times 3}$ as *point clouds* of $n = hw$ samples in $\mathbb{R}^3$. Thus, to each I_ℓ, one has $\hat{Q}_\ell$ with support $\mathbf{X}^{(Q_\ell)} \in \mathbb{R}^{hw \times 3}$. In this experiment, we want to learn a dictionary over $\mathcal{Q} = \{\hat{Q}_\ell\}_{\ell=1}^N$, $N = 50$ artworks of Monet, Delacroix, Magritte, Caravaggio, and Van Gogh, based on their color histograms. We further investigate the geometry induced by the latent codes $\mathcal{A} \in (\Delta_K)^N$ and the WED compared to the Wasserstein distance. Here we discuss our results for $K = 3$, using $N_{tr} = 35$ images to train our dictionary, leaving the rest $N_{ts} = 15$ for test. We can map these images into the latent space using equation 11. We show the result of DiL in figure 7.

We compare the Wasserstein distance with DaDiL and the WED in (i) running time; (ii) their induced geometry. For large-scale images, calculating $W_2(\hat{Q}_{\ell_1}, \hat{Q}_{\ell_2})$ is unfeasible due to the number of pixels. We thus down-sample the support matrices by selecting $n = 2048$ points randomly, hence $\mathbf{X}^{(Q_\ell)} \in \mathbb{R}^{2048 \times 3}$. This step is not necessary for DaDiL since we perform optimization in mini-batches. Concerning running time, the 1225 pairwise W_2 distances took approximately 26 minutes[2]. For DaDiL, training takes an overall of 15 minutes. We remark that, for new distributions, we only need to solve 11 over $N_{ts} \times K$ variables. This is usually faster than calculating pairwise distances.

In Figure 7 (a) we show the learned atoms, which correspond to the predominant colors in the training images. Figure 7 (b) shows the latent codes. The embeddings lead to an intuitive description of images, as they express each image as a combination of blue, yellow, and gray palettes. In Fig-

[2]experiments were performed on a Intel Xeon CPU 2.20 GHz with 12GB of RAM



(a) Atoms learned through dictionary learning.

(b) Latent codes in the simplex Δ_3.

(c) Wasserstein distance embeddings

(d) WED embeddings.

Figure 7: Summary of DaDiL for palette learning. Our algorithm learns palettes that reflect different palette clusters (a). These allow for an useful visualization on the latent space (b). The latent codes can be used to approximate the Wasserstein geometry (c) through the proposed WED (d).

ure 7 (c) and (d), we compare the Wasserstein and WED embeddings using t-distributed Stochastic Neighbor Embeddings (t-SNE) (Van der Maaten & Hinton (2008)), which shows that we are able to capture the Wasserstein geometry faithfully.

5 Conclusion

We present a novel framework for learning dictionaries over datasets. We understand a dictionary as a set of distributions $\{\hat{P}_k\}_{k=1}^K$, and weights $\{\boldsymbol{\alpha}_\ell\}_{\ell=1}^N$, $\boldsymbol{\alpha}_\ell \in \Delta_K$. Each dataset $\{\hat{Q}_\ell\}_{\ell=1}^N$ is expressed as a non-linear combination of $\hat{P}_k$ through Wasserstein barycenters. Each $\boldsymbol{\alpha}_\ell$ is an embedding for $\hat{Q}_\ell$. These embeddings have an associated semi-metric, the WED, that serves as a proxy for the empirical Wasserstein distance. We explore two applications of our method, (i) multi-source domain adaptation, for which our method has theoretical grounding and state-of-the-art results; (ii) learning dictionaries over color palettes, which shows the interest of our modeling in manifold learning. For future works, we want to explore the statistical complexity of our method through the mini-batch OT framework of Fatras et al. (2020), and study our method when the atoms are parametric distributions.

Limitations: first, in DA, our method relies on the quality of feature extractors or embedding functions. In comparison with deep DA, DaDiL is not able to guide representation learning. We plan to pursue this line of work in future research. Second, similarly to Bonneel et al. (2016), our framework is not able to represent distributions outside $\mathcal{M}_\mathcal{P}$. This may lead to inconsistent reconstructions.

REFERENCES

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

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

Pedro C Álvarez-Esteban, E Del Barrio, JA Cuesta-Albertos, and C Matrán. A fixed-point approach to barycenters in wasserstein space. *Journal of Mathematical Analysis and Applications*, 441(2): 744–762, 2016.

David Alvarez-Melis and Nicolo Fusi. Geometric dataset distances via optimal transport. *Advances in Neural Information Processing Systems*, 33:21428–21439, 2020.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3): 334–334, 1997.

François Bolley, Arnaud Guillin, and Cédric Villani. Quantitative concentration inequalities for empirical measures on non-compact spaces. *Probability Theory and Related Fields*, 137(3-4): 541–593, 2007.

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

Luc Brogat-Motte, Rémi Flamary, Céline Brouard, Juho Rousu, and Florence d'Alché Buc. Learning to predict graphs with fused gromov-wasserstein barycenters. *arXiv preprint arXiv:2202.03813*, 2022.

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.

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

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In *International conference on machine learning*, pp. 685–693. PMLR, 2014.

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.

Antoine de Mathelin, François Deheeger, Guillaume Richard, Mathilde Mougeot, and Nicolas Vayatis. Adapt: Awesome domain adaptation python toolbox. *arXiv preprint arXiv:2107.03049*, 2021.

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*, pp. 647–655. PMLR, 2014.

Jiaojiao Fan, Amirhossein Taghvaei, and Yongxin Chen. Scalable computations of wasserstein barycenter via input convex neural networks. In *International Conference on Machine Learning*, pp. 1571–1581. PMLR, 2021.

Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Learning with minibatch wasserstein: asymptotic and gradient properties. In *AISTATS*, 2020.

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.

Sira Ferradans, Nicolas Papadakis, Gabriel Peyré, and Jean-François Aujol. Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences*, 7(3):1853–1882, 2014.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2681–2690. PMLR, 2019.

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.

Maurice Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. In *Annales de l'institut Henri Poincaré*, volume 10, pp. 215–310, 1948.

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.

Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alexander J Smola. A kernel approach to comparing distributions. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, pp. 1637. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007.

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.

Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(9), 2004.

Viet Huynh, He Zhao, and Dinh Phung. Otlda: A geometry-aware optimal transport approach for topic modeling. *Advances in Neural Information Processing Systems*, 33:18573–18582, 2020.

L Kantorovich. On the transfer of masses (in russian). In *Doklady Akademii Nauk*, volume 37, pp. 227–229, 1942.

Hermann Karcher. Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics*, 30(5):509–541, 1977.

Robert J McCann. A convexity principle for interacting gases. *Advances in mathematics*, 128(1): 153–179, 1997.

Alberto Maria Metelli, Amarildo Likmeta, and Marcello Restelli. Propagating uncertainty in reinforcement learning via wasserstein barycenters. In *33rd Conference on Neural Information Processing Systems, NeurIPS 2019*, pp. 4335–4347. Curran Associates, Inc., 2019.

Eduardo F Montesuma, Michela Mulas, Francesco Corona, and Fred Maurice Ngolè Mboula. Cross-domain fault diagnosis through optimal transport for a cstr process. *IFAC-PapersOnLine*, 2022.

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 2021a.

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, 2021b. doi: 10.1109/ICASSP39728.2021.9414199.

Tuan Nguyen, Trung Le, He Zhao, Quan Hung Tran, Truyen Nguyen, and Dinh Phung. Most: Multi-source domain adaptation via optimal transport for student-teacher learning. In *Uncertainty in Artificial Intelligence*, pp. 225–235. PMLR, 2021.

Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

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

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.

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 H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015.

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.

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.

Henri Poincaré. *La science et l'hypothèse*. Flammarion, 1908.

Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 435–446. Springer, 2011.

Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in nlp—a survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6838–6855, 2020.

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.

Ievgen Redko, Nicolas Courty, Rémi Flamary, and Devis Tuia. Optimal transport for multi-source domain adaptation under target shift. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 849–858. PMLR, 2019.

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.

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.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3723–3732, 2018.

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

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.

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.

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.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in artificial intelligence*, pp. 433–453. PMLR, 2020.

Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 267–273, 2003.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pp. 7404–7413. PMLR, 2019.

A THEORETICAL DETAILS

A.1 DIFFERENTIATION

In this section, we discuss the derivatives of (i) $\hat{P} \mapsto W_c(\hat{P}, \hat{Q})$, (ii) $\alpha \mapsto \mathcal{B}(\alpha; \mathcal{P})$, and (iii) $\mathcal{P} \mapsto \mathcal{B}(\alpha; \mathcal{P})$. Since we parametrize empirical distributions by their support, taking derivatives of distribution-valued functionals (e.g., the Wasserstein distance) is equivalent to differentiating w.r.t. the support of the free distribution (e.g., $\hat{P}$).

Considering the Wasserstein distance, one has two alternatives: (i) using the Sinkhorn algorithm (Cuturi (2013)) for calculating $\pi_\epsilon^\star \approx \pi^\star$, where ϵ is a parameter that controls the amount of entropic regularization. Since the Sinkhorn algorithm relies on differentiable operations, one can backpropagate through its iterations. (ii) Differentiating through the *min/inf* using the Envelope or Danskin's Theorem Afriat (1971); Bertsekas (1997) at optimality.

The Sinkhorn algorithm has mainly two drawbacks. First, it is unstable numerically for low regularization values, leading to problematic optimization. As remarked by Xie et al. (2020), a possible workaround would be using stabilized versions of the algorithm, but these come with costly exponential/logarithmic operations. The second issue is that the computed OT plans are no longer sparse, which may lead to degenerated distributions (e.g., take the extreme case when $\epsilon \to \infty$). For those reasons, we decide to use exact OT between mini-batches.

A second approach is using the Envelope theorem, which, as advocated by Feydy et al. (2019), can be faster than differentiating through Sinkhorn's iterations. For completeness, we state the Envelope theorem in the following,

Theorem A.1. (Envelope Theorem Bertsekas (1997)) Let $Z \subset \mathbb{R}^m$ be a compact set and let $\phi : \mathbb{R}^n \times Z \to \mathbb{R}$ be continuous and such that $\phi(\cdot, \mathbf{z}) : \mathbb{R}^n \to \mathbb{R}$ is convex for each $\mathbf{z} \in Z$. The function $f : \mathbb{R}^n \to \mathbb{R}$ given by,

$$f(\mathbf{x}) = \max_{\mathbf{z} \in Z} \phi(\mathbf{x}, \mathbf{z}),$$

is convex. When $Z(\mathbf{x}) = \{\bar{\mathbf{z}} : \phi(\mathbf{x}, \bar{\mathbf{z}}) = \max_{\mathbf{z}} \phi(\mathbf{x}, \mathbf{z})\}$ consists on an unique point $\bar{\mathbf{z}}$, then

$$\frac{\partial f}{\partial x_i} = \frac{\partial \phi(\mathbf{x}, \bar{\mathbf{z}})}{\partial x_i}. \tag{15}$$

The application of this theorem for OT is straightforward. In this case $Z = U(\mathbf{a}, \mathbf{b}) \subset \mathbb{R}^{m = n_P \times n_Q}$ which is compact. Naturally, the variable $\mathbf{z} = \pi$, whereas $\phi(\mathbf{C}, \pi) = \langle \mathbf{C}, \pi \rangle_F$. This is a linear function of $\mathbf{C}$, hence convex. Finally, the uniqueness of $\pi^\star$ depends on the regularity of the ground-cost. This property is particularly true for the Euclidean distance. In this case, the Wasserstein distance's derivative reads as,

$$W_2(\hat{P}, \hat{Q}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \pi_{i,j}^\star \|\mathbf{x}_i^{(P)} - \mathbf{x}_j^{(Q)}\|_2^2 \to \frac{\partial W_2}{\partial \mathbf{X}^{(P)}}(\hat{P}, \hat{Q}) = \frac{2}{n_P} \mathbf{X}^{(P)} - 2(\pi^\star)^T \mathbf{X}^{(Q)},$$

Here, it is essential to note that due to equation 15, to be able to evaluate $\partial W_2 / \partial \mathbf{x}_i^{(P)}$, one needs to compute $\pi^\star = \text{OT}(\mathbf{u}_n, \mathbf{u}_m, \mathbf{C})$ at optimality. This calculation poses a problem for the Sinkhorn algorithm since the number of iterations needed to converge is inversely proportional to ϵ. Thus, in some cases, it is more efficient to compute exact OT.

Now, we may apply the Envelope theorem to the iterations of Álvarez-Esteban et al. (2016) in equation 6. Let π_k be the OT plan between $\mathbf{X}^{(P_k)}$ and $\mathbf{X}^{(B)}$ at convergence. The barycentric operator is thus,

$$\mathbf{X}^{(B)} = \mathcal{B}(\boldsymbol{\alpha}; \mathcal{P}) = \sum_k \alpha_k \pi_k^T \mathbf{X}^{(P_k)},$$

which is linear w.r.t. α_k and $\mathbf{x}_i^{(P_k)}$. The derivatives are, therefore,

$$\frac{\partial \mathbf{X}^{(B)}}{\partial \alpha_k} = \pi_k \mathbf{X}^{(P_k)} \in \mathbb{R}^{n \times d} \text{ and } \frac{\partial \mathbf{x}_j^{(B)}}{\partial \mathbf{x}_i^{(P_k)}} = \alpha_k \text{diag}((\pi_k)_{i,j}) \in \mathbb{R}^{d \times d}.$$

A.2 WASSERSTEIN BARYCENTERS WITH FIXED-POINT ITERATIONS

In previous work (e.g., Montesuma & Mboula (2021b) and Montesuma & Mboula (2021a)), authors have used the iterations of Cuturi & Doucet (2014) for calculating the support $\mathbf{X}^{(B)}$ of Wasserstein barycenters. A similar approach is using the fixed-point approach of Álvarez-Esteban et al. (2016). As follows, let $T_{k,it}(\mathbf{x}) = n\pi_k^T \mathbf{X}^{(B^{(it)})}$, where π_k is the OT plan between $\mathbf{X}^{(P_k)}$ and $\mathbf{X}^{(B^{(it)})}$. Moreover, let $T_{it}(\mathbf{x}) = \sum_{k=1}^K \alpha_k T_{k,it}(\mathbf{x})$. Álvarez-Esteban et al. (2016) introduces the following mapping,

$$\psi(\hat{P}) = T_{it,\sharp}\hat{P} = \frac{1}{n}\sum_{i=1}^{n} \delta_{T_{it}(\mathbf{x})}.$$

In this setting, Álvarez-Esteban et al. (2016) proves that the Wasserstein barycenter is a fixed-point solution o ψ, namely, $\psi(\hat{B}) = \hat{B}$. Their procedure is then equivalent to $\hat{B}^{(it+1)} = \psi(\hat{B}^{(it)})$. This iteration corresponds exactly to those in Equation 6. These are summarized in Algorithm 2.

In section 3, Equation 7 introduced a supervised ground-cost between pairs of samples $(\mathbf{x}_i^{(P)}, y_i^{(P)})$ and $(\mathbf{x}_i^{(Q)}, y_i^{(Q)})$, $C_{i,j} = \|\mathbf{x}_i^{(P)} - \mathbf{x}_j^{(Q)}\|_2^2 + \beta\|\mathbf{Y}_i^{(P)} - \mathbf{Y}_j^{(Q)}\|_2^2$, where $\beta \geq 0$ defines how costly it is to transport samples from different classes. For one-hot encoded vectors $\mathbf{Y} \in \mathbb{R}^{n \times n_c}$, the Euclidean distance is equivalent to the 0-1 loss, $\|\mathbf{Y}_i^{(P)} - \mathbf{Y}_j^{(Q)}\|_2^2 = \delta(y_i^{(P)} - y_j^{(Q)})$. This makes a direct link between our proposed ground-cost, and the semi-supervised regularization proposed by Courty et al. (2016).

In addition, using an Euclidean cost for the labels allows us to interpolate distributions $P(X, Y)$ and $Q(X, Y)$, with respect to features and labels. Indeed, taking derivatives as in the previous section,

$$W_c(\hat{P}, \hat{Q}) = \sum_{i=1}^{n}\sum_{j=1}^{m} \pi_{i,j}^{\star}(\|\mathbf{x}_i^{(P)} - \mathbf{x}_j^{(Q)}\|_2^2 + \beta\|\mathbf{Y}_i^{(P)} - \mathbf{Y}_j^{(Q)}\|_2^2),$$

$$\rightarrow {}^{\partial W_c}/_{\partial \mathbf{X}^{(P)}}(\hat{P}, \hat{Q}) = \frac{2}{n_P}\mathbf{X}^{(P)} - 2(\pi^{\star})^T \mathbf{X}^{(Q)}, \tag{16}$$

$$\rightarrow {}^{\partial W_c}/_{\partial \mathbf{Y}^{(P)}}(\hat{P}, \hat{Q}) = \frac{2\beta}{n_P}\mathbf{Y}^{(P)} - 2\beta(\pi^{\star})^T \mathbf{Y}^{(Q)}. \tag{17}$$

Equating the derivatives in 16 and 17 to 0 yields, respectively, the barycentric mapping of Courty et al. (2016) and the label propagation formula of Redko et al. (2019). This allows us to reformulate the fixed-point iterations for joint distributions. Algorithms 2 and 3 show the proposed iterations for barycenters of features and features-labels distributions respectively.

Algorithm 2 Free-Support Wasserstein Barycenter of Unlabeled Distributions

Require: $\{\mathbf{X}^{(P_k)}\}_{k=1}^K$, $\boldsymbol{\alpha} \in \Delta_K$, $\tau > 0$, n.
1: $\mathbf{X}^{(B^{(0)})} = \textit{initialize}(\{\mathbf{X}^{(P_k)}\}_{k=1}^K)$
2: **while** $\|\mathbf{X}^{(B^{(it)})} - \mathbf{X}^{(B^{(it-1)})}\|_F \leq \tau$ has not converged **do**
3: **for** $k = 1, \cdots K$ **do**
4: $(\mathbf{C}_k)_{i,j} = \|\mathbf{x}_i^{(P_k)} - \mathbf{x}_j^{(B^{(it)})}\|_2^2$.
5: $\pi_k = \text{OT}(\mathbf{u}_{n_k}, \mathbf{u}_n, \mathbf{C}_k)$
6: **end for**
7: $\mathbf{X}^{(B^{(it+1)})} = n\sum_{k=1}^K \alpha_k \pi_k^T \mathbf{X}^{(P_k)}$
8: **end while**
Ensure: Barycenter support samples $\mathbf{X}^{(B)}$.

Algorithm 3 Free-Support Wasserstein Barycenter of Labeled Distributions

Require: $\{\mathbf{X}^{(P_k)}, \mathbf{Y}^{(P_k)}\}_{k=1}^K$, $\boldsymbol{\alpha} \in \Delta_K$, $\tau > 0$, n, $\beta \geq 0$.
1: $\mathbf{X}^{(B^{(0)})}, \mathbf{Y}^{(B^{(0)})} = \textit{initialize}(\{\mathbf{X}^{(P_k)}, \mathbf{Y}^{(P_k)}\}_{k=1}^K)$
2: **while** $\|\mathbf{X}^{(B^{(it)})} - \mathbf{X}^{(B^{(it-1)})}\|_F \leq \tau$ has not converged **do**
3: **for** $k = 1, \cdots K$ **do**
4: $(\mathbf{C}_k)_{i,j} = \|\mathbf{x}_i^{(P_k)} - \mathbf{x}_j^{(B^{(it)})}\|_2^2 + \beta\|\mathbf{Y}_i^{(P_k)} - \mathbf{Y}^{(B_j^{(it)})}\|_2^2$.
5: $\pi_k = \text{OT}(\mathbf{u}_{n_k}, \mathbf{u}_n, \mathbf{C}_k)$
6: **end for**
7: $\mathbf{X}^{(B^{(it+1)})} = n\sum_{k=1}^K \alpha_k \pi_k^T \mathbf{X}^{(P_k)}$
8: $\mathbf{Y}^{(B^{(it+1)})} = n\sum_{k=1}^K \alpha_k \pi_k^T \mathbf{Y}^{(P_k)}$
9: **end while**
Ensure: Barycenter support samples $\mathbf{X}^{(B)}$ and labels $\mathbf{Y}^{(B)}$.

Remark 1: in Algorithms 2 and 3 we use the routine *initialize* to get the initial barycenter support $\mathbf{X}^{(B^{(0)})}$, $\mathbf{Y}^{(B^{(0)})}$. In our experiments we sub-sample n points from the concatenated matrices $\{\mathbf{X}^{(P_k)}, \mathbf{Y}^{(P_k)}\}_{k=1}^K$. Alternatively, one could do as Montesuma & Mboula (2021a;b) and initialize the barycenter support randomly.

A.3 APPROXIMATION PROPERTIES OF WED

Proof of Theorem 3.1: our proof is done in 4 steps: (i) $\text{WED}(\hat{Q}_1, \hat{Q}_2) \geq 0$, $\forall \hat{Q}_1, \hat{Q}_2$, (ii) $\text{WED}(\hat{Q}_1, \hat{Q}_1) = 0$, $\forall \hat{Q}_1$, (iii) $\text{WED}(\hat{Q}_1, \hat{Q}_2) = \text{WED}(\hat{Q}_2, \hat{Q}_1)$ (iv) $\text{WED}(\hat{Q}_1, \hat{Q}_3) \leq \text{WED}(\hat{Q}_1, \hat{Q}_2) + \text{WED}(\hat{Q}_2, \hat{Q}_3)$.

(i) Since π_{k_1,k_2} and $W_c(\hat{P}_{k_1}, \hat{P}_{k_2})$ are non-negative, $\forall \hat{Q}_1, \hat{Q}_2$ one has $\text{WED}(\hat{Q}_1, \hat{Q}_2) \geq 0$.

(ii) First, note that $W_c(\hat{Q}_1, \hat{Q}_1) = 0$. Now, since equation 11 is a convex optimization problem, to each $\hat{Q}_1$ one has a single solution $\boldsymbol{\alpha}_1$. Hence, $\mathbf{I}_K \in U(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_1)$, and $\text{WED}(\hat{Q}_1, \hat{Q}_1) = 0$.

(iii) If $\pi \in U(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$, then $\pi^T \in U(\boldsymbol{\alpha}_2, \boldsymbol{\alpha}_1)$. Due to commutativity of the summation, $\text{WED}(\hat{Q}_1, \hat{Q}_2) = \text{WED}(\hat{Q}_2, \hat{Q}_1)$.

(iv) Let $\pi^{(1)}, \pi^{(2)}$ and $\pi^{(3)}$ be the solutions to $\text{WED}(\hat{Q}_1, \hat{Q}_3)$, $\text{WED}(\hat{Q}_1, \hat{Q}_2)$ and $\text{WED}(\hat{Q}_2, \hat{Q}_3)$, respectively. In this case,

$$\begin{aligned}\text{WED}(\hat{Q}_1, \hat{Q}_3) &= \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \pi^{(1)}_{k_1,k_2} W_c(\hat{P}_{k_1}, \hat{P}_{k_2}), \\ &\leq \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \pi^{(c)}_{k_1,k_2} W_c(\hat{P}_{k_1}, \hat{P}_{k_2}),\end{aligned}$$

since $\pi^{(2)}$ is sub-optimal for the pair $(\hat{Q}_1, \hat{Q}_3)$. Now,

$$\begin{aligned}\text{WED}(\hat{Q}_1, \hat{Q}_3) &\leq \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \pi^{(2)}_{k_1,k_2} W_c(\hat{P}_{k_1}, \hat{P}_{k_2}), \\ &\leq \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} (\pi^{(2)}_{k_1,k_2} + \pi^{(3)}_{k_1,k_2}) W_c(\hat{P}_{k_1}, \hat{P}_{k_2}) \\ &= \text{WED}(\hat{Q}_1, \hat{Q}_2) + \text{WED}(\hat{Q}_2, \hat{Q}_3).\end{aligned}$$

since $\pi^{(3)}$ is non-negative.

Proof of Theorem 3.2: our proof relies on the successive application of the triangle inequality for the Wasserstein distance. Let $\pi \in U(\alpha_1, \alpha_2)$ be the minimizer in equation 10. Then,

$$\begin{aligned}\text{WED}(\hat{Q}_1, \hat{Q}_2) &= \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \pi_{k_1,k_2} W_c(\hat{P}_{k_1}, \hat{P}_{k_2}), \\ &\leq \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \pi_{k_1,k_2} (W_c(\hat{P}_{k_1}, \hat{Q}_2) + W_c(\hat{P}_{k_2}, \hat{Q}_2)), \\ &\leq \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \pi_{k_1,k_2} (W_c(\hat{P}_{k_1}, \hat{Q}_1) + W_c(\hat{Q}_1, \hat{Q}_2) + W_c(\hat{P}_{k_2}, \hat{Q}_2)).\end{aligned}$$

We now break this sum in 3 parts. First, let us consider $W_2(\hat{P}_{k_1}, \hat{Q}_1)$:

$$\begin{aligned}\sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \pi_{k_1,k_2} W_c(\hat{P}_{k_1}, \hat{Q}_1) &= \sum_{k_1=1}^{K} W_c(\hat{P}_{k_1}, \hat{Q}_1) \Big(\sum_{k_2=1}^{K} \pi_{k_1,k_2} \Big), \\ &= \sum_{k_1=1}^{K} \alpha_{1,k_1} W_c(\hat{P}_{k_1}, \hat{Q}_1).\end{aligned}$$

Since $\hat{B}_1$ is the minimizer of the r.h.s of this last equation,

$$\sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \pi_{k_1,k_2} W_c(\hat{P}_{k_1}, \hat{Q}_1) \leq \sum_{k=1}^{K} \alpha_{1,k} W_c(\hat{P}_k, \hat{B}_1).$$

The exact same process can be repeated for the term involving $W_2(\hat{P}_{k_2}, \hat{Q}_2)$, that is,

$$\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}\pi_{k_1,k_2}W_c(\hat{P}_{k_2}, \hat{Q}_2) \leq \sum_{k=1}^{K}\alpha_{1,k}W_c(\hat{P}_k, \hat{B}_2).$$

The final term follows from the fact that $W_2(\hat{Q}_1, \hat{Q}_2)$ is independent of both k_1 nor k_2. Therefore,

$$\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}\pi_{k_1,k_2}W_c(\hat{Q}_1, \hat{Q}_2) = W_c(\hat{Q}_1, \hat{Q}_2)\bigg(\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}\pi_{k_1,k_2}\bigg) = W_c(\hat{Q}_1, \hat{Q}_2).$$

A.4 Theoretical Bounds for Domain Adaptation

In what follows, we consider the theoretical results of Ben-David et al. (2010) and Redko et al. (2017) for giving the theoretical guarantees of both DaDiL-E and DaDiL-R. For completeness, we re-state Lemma 1 and Theorem 1 of Redko et al. (2017), and prove Theorem 3.3.

Lemma A.1. (Due to Redko et al. (2017)) Let P and Q be two probability distributions over $\mathbb{R}^d$. Assume that the cost function $c(\mathbf{x}^{(P)}, \mathbf{x}^{(Q)}) = \|\phi(\mathbf{x}^{(P)}) - \phi(\mathbf{x}^{(Q)})\|_{\mathcal{H}_k}$, where $\mathcal{H}_k$ is a reproducing kernel Hilbert space equipped with kernel $\Phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ induced by $\phi : \mathbb{R}^d \to \mathcal{H}_k$ and $\Phi(\mathbf{x}^{(P)}, \mathbf{x}^{(Q)}) = \langle\phi(\mathbf{x}^{(P)}), \phi(\mathbf{x}^{(Q)})\rangle_{\mathcal{H}_\Phi}$. Assume that the loss function $l_{h,h_0} : \mathbf{x} \mapsto l(h(\mathbf{x}), h_0(\mathbf{x}))$ is convex, symmetric, bounded, obeys the triangular inequality and has the parametric form $|h(\mathbf{x}) - h_0(\mathbf{x})|^q$ for some $q > 0$. Assume that the kernel $\Phi \in \mathcal{H}_\Phi$ is square-root integrable w.r.t. both P and Q and $0 \leq \Phi(\mathbf{x}^{(P)}, \mathbf{x}^{(Q)}) \leq M, \forall \mathbf{x}^{(P)}, \mathbf{x}^{(Q)} \in \mathbb{R}^d$. Then the following holds,

$$\mathcal{R}_Q(h, h') \leq \mathcal{R}_P(h, h') + W_1(P, Q). \tag{18}$$

Lemma A.1 bounds the risk (recall equation 2) of h with respect to h' under distribution Q by the risk under distribution P, plus a term depending on the distance between those 2 distributions. Intuitively, if P is close to Q, its samples are similar and thus the risk are similar. We are now interested in acquiring bounds for the empirical risks $\hat{\mathcal{R}}_P$ and $\hat{\mathcal{R}}_Q$ in terms of $W_1(\hat{P}, \hat{Q})$, which are quantities we can estimate from data. We start by stating Theorem 1.1 of Bolley et al. (2007),

Lemma A.2. (Due to Bolley et al. (2007)) Let P be a probability distribution over $\mathbb{R}^d$, so that for some $\alpha > 0$ we have that $\int_{\mathbb{R}^d} e^{\alpha||\mathbf{x}||^2}dP < \infty$ and $\hat{P}$ be its associated empirical approximation with support $\{\mathbf{x}_i^{(P)}\}_{i=1}^n$ drawn independently from P. Then, for any $d' > d$ and $\zeta < \sqrt{2}$ there is a constant n_0 depending on d' and some square exponential moment of P such that for any $\epsilon > 0$ and $n \geq n_0\max(\epsilon^{-(d'+2)}, 1)$,

$$\mathbb{P}[W_1(\hat{P}, P) > \epsilon] \leq \exp\bigg(-\frac{\zeta}{2}n\epsilon^2\bigg),$$

where d' and ζ' can be calculated explicitly.

Lemma A.2 states the conditions for which $\hat{P}$ and P are close in the sense of Wasserstein. This last bound is on the form $\mathbb{P}[quantity > \epsilon] < \delta$, that is, with high probability $quantity \leq \epsilon$. These types of bounds are ubiquitous in the theoretical analysis of learning algorithms. We can express ϵ explicitly in terms of δ,

$$\epsilon = \sqrt{\frac{2}{n\zeta}\log(\tfrac{1}{\delta})}, \tag{19}$$

which will be useful in the following discussion. These results allowed Redko et al. (2017) to provide theoretical guarantees for the OTDA framework of Courty et al. (2016),

Theorem A.2. (Due to Redko et al. (2017)) Under the assumptions of Lemma A.1, let $\mathbf{X}^{(P)}$ and $\mathbf{X}^{(Q)}$ be 2 samples of size n_P and n_Q, drawn i.i.d. from P and Q. Let $\hat{P}$ and $\hat{Q}$ be the respective empirical approximations. Then for any $d' > d$ and $\zeta < \sqrt{2}$ there exists some constant N_0, depending on d' such that for any $\delta > 0$ and $\min(n_P, n_Q) \geq N_0\max(\delta^{-d'+2}, 1)$ with probability at least $1 - \delta$ for all $h \in \mathcal{H}$, then,

$$\mathcal{R}_Q(h) \leq \mathcal{R}_P(h) + W_1(\hat{P}, \hat{Q}) + \sqrt{2^{\log(1/\delta)}/\zeta}\bigg(\sqrt{1/n_P} + \sqrt{1/n_Q}\bigg) + \lambda.$$

This last theorem effectively states that, besides constant terms that depend on the number of samples n_P and n_Q, there are 2 factors that determine whether the risk under Q is similar to that under P: (i) the distance in distribution between $\hat{P}$ and $\hat{Q}$, (ii) how well can a classifier in $\mathcal{H}$ work on both domains. In practice, what OTDA does is minimizing $W_1(\hat{P}, \hat{Q})$ while keeping λ constant (e.g. by enforcing class-sparsity).

Now, let us discuss how these concepts apply to DaDiL-R. Note that our strategy consists on learning with samples obtained by the barycentric distribution $\hat{B}_T = \mathcal{B}(\boldsymbol{\alpha}_T; \mathcal{P})$, since it approximates the target distribution $\hat{Q}_T$. In this sense,

$$\mathcal{R}_{Q_T}(h) \leq \mathcal{R}_{B_T}(h) + W_1(\hat{Q}_T, \hat{B}_T) + \sqrt{2^{\log(1/\delta)}/\zeta}\left(\sqrt{1/n_T} + \sqrt{1/n}\right) + \lambda.$$

In this last equation, the term $W_1(\hat{Q}_T, \hat{B}_T)$ is the reconstruction error for the target domain, which we directly minimize in our objective function (equation 9). The remaining term is λ, which we cannot directly optimise since target domain labels are not available during training. Nonetheless our algorithm leverages label information from source domains. If we suppose that their class structure is similar to that of the target domain, the regularization term 7 ensures that λ remains bounded, since transport plans will not mix classes.

We now focus on Theorem 3.3, which presents a theoretical guarantee for DaDiL-E.

Proof of Theorem 3.3. This proof relies on the triangle inequality for the risk. Let $h^\star_{T,k} = \text{argmin}_{h\in\mathcal{H}} R_{Q_T}(h) + R_{P_k}(h)$ and $h_{\boldsymbol{\alpha}} = \sum_k \alpha_k R_{P_k}(h^\star_k)$ for $h^\star_k = \text{argmin}_{h\in\mathcal{H}} R_{P_k}(h)$. Then,

$$\begin{aligned}
\mathcal{R}_{Q_T}(h_{\boldsymbol{\alpha}}) &\leq \mathcal{R}_{Q_T}(h^\star_{T,k}) + \mathcal{R}_{Q_T}(h^\star_{T,k}, h_{\boldsymbol{\alpha}}),\\
&= (\mathcal{R}_{Q_T}(h^\star_{T,k}, h_{\boldsymbol{\alpha}}) - \mathcal{R}_{P_k}(h^\star_{T,k}, h_{\boldsymbol{\alpha}})) + (\mathcal{R}_{Q_T}(h^\star_{T,k}) + \mathcal{R}_{P_k}(h^\star_{T,k}, h_{\boldsymbol{\alpha}})),\\
&\leq \underbrace{(\mathcal{R}_{Q_T}(h^\star_k, h_{\boldsymbol{\alpha}}) - \mathcal{R}_{P_k}(h^\star_k, h_{\boldsymbol{\alpha}}))}_{\leq W_1(P_k, Q_T) \text{ (Lemma A.1)}} + \underbrace{(\mathcal{R}_{Q_T}(h^\star_{T,k}) + \mathcal{R}_{P_k}(h^\star_{T,k}))}_{=\lambda_k} + \mathcal{R}_{P_k}(h_{\boldsymbol{\alpha}}),\\
&\leq \mathcal{R}_{P_k}(h_{\boldsymbol{\alpha}}) + W_1(P_k, Q_T) + \lambda_k.
\end{aligned}$$

Summing over k, weighted by α,

$$\begin{aligned}
\mathcal{R}_{Q_T}(h_{\boldsymbol{\alpha}}) = \sum_k \alpha_k \mathcal{R}_{Q_T}(h_{\boldsymbol{\alpha}}) &\leq \sum_k \alpha_k R_{P_k}(h_{\boldsymbol{\alpha}}) + \sum_k \alpha_k (W_1(P_k, Q_T) + \lambda_k),\\
&= R_\alpha(h_{\boldsymbol{\alpha}}) + \sum_k \alpha_k (W_1(P_k, Q_T) + \lambda_k),\\
&\leq R_\alpha(h_{\boldsymbol{\alpha}}) + \sum_k \alpha_k (W_1(\hat{P}_k, \hat{Q}_T) + \lambda_k + c_1),
\end{aligned}$$

This theorem is similar to previous theoretical guarantees for MSDA, such as those of Ben-David et al. (2010), Redko et al. (2017), and WBT$_{reg}$ (Montesuma & Mboula (2021a)). In particular, DaDiL-E uses $\boldsymbol{\alpha}_T$ for weighting $h^\star_k$. Furthermore this choice of $\boldsymbol{\alpha}$ minimizes the r.h.s. of the last equation.

B Technical Details

B.1 Data Preparation

Caltech-Office 10: we use this dataset for shallow DA. As follows, we use the experimental setting of Montesuma & Mboula (2021a), namely, the 5 fold cross-validation partitions and the features (DeCAF 7th layer activations).

Office 31: we use this dataset for deep DA. To that end, we download the raw images from its public repository[3]. We train the feature extractor ourselves, which consists on a ResNet50 (He et al. (2016)). The preprocessing steps taken are: (i) resizing each image to $(224, 224, 3)$, and (ii) applying `tf.keras.applications.resnet50.preprocess_input` function. We initialize its

[3] `https://faculty.cc.gatech.edu/~judy/domainadapt/`

parameters with the weights trained on ImageNet, then fine-tune for 51 epochs on each combination of source domains (e.g., A, D). Fine-tuning was performed using the ADAPT library of de Mathelin et al. (2021). We understand each epoch as a full pass through the entire dataset. We then use the fine-tuned network to extract features (vectors $\mathbf{x} \in \mathbb{R}^{2048}$) from the target domain only.

Remark 2: we use a ResNet backbone since we were not able to reproduce previous results on the standard AlexNet backbone (e.g., Nguyen et al. (2021)).

Remark 3: we give the specifics for the Caltech-Office and Office 31 datasets in Table 2. As follows, we train our algorithms with all source domain data (train and test), and train target domain data. For instance, when training a specific algorithm on Office 31 dataset with the setting $A, D \to W$, at training time the algorithm has 2817 labeled samples from A, 498 labeled samples from D and 636 unlabeled samples from W available. For evaluation, we use the target test set. In the context of our example, 159 unlabeled samples from W.

Table 2: Details about the datasets considered for domain adaptation.

Dataset	# Classes	Domain	# Training Samples	# Test Samples
Caltech-Office 10	10	Amazon	748	210
		dSLR	108	49
		Webcam	224	71
		Caltech	956	224
Office 31	31	Amazon	2253	564
		dSLR	398	100
		Webcam	636	159

Artworks dataset: we select a sub-set of 50 artworks from the Best Artworks of All Time[4], from authors Monet, Delacroix, Magritte, Caravaggio and Van Gogh, with 10 images from each. The artworks were selected so that they group around 3 major color palettes (red, blue and somber).

B.2 Additional Experiments

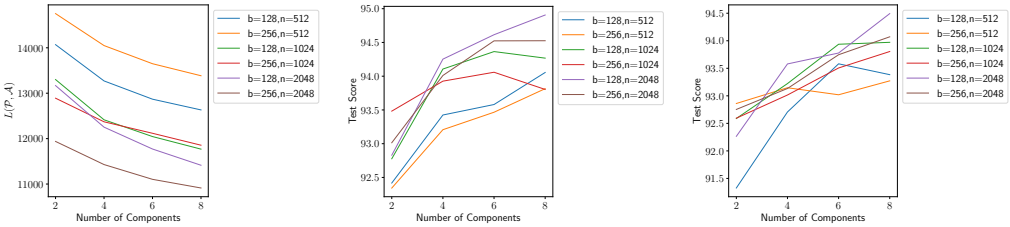
We study how the number of components K, the batch size b, and the number of samples in the atoms support n influence the performance of DiL. Our results are summarized in Figures 8 and 9. We want to verify whether the DiL loss $L(\mathcal{P}, \mathcal{A})$ is a good predictor for the test score.

On one hand, for the Caltech-Office dataset, Figure 8 shows that increasing the number of samples and batch sizes globally leads to a better approximation of the distributions manifold, as $L(\mathcal{P}, \mathcal{A})$ shrinks. This, however, is not necessarily true for the test score, as $b = 128$ and $n = 2048$ leads to better adaptation results. Figures 8 (d) and (e) shows that the DiL is a good proxy for DaDiL-E and DaDiL-R, as it is strongly correlated with the test score. We further remark that $L(\mathcal{P}, \mathcal{A})$ is more correlated with the test score of DaDiL-R, since this method directly relies on the goodness of reconstructions.

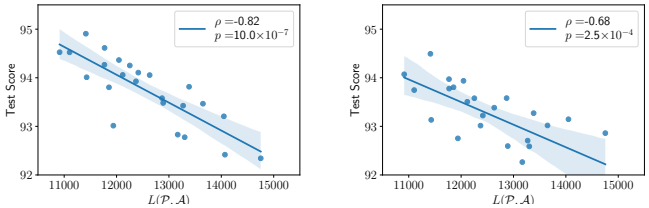
On the other hand, for the Office 31 dataset, Figure 9 (a) shows that the dependence on these parameters is much more complex, showing that in general more samples or components do not lead to better approximations. Moreover, Figures 9 (d) and (e) show that $L(\mathcal{P}, \mathcal{A})$ is not correlated with neither DaDiL-R nor DaDiL-E. We believe that since this dataset encompasses more classes than the Caltech-Office, one needs more atoms to model the particularities of each domain.

We now focus our attention on t-SNE embeddings of reconstructions $\hat{B}_T$. Figures 10 and 11 shows the embeddings for the Caltech-Office and Office 31 datasets. For this first dataset, our reconstructions are correctly aligned with the dataset's classes, indicating that the atoms have correctly captured the characteristics of the domains. For the second dataset, as the number of classes is large, we only show a comparison between original and reconstructed samples.

[4] https://www.kaggle.com/datasets/ikarus777/best-artworks-of-all-time



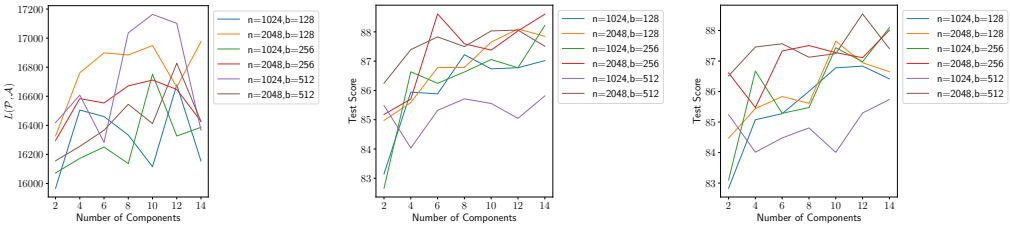
(a) Loss. (b) DaDiL-R validation accuracy. (c) DaDiL-E validation accuracy.



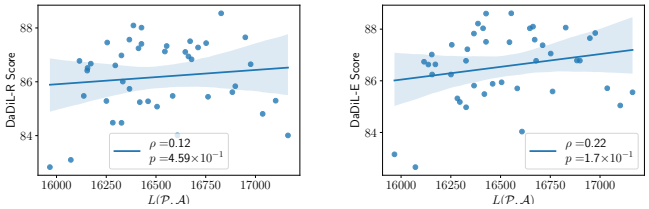
(d) Correlation between DiL loss and DaDiL-R classification performance.

(e) Correlation between DiL loss and DaDiL-E classification performance.

Figure 8: Study of how the batch size, number of samples and number of components impact DiL, DaDiL-R and E performances on Caltech-Office 10 dataset.



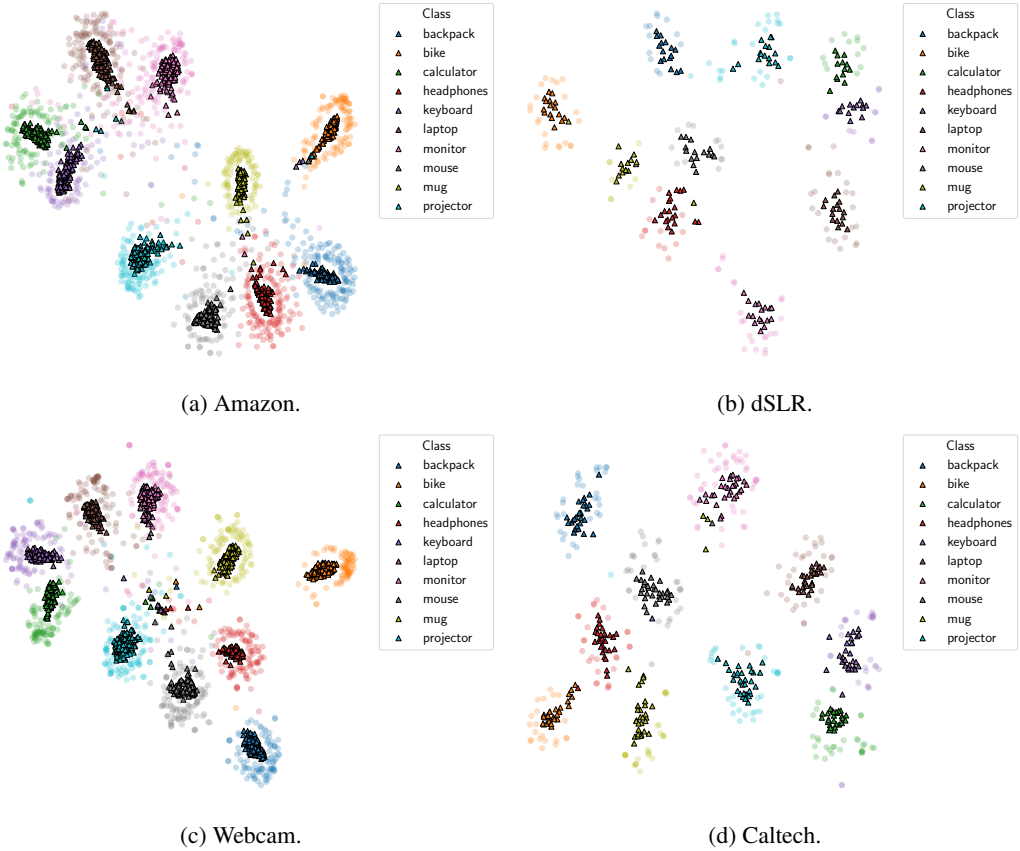
(a) Loss. (b) DaDiL-R validation accuracy. (c) DaDiL-E validation accuracy.



(d) Correlation between DiL loss and DaDiL-R classification performance.

(e) Correlation between DiL loss and DaDiL-E classification performance.

Figure 9: Study of how the batch size, number of samples and number of components impact DiL, DaDiL-R and E performances on Office 31 dataset.



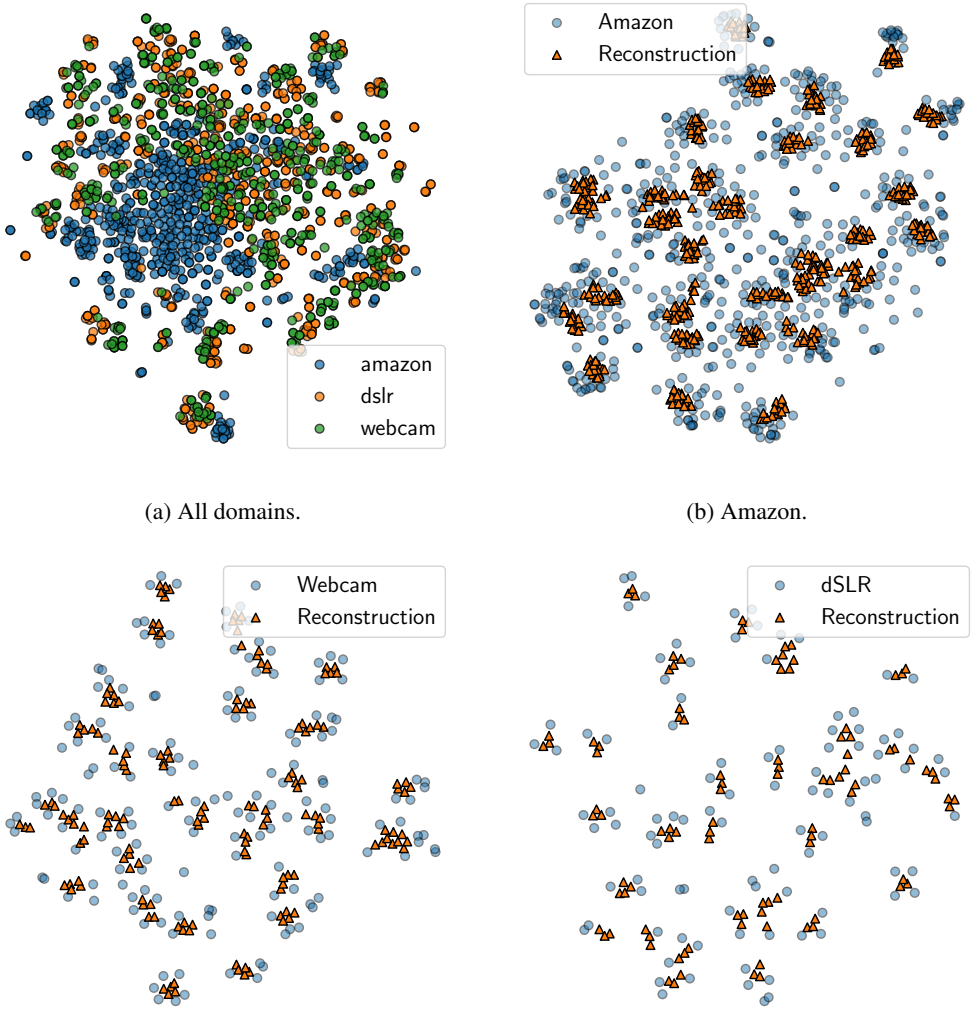
(a) Amazon.

(b) dSLR.

(c) Webcam.

(d) Caltech.

Figure 10: t-SNE visualization of DaDiL-R reconstructions of each target domain. Semi-transparent circles correspond to points in the target domain, whereas triangles correspond to points in $\hat{B}_T$ support.



(a) All domains.

(b) Amazon.

(c) Webcam.

(d) dSLR.

Figure 11: t-SNE visualization of DaDiL-R reconstructions of each domain in the Office 31 dataset. Semi-transparent circles correspond to points in the target domain, whereas triangles correspond to points in $\hat{B}_T$ support.