

Computing Wasserstein Barycenters through Gradient Flows

Eduardo Fernandes Montesuma

Yassir Bendou

Sigma Nova, Paris, France

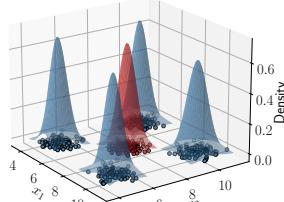
Mike Gartrell

Abstract

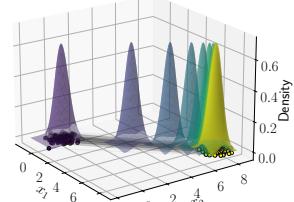
Wasserstein barycenters provide a powerful tool for aggregating probability measures, while leveraging the geometry of their ambient space. Existing discrete methods suffer from poor scalability, as they require access to the complete set of samples from input measures. We address this issue by recasting the original barycenter problem as a gradient flow in the Wasserstein space. Our approach offers two advantages. First, we achieve scalability by sampling mini-batches from the input measures. Second, we incorporate functionals over probability measures, which regularize the barycenter problem through internal, potential, and interaction energies. We present two algorithms for empirical and Gaussian mixture measures, providing convergence guarantees under the Polyak-Lojasiewicz inequality. Experimental validation on toy datasets and domain adaptation benchmarks show that our methods outperform previous discrete and neural net-based methods for computing Wasserstein barycenters.

1 Introduction

Wasserstein barycenters [1] constitute a building block in the analysis of probability measures under Optimal Transport (OT) [2]. This mathematical object extends the notion of the barycenter of points to Wasserstein spaces, thus defining a notion of the average of probability measures. Since it relies on the Wasserstein metric, these barycenters capture the geometry of the underlying space over which the probability measures are defined [3]. As such, they contributed to several areas



(a) Static barycenter.



(b) Barycenter flow.

Figure 1: In (a), we show the usual static notion of the Wasserstein barycenter (in red), which minimizes the sum of distances to the input measures (in blue). In (b), we show our notion of barycenter as a gradient flow, flowing an initial measure P_0 (purple) to the barycenter P^* (yellow) of the input measures.

of machine learning, including model averaging [4], ensembling [5], data augmentation [6], distillation [7, 8], dictionary learning [9], domain adaptation [10, 11, 12], and Bayesian learning [13].

Existing methods for computing Wasserstein barycenters can be divided into three categories. The first category relies on discretization of the input measures and the barycentric measure through empirical measures. These algorithms may be subdivided into fixed-support [14, Algorithm 1], [15, 16], and free-support [14, Algorithm 2], [11].

The second category relies on parametric models for the input measures, such as Gaussian, or Gaussian mixture models. In the Gaussian case, these barycenters are known as Bures-Wasserstein barycenters [17, 18]. For Gaussian mixtures, [19] shows connections with multi-marginal OT [20] between the components of the input mixtures, while [12] shows that, for components with diagonal covariances, there is an iterative algorithm similar to [14, Algorithm 2].

The third category of methods for computing Wasserstein barycenters relies on neural nets and comes from the effort to *scale* OT to measures in high-dimensional spaces with a large number of samples. In this sense, [21] proposes parameterizing the barycentric measure through a specific kind of neural net ar-

chitecture, known as Input Convex Neural Networks (ICNNs) [22]. Furthermore, [23] proposes a bi-level adversarial learning approach that can be used for general ground costs, which is made more robust by semi-unbalanced OT [24]. More recently, [25] proposes computing barycenters with normalizing flows.

Current Wasserstein barycenter algorithms mostly focus on scalability with respect to the number of samples n , and dimensions d . In this sense, neural networks have a clear advantage over discrete methods, since they can operate at the level of mini-batches. There is another level of scalability, oftentimes overlooked, which is with respect the number of input measures K . So far, only [25] has considered experiments in this context. In this regard, methods such as [21, 26] are difficult to scale, as they require $\mathcal{O}(K)$ neural networks to compute the Wasserstein barycenter.

In this paper we propose a new perspective on discrete Wasserstein barycenters, through the lens of *Wasserstein Gradient Flows (WGFs)* [27, 28], which is an appealing framework for optimization in the space of probability measures. This new understanding of Wasserstein barycenters has several advantages. First, we can scale the discrete barycenter problem, by drawing mini-batches from the input measures. Second, we can scale with respect to the number of input measures, as the representation of the barycentric measure is independent of K . Third, based on the established literature on gradient flows, we can define *regularized* Wasserstein barycenters with suitable functionals on the space of probability measures, which were not possible with previous methods. Lastly, we can modify the underlying metric to integrate label information between samples, leading to barycenters that respect class structure and provide more accurate and semantically meaningful estimates. We show an overview of our approach in Figure 1. Our contributions are:

- We introduce a unified framework for computing Wasserstein barycenters based on *gradient flows*, which provides a principled and flexible approach to optimizing in the space of probability measures.
- We generalize algorithms from the state-of-the-art [14, 11, 12] under a common framework.
- We offer theoretical guarantees for the convergence of our algorithms, which also apply to previous works.
- We offer a new theoretical result for the mixture-Wasserstein distance between labeled Gaussian mixtures [19, 12].
- We conduct extensive experiments benchmarking existing methods, including neural network-based

Wasserstein barycenter algorithms [21, 23, 24, 25], on both toy datasets and domain adaptation¹ benchmarks, and show that our approach establishes a new state-of-the-art.

Paper organization. Section 2 presents the notation and preliminaries of our paper. Section 3 covers our proposed algorithms, including the theoretical results. Section 4 describes our experiments. Finally, section 5 concludes this paper.

2 Background

Throughout this paper, (Ω, d) is a metric space. We denote by P and Q , probability measures on $\mathcal{P}(\Omega)$, which is the space of measures over Ω . For $p \in [1, +\infty)$ we further denote by $\mathcal{P}_{p,ac}(\Omega)$, the sub-set of absolutely continuous measures w.r.t. the Lebesgue measure such that $\int_{\Omega} d(z, z_0)^p dP(z) < +\infty$. In our case, we have access to these measures through their samples, denoted $z^{(P)} \sim P$. The set Δ_K denotes the K -simplex, i.e., $y \in \Delta_n = \{y \in \mathbb{R}^n : \sum_{i=1}^n y_i = 1 \text{ and } y_i \geq 0 \forall i\}$. We denote functionals over $\mathcal{P}_{2,ac}$ with blackboard bold letters (e.g., \mathbb{F}), and their (Wasserstein) gradients by \mathbb{W} . We offer a short introduction in Section 3 in the Appendix.

2.1 Empirical Optimal Transport

In this section, given n i.i.d. samples from P , we can approximate a probability P *empirically* through,

$$\hat{P}(z) = \frac{1}{n} \sum_{i=1}^n \delta(z - z_i^{(P)}), \quad (1)$$

which gives a finite parametrization for P . Henceforth we use a hat to indicate empirical measures (e.g., \hat{P}).

OT [2] is a field of mathematics concerned with the transportation of mass at least effort. We refer readers to [29] for a computational treatment of the subject, and [30] for applications in machine learning. Let $P, Q \in \mathcal{P}(\Omega)$, and $c : \Omega^2 \rightarrow \mathbb{R}$ be a ground-cost. OT was originally founded by Monge [31]. In this formulation, we seek for a mapping $T : \Omega \rightarrow \Omega$ such that,

$$T_{P \rightarrow Q}^* = \arg \inf_{T \sharp P = Q} \int_{\Omega} c(z, T(z)) dP(z). \quad (2)$$

Alternatively, Kantorovich [32] proposed a formulation in terms of a transport plan $\gamma \in \Gamma(P, Q)$, where

¹In contrast with discrete methods, most experiments in neural network solvers involve generative modeling. We thus establish a bridge between these methods and domain adaptation

$\Gamma(P, Q)$ is the set of all joint measures with marginals P and Q . In this case,

$$\gamma^* = \arg \inf_{\gamma \in \Gamma(P, Q)} \int_{\Omega} \int_{\Omega} c(z, z') \gamma(z, z) \quad (3)$$

Given $p \in [1, +\infty)$, when $c(z, z') = d(z, z')^p$, the previous problems define the p -Wasserstein distance,

$$\mathbb{W}_p(P, Q)^p = \inf_{\gamma \in \Gamma(P, Q)} \int_{\Omega} \int_{\Omega} d(z, z')^p d\gamma(z, z'). \quad (4)$$

Equation 4 defines a metric in $\mathcal{P}_{p,ac}(\Omega)$. Conceptually, the Wasserstein distance *lifts* the metric d on Ω to \mathbb{W}_p on $\mathcal{P}_{p,ac}(\Omega)$. Based on this idea, we can define barycenters and gradient flows in the space of probability measures.

2.2 Gaussian Mixture Optimal Transport

In comparison with equation 1, we can approximate P through a Gaussian Mixture Model (GMM),

$$P(z) = \sum_{i=1}^n \pi_i^{(P)} \mathcal{N}(z | \mu_i^{(P)}, \Sigma_i^{(P)}), \quad (5)$$

where $\theta = \{\pi_i^{(P)}, \mu_i^{(P)}, \Sigma_i^{(P)}\}$ are the parameters of the GMM. The general theory of OT between GMMs was first presented in [19]. The main advantage is that, when the OT plan is restricted to the set of GMMs, i.e., $\gamma \in \Gamma(P, Q) \cap \text{GMM}_{\infty}(d)$, there is a discrete equivalent formulation in terms of the GMM components,

$$\omega^* = \arg \min_{\omega \in \Gamma(\pi^{(P)}, \pi^{(Q)})} \sum_{i=1}^n \sum_{j=1}^m \omega_{ij} \mathbb{W}_2(P_i, Q_j)^2.$$

$\omega^* \in \mathbb{R}^{n \times m}$ is a component-to-component OT plan. The term $\mathbb{W}_2(P_i, Q_j)$ is the Wasserstein distance between Gaussian components, which has closed-form [33] in terms of their parameters,

$$\mathbb{W}_2(P, P')^2 = \|\mu - \mu'\|_2^2 + \text{Tr}(\Sigma + \Sigma' - 2(\Sigma^{\frac{1}{2}} \Sigma' \Sigma^{\frac{1}{2}})^{\frac{1}{2}}).$$

Based on these ideas, one may define a Wasserstein-type distance between GMMs,

$$\text{MW}_2(P, Q)^2 = \inf_{\omega \in \Gamma(\pi^{(P)}, \pi^{(Q)})} \sum_{i=1}^n \sum_{j=1}^m \omega_{ij} \mathbb{W}_2(P_i, Q_j)^2. \quad (6)$$

2.3 Wasserstein Barycenter and its Variants

In the metric setting, the barycenter problem is known as Fréchet [34] or Karcher [35] means. In our case, given a finite family of measures $\mathcal{Q} = \{Q_k\}_{k=1}^K$ and a set of barycentric coordinates $\lambda \in \Delta_K$, we define the

Wasserstein barycenters over $(\mathcal{P}_{p,ac}(\Omega), \mathbb{W}_p)$ through the following optimization problem,

$$P^* = \arg \min_{P \in \mathcal{P}(\Omega)} \left\{ \mathbb{B}_Q(P) = \sum_{k=1}^K \lambda_k \mathbb{W}_p(P, Q_k)^p \right\}. \quad (7)$$

In general, equation 7 does not have a closed-form solution. However, there are algorithms for computing barycenters for $p = 2$, when the measures in \mathcal{P} are either empirical measures [14], Gaussian measures [1, 36] or Gaussian mixtures [19, 12], and a further link can be made to multi-marginal OT [20].

The work of [14] is of particular interest to us, since theirs was the first algorithm, based on gradient descent, to optimize equation 7 on empirical measures. The idea is to initialize $z_{0,1}^{(P)}, \dots, z_{0,n}^{(P)}$ randomly (e.g., from a Gaussian measure), and iterate

$$z_{\tau+1,i}^{(P)} = (1 - \alpha) z_{\tau,i}^{(P)} + \alpha \sum_{k=1}^K \lambda_k T_{\gamma_k}(z_{\tau,i}^{(P)}), \quad (8)$$

where $T_{\gamma_k}(z_i^{(P)}) = n \sum_{j=1}^n \gamma_{k,i,j} z_j^{(Q_k)}$ is the barycentric map between Q_k and P . This strategy is reminiscent of the fixed-point approach of [36]. Furthermore, [11] shows that, for feature-label joint measures, the labels can be propagated through [37],

$$y_{\tau+1,i}^{(P)} = (1 - \alpha) y_{\tau,i}^{(P)} + \alpha \sum_{k=1}^K \lambda_k T_{\gamma_k}(y_{\tau,i}^{(P)}), \quad (9)$$

where $y_{\tau,i}^{(P)}$ are one-hot encoded vectors.

Previous work has extended equation 8 to the GMM setting, when the components are axis-aligned Gaussians, i.e., $\Sigma_i^{(P)} = \text{diag}(\sigma_i^{(P)})$. In this case,

$$\begin{aligned} \mu_{i,\tau+1}^{(P)} &= (1 - \alpha) \mu_{i,\tau}^{(P)} + \alpha \sum_{k=1}^K \lambda_k T_{\omega_k}(\mu_{i,\tau}^{(P)}), \\ \sigma_{i,\tau+1}^{(P)} &= (1 - \alpha) \sigma_{i,\tau}^{(P)} + \alpha \sum_{k=1}^K \lambda_k T_{\omega_k}(\sigma_{i,\tau}^{(P)}), \end{aligned} \quad (10)$$

There are a few limitations with equations 8 and 10. First, the empirical iterations do not scale well to large datasets. Indeed, they assume the availability of all samples of Q_k *per iteration*. Second, the GMM iterations in equation 10 are restricted to axis-aligned GMMs. **This paper addresses these gaps using gradient flows.**

2.4 Gradient Flows

In the Euclidean setting, let $F : \mathbb{R}^d \rightarrow \mathbb{R}$ be a functional. A gradient flow is the solution to the differen-

tial equation,

$$\begin{cases} \dot{x}(t) = -\nabla F(x(t)) & \text{for } t > 0, \\ x(0) = x_0, \end{cases} \quad (11)$$

where x_0 is the initial condition. In $\mathcal{P}_{p,\text{ac}}(\Omega)$, gradient flows correspond to the continuity equation,

$$\partial_t P_t = -\text{div}(P_t \nabla \mathbb{F}(P_t)), \quad (12)$$

where P_t is a curve in $\mathcal{P}_{p,\text{ac}}(\Omega)$, and $\mathbb{F} : \mathcal{P}_{p,\text{ac}}(\Omega) \rightarrow \mathbb{R}$ is a functional over probability measures. Usually, these functionals take the following form [38],

$$\begin{aligned} \mathbb{F}(P) = & \underbrace{\int G(P(z)) dz}_{\mathbb{G}(P)} + \underbrace{\int V(z) dP(z)}_{\mathbb{V}(P)} + \\ & \underbrace{\int \int U(z, z') dP(z) dP(z')}_{\mathbb{U}(P)}, \end{aligned} \quad (13)$$

where \mathbb{G} , \mathbb{V} , and \mathbb{U} are the internal, potential, and interaction energies, respectively. Here, $G : \mathbb{R} \rightarrow \mathbb{R}$ is convex and superlinear, and $V : \Omega \rightarrow \mathbb{R}$ and $U : \Omega^2 \rightarrow \mathbb{R}$ are convex and sufficiently smooth. Henceforth, we denote $\mathbb{F}^* = \inf_{P \in \mathcal{P}_{2,\text{ac}}(\Omega)} \mathbb{F}(P)$.

3 Wasserstein Barycenters as Gradient Flows

In this section, we describe a new method for computing empirical and Gaussian mixture Wasserstein barycenters. Our main idea is using the functional:

$$\mathbb{F}(P) = \mathbb{B}_Q(P) + \mathbb{G}(P) + \mathbb{V}(P) + \mathbb{U}(P), \quad (14)$$

where $\mathbb{B}_Q(P)$ is the barycenter objective defined in equation 7, and \mathbb{G}, \mathbb{V} and \mathbb{U} are the energies defined in equation 13.

3.1 Empirical Flow

In the empirical case, the barycenter is approximated through a finite set of particles (see equation 1) $\{z_i^{(P)}\}_{i=1}^n$. Hence, we may rewrite equation 14 as,

$$\begin{aligned} \min_{\substack{z_1^{(P)}, \dots, z_n^{(P)} \in \Omega, \\ \gamma_k \in \Gamma(\hat{P}, \hat{Q}_k)}} & \sum_{k=1}^K \lambda_k \underbrace{\sum_{i=1}^n \sum_{j=1}^{n_k} \gamma_{k,i,j} d(z_i^{(P)}, z_j^{(Q_k)})^p}_{\mathbb{B}_Q(\hat{P})} + \\ & \underbrace{\frac{1}{n} \sum_{i=1}^n V(z_i^{(P)})}_{\mathbb{V}(\hat{P})} + \underbrace{\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n U(z_i^{(P)}, z_j^{(P)})}_{\mathbb{U}(\hat{P})} \end{aligned} \quad (15)$$

Here, we omit the internal energy term, $\mathbb{G}(P)$, as it is not defined for empirical measures. We solve the previous optimization problem through a block-coordinate descent strategy, that is,

$$\begin{aligned} \gamma_k &= \arg \min_{\gamma \in \Gamma(\hat{P}_\tau, \hat{Q}_k)} \sum_{i=1}^n \sum_{j=1}^{n_k} \gamma_{i,j} d(z_{\tau,i}^{(P)}, z_j^{(Q_k)})^p, \\ z_{\tau+1,i}^{(P)} &= z_{\tau,i}^{(P)} - \alpha \nabla \mathbb{F}(\hat{P}_\tau). \end{aligned} \quad (16)$$

In equation 16, γ_k is used to compute \mathbb{B}_Q and is updated for each τ . We provide in Algorithm 1 the pseudo-code for this strategy. Our algorithm assumes access to Q_k through sampling, that is, we do not necessarily have access to all samples of Q_k at once.

Algorithm 1 Empirical barycenter Wasserstein flow using Gradient Descent.

Input: $\lambda \in \Delta_K$, $\mathcal{Q} = \{Q_k\}_{k=1}^K$, $V : \Omega \rightarrow \mathbb{R}$, $U : \Omega^2 \rightarrow \mathbb{R}$, $n \in \mathbb{N}$, $\alpha \geq 0$, $\hat{P}_0 = n^{-1} \sum_{i=1}^n \delta_{z_{i,0}^{(P)}}$, $n_{\text{iter}} \in \mathbb{N}$.

Output: Barycenter support $\{z_i^{(P)}\}_{i=1}^n$

- 1: **for** $\tau \leftarrow 1$ to n_{iter} **do**
 - 2: $\mathbb{F}(\hat{P}_\tau) \leftarrow \mathbb{V}(\hat{P}_\tau) + \mathbb{U}(\hat{P}_\tau)$
 - 3: **for** $k \leftarrow 1$ to K **do**
 - 4: Sample $\{z_i^{(Q_k)}\}_{i=1}^m$ i.i.d. from Q_k .
 - 5: $\mathbb{F}(\hat{P}_\tau) \leftarrow \mathbb{F}(\hat{P}_\tau) + \lambda_k \mathbb{W}_p(\hat{P}_\tau, \hat{Q}_k)^p$
 - 6: **end for**
 - 7: $z_{\tau+1,i}^{(P)} \leftarrow z_{\tau,i}^{(P)} - \alpha \nabla \mathbb{F}(\hat{P}_\tau)$
 - 8: **end for**
-

Remark 3.1. In line 5 of Algorithm 1, \hat{Q}_k denotes the empirical measure (c.f., equation 1) supported on the mini-batch $\{z_i^{(Q_k)}\}_{i=1}^m$ sampled from Q_k .

3.2 Gaussian Mixture Flow

In the case of GMMs, we flow the parameters $\theta = \{(\pi_k, \mu_k, \Sigma_k)\}_{k=1}^K$ of $P_\theta = \sum \pi_i \mathcal{N}(\cdot | \mu_i, \Sigma_i)$. Similarly to equation 16, we have:

$$\begin{aligned} \omega_k &= \arg \min_{\omega \in \Gamma(\pi, \pi_k)} \sum_{i=1}^n \sum_{j=1}^{n_k} \omega_{i,j} \mathbb{W}_2(P_{\theta_\tau, i}, Q_{k,j})^2, \\ \theta_{\tau+1,i} &= \theta_{\tau,i} - \alpha \nabla \mathbb{F}(P_{\theta_\tau}). \end{aligned} \quad (17)$$

Here, ω_k is a transport plan between GMM components. In general, the energy functionals do not have closed form with respect to the GMM parameters. However, we can estimate them using Monte-Carlo and the reparametrization trick, for instance,

$$z_i = L_i \varepsilon + \mu_i, \quad i \sim \pi, \varepsilon \sim \mathcal{N}(\cdot | 0, \text{Id}), \quad (18)$$

This strategy allows us to propagate the gradients to the parameters of P_τ . We show the overall approach in Algorithm 2.

Algorithm 2 GMM barycenter flow using Gradient Descent.

Input: $\lambda \in \Delta_K$, $\mathcal{Q} = \{Q_k\}_{k=1}^K$, $V : \Omega \rightarrow \mathbb{R}$, $U : \Omega^2 \rightarrow \mathbb{R}$, $n \in \mathbb{N}$, $\alpha \geq 0$, $\hat{P}_0 = n^{-1} \sum_{i=1}^n \delta_{z_{i,0}^{(P)}}$, $n_{\text{iter}} \in \mathbb{N}$.

Output: Barycenter support $\{z_i^{(P)}\}_{i=1}^n$

```

1: for  $\tau \leftarrow 1$  to  $n_{\text{iter}}$  do
2:   Sample  $\{z_i = L_i \varepsilon + \mu_i : i \sim \pi, \varepsilon \sim \mathcal{N}(\cdot | 0, \text{Id})\}$ ,
3:    $\mathbb{F}(\hat{P}_\tau) \leftarrow \mathbb{G}(\hat{P}_\tau) + \mathbb{V}(\hat{P}_\tau) + \mathbb{U}(\hat{P}_\tau)$ 
4:   for  $k \leftarrow 1$  to  $K$  do
5:      $\mathbb{F}(P_{\theta_\tau}) \leftarrow \mathbb{F}(P_{\theta_\tau}) + \lambda_k \text{MW}_2(P_{\theta_\tau}, Q_k)^2$ 
6:   end for
7:    $z_{\tau+1,i}^{(P)} \leftarrow z_{\tau,i}^{(P)} - \alpha \mathbb{W}\mathbb{F}(P_{\theta_\tau})$ 
8: end for

```

Covariance matrix parametrization. Optimizing \mathbb{F} with respect to Σ_k is challenging, due to the symmetric positive definite constraint. We enforce this constraint by writing $\Sigma_k = L_k L_k^T$, where L_k is the lower-triangular Cholesky factor. There are two advantages to this strategy. First, optimizing with respect to L_k is empirically more stable than Σ_k . Second, we can use L_k directly in the reparametrization trick in equation 18.

3.3 Flows over Joint Measures

In [14], the authors considered barycenters for $p = 2$, $\Omega = \mathbb{R}^d$, and $d(z, z') = \|z - z'\|_2$. Therefore, we generalize this setting, since we only need the differentiability of d for our algorithm to work. As we show in our experiments, in machine learning applications one encounters $\Omega = \mathcal{X} \times \mathcal{Y}$, i.e., measures over the joint space of features and labels. Whenever we deal with labeled measures (i.e., $z = (x, y)$), we adopt,

$$d(z, z') = \sqrt{\|x - x'\|_2^2 + \beta \|y - y'\|_2^2}, \quad (19)$$

where $\beta \geq 0$ is a parameter that balances the feature distance terms and the label distance terms.

For regression applications (i.e., $\mathcal{Y} = \mathbb{R}$), this distance is quite natural. However, for classification, \mathcal{Y} is categorical (e.g., $\mathcal{Y} = \{1, \dots, n_{\text{classes}}\}$). One possible strategy, used in [10] and [39], is fixing the labels and flowing only the features. In contrast to these works, we embed \mathcal{Y} into the compact continuous space Δ_K , through a one-hot encoding operation. For our flow, we parametrize labels through a change of variables,

$$y_{i,c}^{(P)} = \text{softmax}(\ell_{i,1}^{(P)}, \dots, \ell_{i,n_c}^{(P)}) = \frac{\exp(\ell_{i,c}^{(P)})}{\sum_{c=1}^{n_c} \exp(\ell_{i,c}^{(P)})},$$

thus, instead of optimizing over $z = (x, y)$, we optimize over $z = (x, \ell)$. From the soft probabilities, we

can retrieve the actual discrete labels with an argmax, $y_i^{(P)} = \text{argmax}_{c=1, \dots, n_c} y_{i,c}^{(P)}$.

For GMMs, we equip their components with labels $\nu_k \in \Delta_{n_{\text{classes}}}$. Using the ground-cost in equation 19, we are able to write the $\text{MW}_2(P, Q)^2$ as,

$$\sum_{i=1}^n \sum_{j=1}^m \omega_{ij}^* (\mathbb{W}_2(P_{i,x}, Q_{j,x})^2 + \beta \|\nu_i^{(P)} - \nu_j^{(Q)}\|_2^2),$$

where $P_{i,x} = \mathcal{N}(\cdot | \mu_i^{(P)}, \Sigma_i^{(P)})$ is the Gaussian feature marginal. This approach was first proposed in [12], but the authors relied on heuristic arguments to justify this modeling choice. We present the following proposition justifying the label term,

Proposition 3.1. Let $P = \sum \pi_i^{(P)} (\mathcal{N}(\mu_i^{(P)}, \Sigma_i^{(P)}) \otimes \delta_{\nu_i^{(P)}})$ and $Q = \sum \pi_j^{(Q)} (\mathcal{N}(\mu_j^{(Q)}, \Sigma_j^{(Q)}) \otimes \delta_{\nu_j^{(Q)}})$ be two GMMs over $\Omega = \mathcal{X} \times \mathcal{Y}$. Let the ground cost c be,

$$c(z, z') = \|x - x'\|_2^2 + \rho(y, y')^2,$$

where $\rho(y, y')$ is a metric over \mathcal{Y} . Then,

$$\text{MW}_2(P, Q)^2 = \min_{\omega \in \Gamma(\pi^{(P)}, \pi^{(Q)})} \sum_{i=1}^n \sum_{j=1}^m \omega_{ij} C_{ij},$$

where $C_{ij} = \mathbb{W}_2(P_{i,x}, Q_{j,x})^2 + \rho(\nu_i^{(P)}, \nu_j^{(Q)})^2$.

Functionals for joint measures. One of the advantages of our proposed method is regularizing the barycenter calculation with internal, interaction, and potential energy functionals. This idea was already used in practice by [39] for transfer learning problems. Here we propose the following functionals,

$$V_E(z) = - \sum_{c=1}^{n_{\text{classes}}} y_c \log y_c, \quad (20)$$

$$U_R(z, z') = \begin{cases} h(d(x, x')) & \text{if } y \neq y', \\ 0 & \text{otherwise,} \end{cases} \quad (21)$$

where $h : \mathbb{R} \rightarrow \mathbb{R}$ is lower semi-continuous and bounded from below. For instance, in our experiments we use the hinge loss, $h(u) = \max(0, \text{margin} - d)$, where $\text{margin} \geq 0$ is a margin parameter. Equations 20 and 21 correspond to entropy and repulsion terms. The first functional penalizes barycenters that have fuzzy labels. The second functional encourages classes to be well separated.

3.4 Convergence

One of the difficulties in analyzing the gradient flow of the functional in equation 14 comes from the fact that $P \mapsto \mathbb{W}_p(P, Q)^p$ is not geodesically convex in

$\mathcal{P}_{p,ac}(\Omega)$ [28, Section 4.4]. As a result, we are minimizing a non-convex functional. Henceforth, we focus on $p = 2$, as this is the setting where most results are available. **Explicit constants and proofs are available in Section 4 in the Appendix.**

Our analysis relies on a measure-theoretic version of the Polyak–Łojasiewicz (PL) inequality [40, 41],

$$\|\nabla \mathbb{B}(P)\|_{L_2(P)} \geq C_{\text{PL}}(\mathbb{B}(P) - \mathbb{B}^*), \quad (22)$$

which is covered in [42]. Under this assumption, we have the following convergence result,

Theorem 3.1. *Let $\Omega = \mathcal{B}(0, R)$ be the closed ball on \mathbb{R}^d with radius $R > 0$. Let $P^* \in \mathcal{P}_{2,ac}(\mathcal{B}(0, R))$ be the barycenter of $\mathcal{Q} = \{Q_k\}_{k=1}^K$, $Q_k \in \mathcal{P}_{2,ac}(\mathcal{B}(0, R))$, $k = 1, \dots, K$, with barycentric coordinates $\lambda = (\lambda_1, \dots, \lambda_K) \in \Delta_K$. Assume the inequality 22. Then, the following holds,*

$$\mathbb{E}[\hat{\mathbb{B}}(P_\tau) - \hat{\mathbb{B}}^*] \leq e^{-C_{PL}\tau}(\mathbb{B}(P_0) - \mathbb{B}^*) + C_R \sqrt{\frac{C_{d,m}}{n}}, \quad (23)$$

where the expectation on the l.h.s. is taken with respect to samples from Q_k . The constants C_R and $C_{d,m}$ depend on the radius R , number of barycenter samples m , and dimensions d .

The right hand side of inequality 23 is composed of two terms. The first term comes from the PL inequality, and covers the convergence towards the minimizer of \mathbb{B} . The second term covers the error of empirical approximations. In this sense, letting $\tau \rightarrow +\infty$ leads to an error governed by the empirical approximation.

Next, we derive a new result for the convergence of the GMM gradient flow (cf. Algorithm 2). Here, we assume P and $Q_k \in \mathcal{Q}$ are GMMs. Furthermore, we denote by \hat{Q}_k the GMM fitted on data from Q_k with the expectation-maximization algorithm. Our result relies on [43, Theorem 3.1]. Therefore, we assume,

1. **bounded GMM parameters**, i.e., $\|\mu_i^{(Q_k)}\|_2 \leq R_\mu$, $\sqrt{\text{Tr}(\Sigma_i^{(Q_k)})} \leq R_\Sigma \forall i, \forall k$ (resp. $\hat{\mu}_i^{(Q_k)}$ and $\hat{\Sigma}_i^{(Q_k)}$, the estimated parameters); furthermore,
2. **Empirical convergence rates**, i.e., $\mathbb{E}[\|\pi^{(Q_k)} - \hat{\pi}^{(Q_k)}\|_1] \leq \rho_\pi$, $\mathbb{E}[\|\mu_i^{(Q_k)} - \hat{\mu}_i^{(Q_k)}\|_2] \leq \rho_\mu$, and $\mathbb{E}[d_{\text{Bures}}(\Sigma_i^{(Q_k)}, \hat{\Sigma}_i^{(Q_k)})] \leq \rho_\Sigma, \forall i \text{ and } \forall k$.

Here the expectations are taken with respect to samples drawn from Q_k . Under these conditions we have,

Theorem 3.2. *Let P and Q be two labeled GMMs with bounded parameters and satisfying the empirical convergence rates. Then, the following holds,*

$$\mathbb{E}[\hat{\mathbb{B}}(P_\tau) - \hat{\mathbb{B}}^*] \leq e^{-C_{PL}\tau}(\mathbb{B}(P_0) - \mathbb{B}^*) + C_{\lambda,Q},$$

where $C_{\lambda,Q}$ is a constant that depends on the coordinates λ and GMMs in \mathcal{Q} .

Here $C_{\lambda,Q}$ depends on the $(\pi^{(Q_k)}, \mu^{(Q_k)}, \Sigma^{(Q_k)})$ of each GMM, but not on $\nu^{(Q_k)}$. Indeed, since we fit a GMM per class, the labels are estimated exactly (i.e., $P_{i,x}$ and $\hat{P}_{i,x}$ belong to the same class).

4 Experiments

We divide our experiments into a toy example (Section 4.1) to illustrate our methods, and multi-source domain adaptation experiments (Section 4.2). Due to length constraints, we include additional experiments in Sections 5 and 6 in the Appendix. In particular, in Table 1 in the Appendix, we show a running-time comparison demonstrating the scalability of our method.

4.1 Toy Example

In Figure 2, we show the Swiss roll measure Q_0 , alongside four variations obtained via $T_{k,\#}Q_0$, $k = 1, \dots, 4$. This example has been used in several state-of-the-art Wasserstein barycenter works [21, 25]. Originally there are labels associated with the samples of the Swiss-roll measure, corresponding to the position of the points in the underlying manifold. We show a summary of our results in Figure 3.

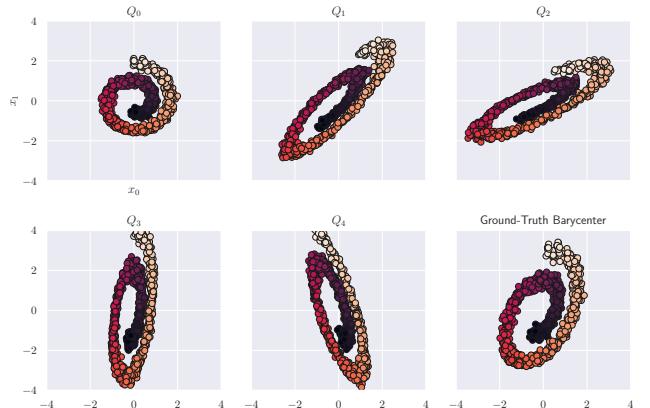


Figure 2: Location-scatter family generated by a Swiss-roll measure, Q_0 . Each measure $Q_k = T_{k,\#}Q_0$, for $T_k(x) = A_kx + b_k$.

First, we compare unsupervised barycenter solvers in Figures 3 (a) through (h). Quantitatively, empirical methods, notably [14, Algorithm 2] and our WGF algorithm (cf. Algorithm 1) achieve the lowest Wasserstein distance to the ground truth in Figure 2. We note that, overall, while the neural network solvers are usually more scalable than discrete methods in terms of number of samples, their optimization problem is more complicated and very sensitive to hyper-parameters.

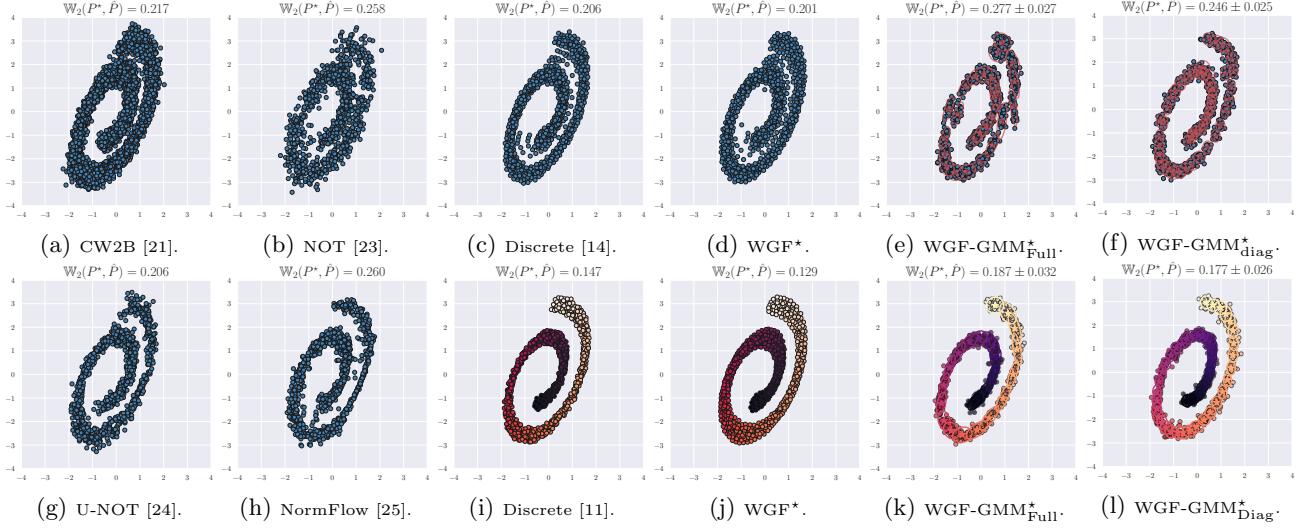


Figure 3: Comparison between Wasserstein barycenter solvers. Colored scatter plots indicate labeled barycenters. For each solver, we compute the Wasserstein distance between its solution \hat{P} and the ground-truth P^* , shown in the title of each sub figure (best seen on screen). Overall, using label information leads to barycenters that better approximate the ground-truth barycenter.

Second, we experiment with integrating labels in the ground cost as described in Section 3.3. These methods are shown in Figures 3 (i) through (l). In all cases, integrating labels produces barycenters that are closer to the ground-truth. We conclude that using the labels gives a strong inductive bias in the barycenter computation, which helps explain the gain in performance of labeled barycenters in the next section.

4.2 Multi-Source Domain Adaptation

Problem Formulation. One of the main applications of Wasserstein barycenters is multi-source domain adaptation (MSDA) [44, 45, 46]. In this setting one needs to adapt multiple labeled source measures Q_1, \dots, Q_K to a single unlabeled target measure Q_T . The goal is to learn, from samples $\{\{x_i^{(Q_k)}, y_i^{(Q_k)}\}_{i=1}^{n_k}\}_{k=1}^K$ and $\{x_i^{(Q_T)}\}_{i=1}^{n_T}$, a classifier h that achieves low risk in the target domain measure,

$$\mathcal{R}_{Q_T}(h) = \mathbb{E}_{(x,y) \sim Q_T} [\mathcal{L}(y, h(x))],$$

for a loss function \mathcal{L} (e.g., cross-entropy loss). We isolate the quality of barycenters by doing adaptation at the level of embeddings. This approach allows us to perform domain adaptation in a higher semantic space, where distributions are more meaningful and comparable across domains. Thus, we assume that a meaningful feature extractor ϕ , called the backbone, has been previously learned. We obtain the feature extractor by fine-tuning a neural network on the labeled source domain data. In addition, similar to [10, 12], we align the barycenter with the target through OT [47]. More details are available in Section 6 in the Appendix.

Benchmark	Backbone	# Samples	# Domains	# Dim.	# Classes
Office31	ResNet50 [48]	3287	3	2048	31
BCI-CIV-2a	CBraMod [49]	5184	10	200	4
TEP	CNN [50]	17289	6	128	29
Office Home	ResNet101 [48]	15500	4	2048	65
ISRUIC	CBraMod [49]	89240	100	512	5

Table 1: Overview of benchmarks used for domain adaptation, sorted by number of samples.

Experimental Setting. We run our experiments on five benchmarks: Office 31 [51], Office Home [52], BCI-CIV-2a [53], ISRUC [54], and TEP [50]. The first two, second two, and last benchmarks correspond to computer vision, neuroscience, and chemical engineering benchmarks, respectively. We show in Table 1 an overview of our experimental setting.

Compared Methods. Overall, we compare seven Wasserstein barycenter strategies with ours. These methods are: discrete barycenters [14, Algorithm 2], normalizing flows [25], continuous 2-Wasserstein barycenters [21], neural OT barycenters [23], and unbalanced neural OT barycenters [24]. We also include *labeled barycenter strategies* [11, 55]. A brief review of these methods is available in Section 2 in the Appendix. For completeness, we include four other state-of-the-art methods in domain adaptation over embedding vectors. Those methods are: WJDOT [56], DaDiL-R and E [11], and GMM-DaDiL [12]. For each benchmark, we use one domain as the target domain (e.g., Amazon vs. {dSLR, Webcam} in the Office 31 benchmark), and we measure the classification accuracy, i.e., the percentage of correct predictions.

Benchmark	$\mathcal{X} \times \mathcal{Y}$	Office31	OfficeHome	BCI-CIV-2a	ISRUC	TEP
Backbone	-	ResNet-50	ResNet-101	CBraMod	CBraMod	CNN
Source-Only	-	86.40	75.95	50.30	76.63	78.48
WJDOT [56]	-	86.80	76.59	N/A	76.95	86.13
DaDiL-R [11]	-	89.91	77.86	53.41	74.68	86.14
DaDiL-E [11]	-	89.79	78.14	N/A	75.89	85.87
GMM-DaDiL [12]	-	90.63	78.81	57.10	75.47	86.85
Discrete [14]	✗	81.94	70.42	57.38	70.67	83.81
NormFlow [25]	✗	85.91	76.73	55.45	78.04	82.89
CW2B [21]	✗	86.37	76.44	57.25	75.84	85.83
NOT [23]	✗	86.22	75.36	57.29	76.39	84.93
U-NOT [24]	✗	86.97	76.85	57.17	77.06	85.43
Discrete [11]	✓	87.93	77.09	57.43	78.20	86.09
GMM [12]	✓	88.54	77.87	57.04	74.58	84.67
WGF (ours)	✓	88.14	<u>77.83</u>	57.50	79.78	<u>86.21</u>
WGF-GMM (ours)	✓	89.31	78.71	57.35	<u>78.72</u>	86.77

Table 2: Average classification accuracy on target domains. In total, we compare eight methods against our WGF strategy. For barycenter solvers, we indicate by ✓methods that compute barycenters on $\Omega = \mathcal{X} \times \mathcal{Y}$. Bold numbers represent the best **barycenter** method, and underlined numbers represent second best.

Main results. We present our main results in Table 2, which reports the average performance per domain on each benchmark. We provide fine-grained results in Section 6 in the Appendix. Overall, labeled barycenter methods (the four last rows in Table 2) have a clear advantage with respect to the unsupervised methods. We argue that label information is pivotal in domain adaptation success, which is consistent with previous research [47, 10, 11, 12]. Among the labeled barycenter methods, our gradient flow framework achieves either the best performance, or remains competitive, surpassing previous methods in MSDA, such as WJDOT and DaDiL in the ISRUC benchmark.

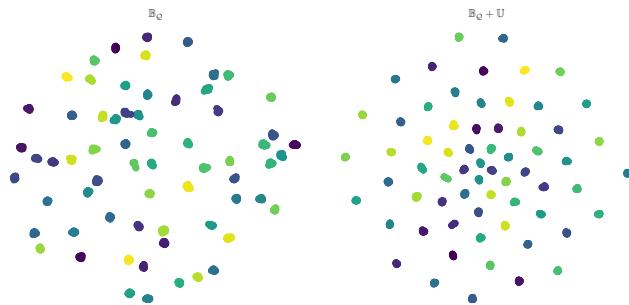


Figure 4: t-SNE [57] visualization of the barycenter of [Art, Product, Real-World] source domains in the Office home benchmark. Colors represent different classes from 1 to 65. Overall, the classes tend to be more separated when using the repulsion functional \mathbb{U} .

Visualization. We now visualize the effect of adding the repulsion interaction energy functional on the Office-Home benchmark. Adding \mathbb{U} leads to classes that are well separated. For computer vision experiments we use ResNets [48] as the backbone, which produce 2048-dimensional embeddings. For that reason, we use the cosine distance $d(x, x') = 1 - \text{cossim}(x, x')$

in equation 21 and we fix margin = 1. This choice encourages class clusters to live in orthogonal sub-spaces of the latent space. We show a comparison of the obtained barycenters in Figure 4. Overall, the classes tend to have less overlap when the interaction energy is added to the barycenter functional.

Ablation. We ablate the effect of \mathbb{V} and \mathbb{U} in the Office Home benchmark. For this benchmark, we use $\mathbb{V}(P) = \mathbb{W}_2(\hat{P}, \hat{Q}_T)^2$. The intent here is to isolate the contribution of each component to the state-of-the-art results in Table 2. Our results are summarized in Table 3, where we further compare our results with other related barycenter methods. In the empirical setting, the use of Algorithm 1 alone already presents a gain in performance compared to its discrete counterpart. Overall, the combination of $\mathbb{B} + \mathbb{V} + \mathbb{U}$ consistently gives the best performance, showing the importance of adding underlying structure to the Wasserstein barycenter in domain adaptation.

Method	$\mathcal{X} \times \mathcal{Y}$	\mathbb{B}	$\mathbb{B} + \mathbb{V}$	$\mathbb{B} + \mathbb{U}$	$\mathbb{B} + \mathbb{V} + \mathbb{U}$
Discrete [14]	✗	70.42	N/A	N/A	N/A
Discrete [11]	✓	77.09	N/A	N/A	N/A
WGF (ours)	✓	77.48	77.07	77.08	77.83
GMM [12]	✓	77.87	N/A	N/A	N/A
WGF-GMM (ours)	✓	77.28	78.28	78.12	78.71

Table 3: Ablation of average target domain classification accuracy on the Office Home benchmark.

5 Conclusion

In this paper we have introduced a new framework for computing Wasserstein barycenters, based on Wasserstein gradient flows [27, 28]. Our approach addresses scalability issues in discrete barycenter solvers [14], and is capable of regularizing the underlying barycenter through internal, potential, and interaction functionals. We presented gradient flow Algorithms 1 and 2, for empirical and Gaussian mixture measures, respectively. We further proved convergence guarantees based on the PL inequality.

We empirically tested our method against existing discrete [14, 11] and neural-network [21, 23, 24, 25]-based barycenter solvers on computer vision, neuroscience, and chemical engineering benchmarks, demonstrating that our proposed methods consistently achieve state-of-the-art performance. Our empirical findings (Table 2) further demonstrate that incorporating label information in the OT objective is key for domain adaptation. This finding highlights an interesting gap in the literature, where neural net-based solvers are not capable of exploiting this information, and thus have sub-optimal performance in domain adaptation. We leave an investigation of this issue for future work.

References

- [1] Martial Aguech and Guillaume Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.
- [2] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2008.
- [3] Benoît Kloeckner. A geometric study of wasserstein spaces: Euclidean spaces. *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze*, 9(2):297–323, 2010.
- [4] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020.
- [5] Camille Le Coz, Alexis Tantet, Rémi Flamary, and Riwal Plougonven. A barycenter-based approach for the multi-model ensembling of subseasonal forecasts. *arXiv preprint arXiv:2310.17933*, 2023.
- [6] Jiacheng Zhu, Jielin Qiu, Aritra Guha, Zhuolin Yang, XuanLong Nguyen, Bo Li, and Ding Zhao. Interpolation for robust learning: Data augmentation on wasserstein geodesics. In *International conference on machine learning*, pages 43129–43157. PMLR, 2023.
- [7] Haoyang Liu, Yijiang Li, Tiancheng Xing, Vibhu Dalal, Luwei Li, Jingrui He, and Haohan Wang. Dataset distillation via the wasserstein metric. *arXiv preprint arXiv:2311.18531*, 2023.
- [8] Eduardo Fernandes Montesuma, Fred Ngolé Mboula, and Antoine Souloumiac. Multi-source domain adaptation meets dataset distillation through dataset dictionary learning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5620–5624. IEEE, 2024.
- [9] 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.
- [10] 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*, pages 16785–16793, 2021.
- [11] Eduardo Montesuma, Fred Maurice Ngole Mboula, and Antoine Souloumiac. Multi-source domain adaptation through dataset dictionary learning in wasserstein space. In *ECAI 2023*, pages 1739–1746. IOS Press, 2023.
- [12] Eduardo Fernandes Montesuma, Fred Ngolé Mboula, and Antoine Souloumiac. Lighter, better, faster multi-source domain adaptation with gaussian mixture models and optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 21–38. Springer, 2024.
- [13] Sanvesh Srivastava, Cheng Li, and David B Dunson. Scalable bayes via barycenter in wasserstein space. *Journal of Machine Learning Research*, 19(8):1–35, 2018.
- [14] Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In *International conference on machine learning*, pages 685–693. PMLR, 2014.
- [15] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.
- [16] Hicham Janati, Marco Cuturi, and Alexandre Gramfort. Debiased sinkhorn barycenters. In *International Conference on Machine Learning*, pages 4692–4701. PMLR, 2020.
- [17] Sinho Chewi, Tyler Maunu, Philippe Rigollet, and Austin J Stromme. Gradient descent algorithms for bures-wasserstein barycenters. In *Conference on Learning Theory*, pages 1276–1304. PMLR, 2020.
- [18] Alexey Kroshnin, Vladimir Spokoiny, and Alexandra Suvorikova. Statistical inference for bures–wasserstein barycenters. *The Annals of Applied Probability*, 31(3):1264–1298, 2021.
- [19] Julie Delon and Agnès Desolneux. A wasserstein-type distance in the space of gaussian mixture models. *SIAM Journal on Imaging Sciences*, 13(2):936–970, 2020.
- [20] Brendan Pass. Multi-marginal optimal transport: theory and applications. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1771–1790, 2015.
- [21] Alexander Korotin, Lingxiao Li, Justin Solomon, and Evgeny Burnaev. Continuous wasserstein-2 barycenter estimation without minimax optimization. *arXiv preprint arXiv:2102.01752*, 2021.

- [22] Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International conference on machine learning*, pages 146–155. PMLR, 2017.
- [23] Alexander Kolesov, Petr Mokrov, Igor Udvichenko, Milena Gazdieva, Gudmund Pammer, Evgeny Burnaev, and Alexander Korotin. Estimating barycenters of distributions with neural optimal transport. *arXiv preprint arXiv:2402.03828*, 2024.
- [24] Milena Gazdieva, Jaemoo Choi, Alexander Kolesov, Jaewoong Choi, Petr Mokrov, and Alexander Korotin. Robust barycenter estimation using semi-unbalanced neural optimal transport. *arXiv preprint arXiv:2410.03974*, 2024.
- [25] Gabriele Visentin and Patrick Cheridito. Computing optimal transport maps and wasserstein barycenters using conditional normalizing flows. *arXiv preprint arXiv:2505.22364*, 2025.
- [26] Alexander Korotin, Vage Egiazarian, Lingxiao Li, and Evgeny Burnaev. Wasserstein iterative networks for barycenter estimation. *Advances in Neural Information Processing Systems*, 35:15672–15686, 2022.
- [27] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.
- [28] Filippo Santambrogio. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 7:87–154, 2017.
- [29] 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.
- [30] Eduardo Fernandes Montesuma, Fred Maurice Ngolè Mboula, and Antoine Souloumiac. Recent advances in optimal transport for machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(2):1161–1180, 2025.
- [31] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, 1781.
- [32] L Kantorovich. On the transfer of masses (in russian). In *Doklady Akademii Nauk*, volume 37, pages 227–229, 1942.
- [33] Asuka Takatsu. Wasserstein geometry of gaussian measures. *Osaka Journal of Mathematics*, 2011.
- [34] Maurice Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. In *Annales de l'institut Henri Poincaré*, volume 10, pages 215–310, 1948.
- [35] Hermann Karcher. Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics*, 30(5):509–541, 1977.
- [36] 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.
- [37] 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*, pages 849–858. PMLR, 2019.
- [38] Filippo Santambrogio. *Optimal transport for applied mathematicians*, volume 87. Springer, 2015.
- [39] David Alvarez-Melis and Nicolò Fusi. Dataset dynamics via gradient flows in probability space. In *International conference on machine learning*, pages 219–230. PMLR, 2021.
- [40] Boris T Polyak. Gradient methods for solving equations and inequalities. *USSR Computational Mathematics and Mathematical Physics*, 4(6):17–32, 1964.
- [41] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-lojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 795–811. Springer, 2016.
- [42] Sinho Chewi, Jonathan Niles-Weed, and Philippe Rigollet. Statistical optimal transport. *arXiv preprint arXiv:2407.18163*, 3, 2024.
- [43] Samuel Boité, Eloi Tanguy, Julie Delon, Agnès Desolneux, and Rémi Flamary. Differentiable expectation-maximisation and applications to gaussian mixture model optimal transport. *arXiv preprint arXiv:2509.02109*, 2025.
- [44] Wouter M Kouw and Marco Loog. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):766–785, 2019.

- [45] Shiliang Sun, Honglei Shi, and Yuanbin Wu. A survey of multi-source domain adaptation. *Information Fusion*, 24:84–92, 2015.
- [46] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.
- [47] 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.
- [48] 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*, pages 770–778, 2016.
- [49] Jiquan Wang, Sha Zhao, Zhiling Luo, Yangxuan Zhou, Haiteng Jiang, Shijian Li, Tao Li, and Gang Pan. Cbramod: A criss-cross brain foundation model for eeg decoding. *arXiv preprint arXiv:2412.07236*, 2024.
- [50] Eduardo Fernandes Montesuma, Michela Mulas, Fred Ngolé Mboula, Francesco Corona, and Antoine Souloumiac. Benchmarking domain adaptation for chemical processes on the tennessee eastman process. In *ML4CCE Workshop*, 2024.
- [51] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.
- [52] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.
- [53] Clemens Brunner, Robert Leeb, Gernot Müller-Putz, Alois Schlögl, and Gert Pfurtscheller. Bci competition 2008–graz data set a. *Institute for knowledge discovery (laboratory of brain-computer interfaces), Graz University of Technology*, 16(1-6):34, 2008.
- [54] Sirvan Khalighi, Teresa Sousa, Gabriel Pires, and Urbano Nunes. Automatic sleep staging: A computer assisted approach for optimal combination of features and polysomnographic channels. *Expert Systems with Applications*, 40(17):7046–7059, 2013.
- [55] Eduardo Fernandes Montesuma. *Multi-Source Domain Adaptation through Wasserstein Barycenters*. PhD thesis, Université Paris-Saclay, 2024.
- [56] Rosanna Turrisi, Rémi Flamary, Alain Rakotomamonjy, and Massimiliano Pontil. Multi-source domain adaptation via weighted joint distributions optimal transport. In *Uncertainty in artificial intelligence*, pages 1970–1980. PMLR, 2022.
- [57] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.