TOWARDS UNIVERSAL WASSERSTEIN BARYCENTERS THROUGH FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Defining a weighted mean over probability measures under probability metrics is a central tool in probabilistic machine learning. Under the Wasserstein metric, these are called *Wasserstein barycenters*. While most approaches compute barycenters for a fixed weight vector, approximating the whole family of barycenters over the simplex, which we call the *Wasserstein simplex*, remains underexplored. We refer to this problem as *Universal Barycenter Approximation*, and propose `BaryFM`, a flow matching model transporting the marginal measures into any barycenter in the Wasserstein simplex. Once trained, the network can draw samples from measures in the Wasserstein simplex through an ordinary differential equation. We validate our method on 4 downstream tasks: domain adaptation, generalization, Bayesian posterior aggregation and algorithmic fairness. `BaryFM` achieves the best average rank among 15 competing methods across 10 domain adaptation benchmarks, matching or surpassing non-universal solvers.

1 INTRODUCTION

In probabilistic machine learning, one is often faced with the question of how to compute an average or centroid over a family of probability measures (Srivastava et al., 2018; Montesuma & Mboula, 2021) under a suitable metric. Previous studies (Cuturi & Doucet, 2014; Korotin et al., 2022a; Howard et al., 2025; Montesuma et al., 2026a) consider this question through the lens of the Wasserstein distance, in which case the centroid is known as Wasserstein barycenter (Agueh & Carlier, 2011), but despite specific cases (Fan et al., 2020, Appendix B), the question of how to approximate the family of *interpolating measures* remains open. Figure 1 shows an example of this family over $K = 3$ Gaussian marginals p_1, p_2, p_3 ($p_{1:3}$, in short). Therefore, we study the *Wasserstein simplex*,

$$\mathcal{W}(p_{1:K}) = \{\text{Bar}(\lambda, p_{1:K}) : \lambda \in \Delta_K\}, \tag{1}$$

where $\Delta_K = \{\lambda \in \mathbb{R}^K_{\geq 0} : \sum_k \lambda_k = 1\}$ is the simplex, and $q^\star_\lambda = \text{Bar}(\lambda, p_{1:K})$ denotes the Wasserstein barycenter (c.f. equation 10) of marginals $p_{1:K}$ with weights $\lambda \in \Delta_K$.

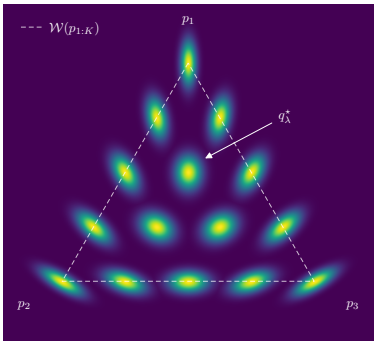


Figure 1: Wasserstein simplex over the set of Gaussian measures. In this paper, we study how to approximate every barycenter in the family $\mathcal{W}(p_{1:K})$.

In this context, barycenter methods can be classified in 2 axes of *amortization*. Most methods are designed to amortize sampling, meaning, drawing new examples from a barycentric measure $q^\star_\lambda$, usually with uniform weights $\lambda = \bar{\lambda} = (K^{-1}, \cdots, K^{-1}) \in \Delta_K$. The second axis of amortization concerns the measures $q^\star_\lambda \in \mathcal{W}(p_{1:K})$ themselves, namely, the possibility of drawing samples from $q^\star_\lambda$ on any query $\lambda \in \Delta_K$ without solving a new optimization problem. The second problem is underexplored, and, to the best of our knowledge, only (Fan et al., 2020, Appendix B) has considered it in restricted scenarios (e.g., Gaussian measures). We call this problem **Universal Barycenter Approximation**, and propose a universal barycenter approximation algorithm, i.e., a method that amortizes sampling along the 2 prescribed axes. In the following, we discuss prior art on each axis.

On the first axis, existing approaches are mainly based on generative models and neural networks. For instance, Fan et al. (2020) and Korotin et al. (2021) propose approaches based on Input Con-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

vex Neural Networks (ICNNs) (Amos et al., 2017) to approximate transport maps. On the generative modeling side, Korotin et al. (2022b) parametrize the barycentric measure as a map from a latent space to the data space, Visentin & Cheridito (2025) computes barycenters via normalizing flows (Papamakarios et al., 2021), and Noble et al. (2023); Howard et al. (2025) use Schrodinger bridges (De Bortoli et al., 2021). Non-amortized methods (Cuturi & Doucet, 2014; Montesuma et al., 2026a), also called empirical, directly optimize the support of the barycentric measure, which limits its scalability since each new drawn requires a new problem solve.

On the second axis, the aforementioned families share a common limitation, that is, they need to solve equation 10 from scratch for each $\lambda \in \Delta_K$. A notable exception is (Fan et al., 2020, Appendix B), who adds λ as an input to their $(2K+1)$-ICNNs, and validates their approach over Gaussian measures. Beyond these amortization axes, downstream tasks such as domain adaptation and generalization require barycenters to computed over joint measures (Montesuma & Mboula, 2021; Montesuma et al., 2026a), i.e., barycenters over $\Omega = \mathbb{R}^d \times \Delta_C$, for C-classes. To the best of our knowledge, no existing neural or universal method handles joint measures (c.f. Table 2).

We propose `BaryFM`, a Conditional Flow Matching (CFM) (Lipman et al., 2022) model that transports the marginal measures $p_{1:K}$ into any measure in their Wasserstein simplex $\mathcal{W}(p_{1:K})$. Our architecture (c.f. Figure 2) is composed of a shared encoding layer that embeds the variables associated with the flow, namely $z = (x, y)$, $t \in [0, 1]$, $k \in \{1, \cdots, K\}$ and $\lambda \in \Delta_K$. This architecture has 2 heads. One CFM head that predicts the velocity field of the associated flow, and one dual Optimal Transport (OT) head, that serves to match samples from the involved measures, as well as to transport them. The distinguishing feature of our CFM method is that it estimates its targets *on-the-fly*. Concretely, our contributions are as follows,

1. We propose `BaryFM` and its architecture, to the best of our knowledge, the first neural, universal barycenter solver that handles joint measures,
2. We provide a comprehensive evaluation of our method against prior art on 4 downstream applications of Wasserstein barycenters: domain adaptation (Section 4.1), generalization (Section 4.2), fairness (Section F) and Bayesian Posterior Aggregation (Section E).

2 Background

2.1 Optimal Transport and Barycenters

Let $\Omega \subseteq \mathbb{R}^{\text{dim}}$ be a convex sub-set of the Euclidean space. OT is a field of mathematics studying the transport of probability measures at least effort (Villani, 2009; Peyré & Cuturi, 2019). It was originally founded by Gaspar Monge Monge (1781), where one seeks for the OT map from $p \in \mathcal{P}_2(\Omega)$ to $q \in \mathcal{P}_2(\Omega)$, where $\mathcal{P}_2(\Omega)$ is the space of probability measures with finite 2nd moments:

$$T^{\star}_{p \to q} = \underset{T \in \mathcal{T}(p,q)}{\arg\inf} \; \mathbb{E}_{z \sim p}[c(z, T(z))] \quad \mathcal{T}(p,q) = \{T : \Omega \to \Omega : T_{\sharp} p = q\}, \tag{2}$$

where $T_{\sharp} p$ denotes the push-forward of p by the map T, and $c : \Omega \times \Omega \to \mathbb{R}$ is a ground-cost. In the 20th century, Kantorovich (1942) introduced a relaxation of the Monge problem,

$$\gamma^{\star} = \underset{\gamma \in \Gamma(p,q)}{\arg\inf} \; \mathbb{E}_{(z,z') \sim \gamma}[c(z, z')] \quad \Gamma(p,q) = \{\gamma \in \mathcal{P}_2(\Omega \times \Omega) : \pi_{1,\sharp}\gamma = p, \pi_{2,\sharp}\gamma = q\}, \tag{3}$$

where $\mathcal{P}_2(\Omega)$ denotes the set of probability measures with finite second moment and $\pi_i(z_1, \cdots, z_n) = z_i$ denotes the $i-$th coordinate projection. Following Cuturi (2013), one can regularize the Kantorovich problem with the entropy of γ, leading to,

$$\gamma^{\star}_{\epsilon} = \text{OT}_{\epsilon}(p,q) = \underset{\gamma \in \Gamma(p,q)}{\arg\inf} \; \mathbb{E}_{(z,z') \sim \gamma}[c(z, z')] - \epsilon H(\gamma) \quad H(\gamma) = -\int \log \frac{d\gamma}{d(p \otimes q)} d\gamma, \tag{4}$$

where $p \otimes q$ denotes the product measure, and the regularization strength $\epsilon \geq 0$. Here, we use the convention $H(\gamma) = -\infty$ when $\gamma \not\ll p \otimes q$. Through equations 3 and 4 one can define notions of dissimilarity on $\mathcal{P}_2$, known as Wasserstein distance ($\epsilon = 0$) or its entropic approximation ($\epsilon > 0$). Let $c(z, z') = \|z - z'\|_2^2$ be the Euclidean metric on Ω,

$$W_{2,\epsilon}(p,q)^2 = \inf_{\gamma \in \Gamma(p,q)} \mathbb{E}_{(z,z') \sim \gamma}[d(z, z')^2] - \epsilon H(\gamma). \tag{5}$$

In general OT lifts $(\Omega, \|\cdot\|_2)$ into $(\mathcal{P}_2(\Omega), W_{2,\epsilon})$, and more precisely, when $\epsilon = 0$ $(\mathcal{P}_2(\Omega), W_{2,0})$ defines a metric space. The Kantorovich problem (equation 3) defines a linear program. As such, via Fenchel-Rockafellar duality (Rockafellar, 1997; Villani, 2021; Genevay et al., 2016), the primal forms in equations 3 and 4 admit a dual,

$$(f_\epsilon^\star, g_\epsilon^\star) = \text{DualOT}_\epsilon(p, q) = \sup_{f,g \in \mathcal{C}(\Omega)} \mathbb{E}_{z\sim p}[f(z)] + \mathbb{E}_{z'\sim q}[g(z')] - I_\epsilon(f, g, c), \tag{6}$$

where $I_\epsilon(f, g, c) = \epsilon \mathbb{E}_{z\sim p, z'\sim q}[\exp(\frac{f(z)+g(z')-c(z,z')}{\epsilon})]$ for $\epsilon > 0$ and $I_0(f, g, c) = \iota_{\Phi(c)}(f, g)$, where $\iota_A(x)$ is the indicator function of set A (i.e., 0 if $x \in A$ and $+\infty$ otherwise), $\Phi(c) = \{(f, g) : f(z) + g(z') \leq c(z, z')\}$, and $\mathcal{C}(\Omega)$ is the set of continuous functions over Ω. Following Genevay et al. (2016), the dual problem can be cast as a single function maximization using the c-transform,

$$f_\epsilon^{(c)}(z') = \text{c-transform}_\epsilon(f_\epsilon^\star)(z') = \begin{cases} \min_{z\in\Omega}\{c(z, z') - f_\epsilon^\star(z)\} & \epsilon = 0, \\ -\epsilon \log \mathbb{E}_{z\sim p}[\exp(\frac{f_\epsilon^\star(z)-c(z,z')}{\epsilon})] & \epsilon > 0. \end{cases} \tag{7}$$

We refer readers to (Peyré & Cuturi, 2019, Remark 4.25) for further computational insight. Equations 6 and 7 will be useful in Section 3, as we approximate $f_\epsilon^\star$ with a neural network, similarly to Seguy et al. (2018). $g_\epsilon^\star$ is recovered analytically through 7. Furthermore, from common entropic OT theory, we recover the OT plan, up to a normalization constant, through,

$$\frac{d\gamma_\epsilon^\star}{d(p \otimes q)}(z, z') \propto \exp\left(\frac{f_\epsilon^\star(z) + f_\epsilon^{(c)}(z') - c(z, z')}{\epsilon}\right). \tag{8}$$

From the entropic transport plan $\gamma_\epsilon^\star$ and the dual potentials $(f_\epsilon^\star, f_\epsilon^{(c)})$, we can retrieve the notion of *barycentric mapping* as in (Pooladian & Niles-Weed, 2021, Equations 12 and 13),

$$T_\epsilon^\star(z) = \mathbb{E}_{z'\sim\gamma_\epsilon^\star(\cdot|z)}[z'] = \frac{1}{Z(z)} \int z' \exp\left(\frac{f_\epsilon^{(c)}(z') - c(z, z')}{\epsilon}\right) dq(z'), \tag{9}$$

$Z = \int \exp(({f_\epsilon^{(c)}(z') - c(z,z')})/{\epsilon}) dq(z')$ and $\gamma(\cdot|z)$ denote the conditional law of z' given z under γ.

The central question of this paper is the computation of Wasserstein barycenters. Given measures $p_{1:K} \subset \mathcal{P}_2(\Omega)$, *barycentric coordinates* $\lambda \in \Delta_K$, the barycenter problem is,

$$q_\lambda^\star = \text{Bar}(\lambda, p_{1:K}) = \inf_{q\in\mathcal{P}_2(\Omega)} \sum_{k=1}^K \lambda_k W_2(p_k, q)^2, \tag{10}$$

Introduced by Agueh & Carlier (2011), this problem admits minimizers for any $p_{1:K} \subset \mathcal{P}_2(\Omega)$, and a unique solution when some p_k with $\lambda_k > 0$ vanishes on small sets. To ensure uniqueness for every $\lambda \in \Delta_K$, we further assume that all p_k vanish on small sets, and refer to $q_\lambda^\star$ as *the barycenter*. Álvarez-Esteban et al. (2016) introduced a fixed-point algorithm for Wasserstein barycenters, through the operator,

$$G : \mathcal{P}_2(\Omega) \to \mathcal{P}_2(\Omega), \text{ such that } G(q) = \left(\sum_{k=1}^K \lambda_k T_k^\star\right)_\sharp q, \tag{11}$$

where $T_k^\star$ is the Monge map pushing q to p_k, well-defined for q vanishing on small sets.

2.2 Conditional Flow Matching

CFM (Lipman et al., 2022) is a generative modeling technique that learns a flow from a source measure p to a target measure q. In brief, it models a flow of particles via $\phi(z, t)$, driven by a vector field $v(z, t)$ through $\partial_t \phi(z, t) = v(\phi(z, t), t)$, where $\phi(z, 0) = z$. This flow defines a curve in $\mathcal{P}_2(\Omega)$, via $\rho_t = \phi(\cdot, t)_\sharp p$, with $\rho_0 = p$ and $\rho_1 = q$. In this setting, ρ_t suffices the continuity equation, $\partial_t \rho_t = -\text{div}(\rho_t v_t)$ with velocity field $v_t = v(\cdot, t)$, and div denotes the divergence operator.

At this point, the main question is how to determine v_t. Formally, $v_t(z) = \mathbb{E}[v_t(z_1|z_0)|z_t = z]$, where $v_t(z_1|z_0) = z_1 - z_0$ denotes the conditional velocity of a pair (z_0, z_1), and $z_t = (1-t)z_0 + tz_1$ is the linear interpolant. This conditional expectation is intractable because it requires knowing the

posterior over (z_0, z_1) given z_t. Based on this limitation, Lipman et al. (2022) introduced CFM, which learns a neural net $v_\theta(z_t, t)$ with parameters θ by the following objective,

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,z_0,z_1}[\|v_\theta(z_t,t) - (z_1 - z_0)\|_2^2], \text{ where } t \sim \mathcal{U}[0,1], z_0 \sim p_0, z_1 \sim p_1. \tag{12}$$

Here, Lipman et al. (2022) imposes $p_0 = \mathcal{N}(0, I)$. CFM flows pure noise to a structured, complex measure. In a later development, Tong et al. (2023) relaxed the assumption that $z_0 \sim p_0 = \mathcal{N}(0, I)$ by using a coupling $\gamma \in \Gamma(p, q)$,

$$\mathcal{L}_{\text{CFM}}(\theta;\gamma) = \mathbb{E}_{t,z_0,z_1}[\|v_\theta(z_t,t) - (z_1 - z_0)\|_2^2], \text{ where } t \sim \mathcal{U}[0,1], \text{ and } (z_0, z_1) \sim \gamma. \tag{13}$$

For $\gamma = p \otimes q$, equation 13 is equivalent to 12. More interestingly, using an OT coupling, $(z_0, z_1) \sim \gamma^\star = \text{OT}(p_0, p_1)$ leads to the so-called OT-CFM, $\mathcal{L}_{\text{OT-CFM}}(\theta) = \mathcal{L}_{\text{CFM}}(\theta;\gamma^\star)$. This construction allows building a flow between an arbitrary p_0 and p_1.

3 BARYFM: AMORTIZED, UNIVERSAL WASSERSTEIN BARYCENTERS

In this section, we present our `BaryFM` strategy. In a nutshell, our goal is to build an end-to-end amortized barycenter sampler. Our architecture is based on a 2 head network. The first head estimates the OT problem between multiple marginals $p_1, \cdots, p_K$ and the barycenter $q_\lambda^{(L_{\text{FP}})}$. The second head estimates a velocity field, moving samples from the marginals $p_1, \cdots, p_K$ to the estimated barycenter $\hat{q}_\lambda^{(L_{\text{FP}})}$. We provide an illustration of this architecture in Figure 2.

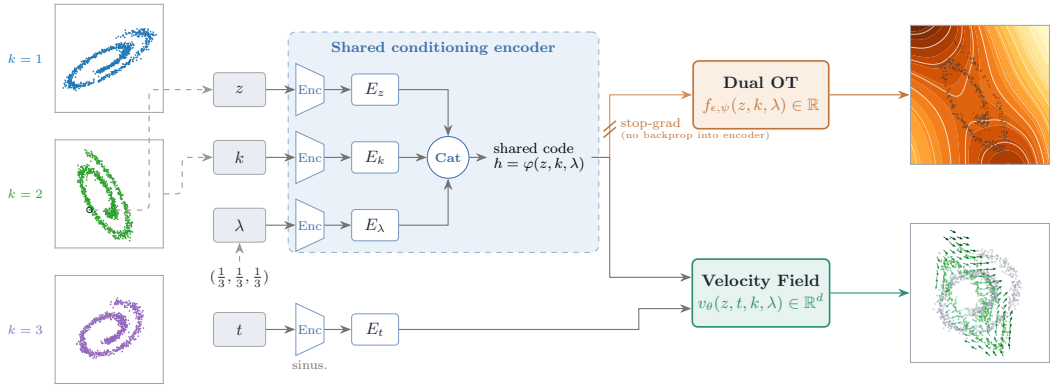


Figure 2: BaryFM architecture. A shared encoder maps (z, k, λ) into a latent code h, which is used by the velocity field estimation branch, and the dual OT branch.

Architecture. Our method is a single network, composed of a shared encoder and 2 task-specific heads. Our flow has K marginals, $p_{1:K}$, and a moving target $\hat{q}_\lambda^{(L_{\text{FP}})}$ (c.f. equation 16) since λ is free. Therefore, we condition the network on (k, λ). We thus build a shared code $h = \text{Cat}(E_z, E_k, E_\lambda)$, where Cat denotes the concatenation operation, and E_z (resp. k, λ) denote the embedding of $z \in \Omega$, $E_z = \text{Enc}(z)$. Our first head, $f_{\epsilon,\psi}$, approximates the dual OT variable $f_\epsilon^\star$ of the pair $(p_k, \hat{q}_\lambda^{(L_{\text{FP}})})$. We estimate the 2nd dual via the $c-$transform (c.f. equation 7). The second head, v_θ, estimates the velocity field flowing p_k into $\hat{q}_\lambda^{(L_{\text{FP}})}$. Its input consists of $h_t = \text{Cat}(h, E_t)$, where $E_t = \text{Enc}(t)$ is a sinusoidal positional encoding (Vaswani et al., 2017). Therefore, our design is different from (Fan et al., 2020, Appendix B), who adds λ as an input to the $(2 \times K + 1)-$ICNNs.

The Dual OT Estimation. The dual OT head estimates the Kantorovich potential of the OT problem involving $(p_k, \hat{q}_\lambda^{(L_{\text{FP}})})$. This is a *multi-source, moving target problem*, since λ is not fixed, which forces $f_{\epsilon,\psi}$ to depend on λ. We learn the parameters $\psi^\star$ of $f_{\epsilon,\psi}$ via stochastic gradient ascent (Genevay et al., 2016; Seguy et al., 2018) of the dual loss,

$$\mathcal{L}_D(\psi, \lambda) = \sum_{k=1}^{K} \lambda_k \Big(\mathbb{E}_z[f_{\epsilon,\psi}(z,k,\lambda)] + \mathbb{E}_{\bar{z}}[f_{\epsilon,\psi}^{(c)}(\bar{z},k,\lambda)] - \epsilon(\mathbb{E}_{z,\bar{z}}[\gamma_{\epsilon,\psi}(z,\bar{z},k,\lambda)] - 1) \Big), \tag{14}$$

where we sample $\bar{z} \sim \hat{q}_\lambda^{(L_{\text{FP}})}$, $z \sim p_k$ and $(z, \bar{z}) \sim p_k \otimes \hat{q}_\lambda^{(L_{\text{FP}})}$. The transport plan $\gamma_{\epsilon,\psi}$ is defined via equation 8, and $f_{\epsilon,\psi}^{(c)}$ is the c-transform of $f_{\epsilon,\psi}$ (c.f. equation 7).

Estimating the CFM Targets. The dual (c.f. equation 14) and CFM (c.f. equation 17) losses depend on the samples from the approximate barycenter measure, $\bar{z}$. Ideally, these samples would be acquired by iterating Álvarez-Esteban et al. (2016) operator $q_\lambda^{(\ell+1)} = G(q_\lambda^{(\ell)})$. That is not possible, since they depend on the Monge maps $T_k^\star$, pushing $q_\lambda^{(\ell)}$ to p_k. These mappings are not available in closed-form, which makes the computation of G intractable. An approximation for $T_k^\star$ is available, via the barycentric mapping (c.f. equation 9),

$$\hat{T}_{\epsilon,\psi}(\bar{z}_i^{(\ell)}, k, \lambda) = \mathop{\mathbb{E}}_{z\sim\gamma_{\epsilon,\psi}(\cdot|\bar{z}_i^{(\ell)},k,\lambda)}[z], \quad \frac{d\gamma_{\epsilon,\psi}}{dp_k}(z|\bar{z}, k, \lambda) \propto \exp\left(\frac{f_{\epsilon,\psi}(z,k,\lambda) - c(z,\bar{z})}{\epsilon}\right), \quad (15)$$

In practice, given a batch $\{z_{k,j}\}_{j=1}^B$, $z_{k,j} \sim p_k$, the conditional expectation in equation 15 is estimated via a softmax $\hat{T}_{\epsilon,\psi}(\bar{z}_i^{(\ell)}, k, \lambda) = \sum_j \text{softmax}_j((f_{\epsilon,\psi}(z_{k,j}, k, \lambda) - c(z_{k,j}, \bar{z}_i^{(\ell)}))/\epsilon) z_{k,j}$, where softmax_j denotes the softmax operation over the batch index j.

The barycentric map is an approximation of the Monge map, when the latter exists (Deb et al., 2021; Pooladian & Niles-Weed, 2021), for $\epsilon \to 0$. Effectively, we have,

$$\bar{z}_i^{(\ell+1)} = \text{sg}\left(\sum_{k=1}^K \lambda_k \hat{T}_{\epsilon,\psi}(\bar{z}_i^{(\ell)}, k, \lambda)\right), \quad (16)$$

$$\hat{q}_\lambda^{(\ell+1)} = \hat{G}_{\epsilon,\psi}(\hat{q}_\lambda^{(\ell)}), \quad \hat{G}_{\epsilon,\psi}(q) = \left(\sum_{k=1}^K \lambda_k \hat{T}_{\epsilon,\psi}(\cdot, k, \lambda)\right)_\sharp q$$

where $\bar{z}_i^{(0)} = \sum_{k=1}^K \lambda_k z_{k,i}$, $\hat{q}_\lambda^{(\ell)}$ denotes an empirical measure with B samples in its support $\{\bar{z}_i^{(\ell)}\}_{i=1}^B$, and sg denotes the stop-grad operation. The dual OT head amortizes OT, that is, we avoid solving K OT problems for estimating $q_\lambda^{(L_{\text{FP}})}$. An illustration is shown in Figure 3.

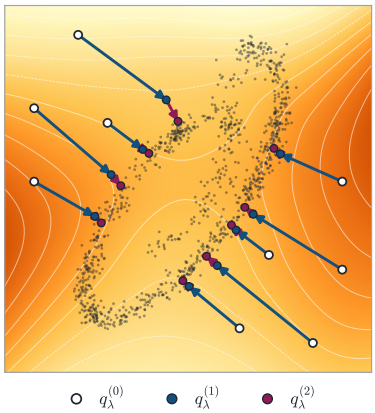


Figure 3: Illustration of the convergence of the fixed-point iterations using the neural barycentric map in equation 15.

Conditional Flow Matching. We follow Tong et al. (2023), by drawing pairs $(z_0, z_1) \sim \gamma_{\epsilon,\psi}(\cdot, \cdot, k, \lambda) \in \Gamma(p_k, \hat{q}_\lambda^{(L_{\text{FP}})})$. More specifically, we sample $z_0 \sim p_k$, $z_1 \sim \gamma_{\epsilon,\psi}(\cdot|z_0, k, \lambda)$,

$$\frac{d\gamma_{\epsilon,\psi}}{d\hat{q}_\lambda^{(L_{\text{FP}})}}(\bar{z}|z_0, k, \lambda) \propto \exp\left(\frac{f_{\epsilon,\psi}^{(c)}(\bar{z}, k, \lambda) - c(z_0, \bar{z})}{\epsilon}\right),$$

which, again, is estimated via a softmax. We provide an illustration of the sampling strategy just described in Figure 4. Given this sampling strategy, we minimize,

$$\mathcal{L}_C(\theta, \lambda, \psi^\star) = \mathop{\mathbb{E}}_{t\sim\mathcal{U}[0,1], k\sim\text{Cat}(\lambda), z_0\sim p_k, z_1\sim\gamma_{\epsilon,\psi^\star}(\cdot|z_0,k,\lambda)}\left[\|v_\theta(z_t, t, k, \lambda) - (z_1 - z_0)\|_2^2\right], \quad (17)$$

where $z_t = (1-t)z_0 + tz_1$. We illustrate the sampling strategy in Figure 4.

Barycentric Flow Matching. In summary, we put forth a shared encoder for the variables (z, k, λ). (k, λ) are necessary to make the flow and OT problems tractable under a *multi-source, moving target* setting. On top of the shared encoder, we use two task-specific heads: a *velocity head* $v_\theta(z, t, k, \lambda)$ that takes the embeddings of (z, k, λ) and t and produces the velocity moving the samples from p_k towards $q_\lambda^{(L_{\text{FP}})}$, and a dual OT head $f_\psi(z, k, \lambda)$, taking the embeddings of (z, k, λ) and producing OT objects, such as conditional OT plans and barycentric mappings. We introduced two losses, $\mathcal{L}_{\text{D}}(\psi, \lambda)$ and $\mathcal{L}_{\text{C}}(\theta, \lambda)$, which are optimized w.r.t. neural net parameters (θ, ψ),

$$\psi^\star = \mathop{\arg\max}_{\psi\in\Psi} \mathbb{E}_{\lambda\sim\text{Dir}(\alpha)}[\mathcal{L}_D(\psi, \lambda)], \qquad \theta^\star = \mathop{\arg\min}_{\theta\in\Theta} \mathbb{E}_{\lambda\sim\text{Dir}(\alpha)}[\mathcal{L}_C(\theta, \lambda; \psi^\star)], \quad (18)$$

where the expectation over $\lambda \sim \text{Dir}(\alpha)$ ensures v_θ and f_ψ are trained over the whole simplex. Our training strategy is summarized in Algorithm 1. (θ, ψ) are learned jointly via block-coordinate updates. At each iteration, ψ is updated to maximize $\mathcal{L}_D$, which is independent of θ, and θ is updated to minimize $\mathcal{L}_C$ for a fixed ψ, which boils down to conventional OT-CFM.

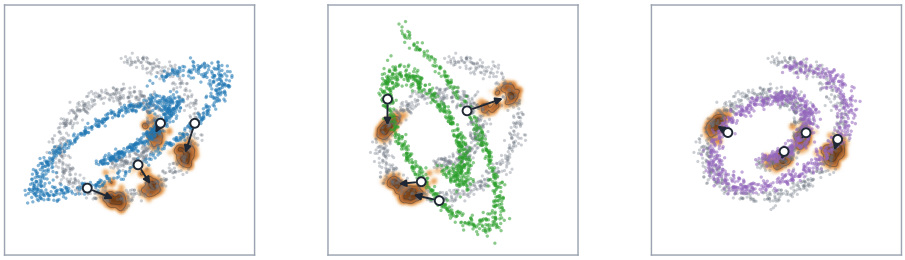


Figure 4: Sampling of pairs (z_0, z_1) for the CFM training. In each panel, $z_0 \sim p_k$, $k = 1, \cdots, 3$ respectively. The amber shaded region corresponds to $\gamma_{\epsilon,\psi}(\cdot|z_0, k)$, from which z_1 is sampled.

Algorithm 1 `BaryFM`: Barycentric Flow Matching Training

Input: Measures $p_1, \cdots, p_K$, velocity field v_θ, dual potential f_ψ, batch size B, iterations N, # of fixed-point iterations L_{FP}, # of dual iterations L_{Dual}, entropic regularization ϵ

for iter $= 1, \ldots, N$ **do**
  Sample $\lambda \sim \text{Dir}(\alpha)$ and $\{\{z_{k,j}\}_{j=1}^{B}\}_{k=1}^{K}$, where $z_{k,j} \sim p_k$
  Initialize $\bar{z}_i^{(0)} \leftarrow \sum_{k=1}^{K} \lambda_k z_{k,i}$
  for $\ell = 1, \cdots, L_{\text{FP}}$ **do**
    $\bar{z}_i^{(\ell)} \leftarrow \text{sg}\left(\sum_{k=1}^{K} \lambda_k \hat{T}_{\epsilon,\psi}(\bar{z}_i^{(\ell-1)}, k, \lambda)\right)$ ▷ No backprop through ψ
  end for
  for $s = 1, \ldots, L_{\text{Dual}}$ **do**
    $\mathcal{L}_{\text{D}} \leftarrow \sum_{k=1}^{K} \lambda_k \left(\mathbb{E}_z[f_{\epsilon,\psi}(z, k, \lambda)] + \mathbb{E}_{\bar{z}}[f^{(c)}_{\epsilon,\psi}(\bar{z}, k, \lambda)] - \epsilon(\mathbb{E}_{z,\bar{z}}[\gamma_{\epsilon,\psi}(z, \bar{z}, k, \lambda)] - 1)\right)$,
    $\psi \leftarrow \psi + \text{LR}_D \nabla_\psi \mathcal{L}_{\text{D}}$
  end for
  $\mathcal{L}_{\text{C}} \leftarrow \mathbb{E}_{t\sim\mathcal{U}[0,1], k\sim\text{Cat}(\lambda), z_0\sim p_k, z_1\sim\gamma_{\epsilon,\psi}(\cdot|z_0,k,\lambda)}\left[\|v_\theta(z_t, t, k, \lambda) - (z_1 - z_0)\|_2^2\right]$
  $\theta \leftarrow \theta - \text{LR}_{\text{C}} \nabla_\theta \mathcal{L}_{\text{C}}$
end for
return θ, ψ

Inference. Once trained, we can sample from any measure in $\mathcal{W}(p_{1:K}) = \{\text{Bar}(\lambda, p_{1:K}) : \lambda \in \Delta_K\}$ by using the velocity field $v_\theta(z, t, k, \lambda)$. For instance, using the forward Euler discretization of the Ordinary Differential Equation (ODE) $\partial_t z(t) = v_\theta(z, t, k, \lambda)$, we have

$$z_{r+1,i} = z_{r,i} + v_\theta(z_{r,i}, rdt, k, \lambda)dt, \quad \text{where } r = 0, \cdots, T-1,\ dt = T^{-1}, \text{ and } z_{0,i} \sim p_k. \tag{19}$$

In practice, we are integrating the velocity field with $\phi(z, t, k, \lambda) = z + \int_0^t v_\theta(\phi(z, s, k, \lambda), s, k, \lambda)ds$. An illustrative example is shown in Figure 5. The main advantage of this technique is amortizing the sampling of the barycenter support with respect particles z, and barycentric coordinates $\lambda \in \Delta_K$, that is, our method is, by design, *universal*.

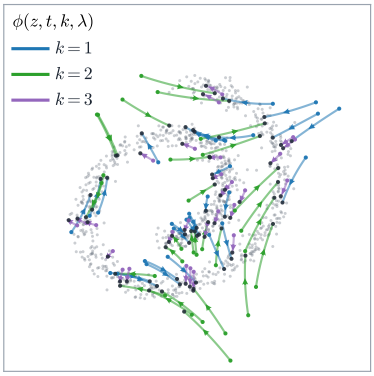


Figure 5: Inference. One ODE solve carries each $z \sim p_k$ (coloured) along the paths shown to $\phi(z, 1, k, \lambda)$ (dark), landing on the barycenter support (gray).

Flows over joint measures. Neural methods so far have treated the case of barycenters over $\Omega = \mathbb{R}^d$. In this work we take a step forward, and define a flow over $\Omega = \mathbb{R}^d \times \Delta_C$, i.e., the space over features $\times$ soft-labels joint variables. This has been previously done for empirical solvers (Montesuma et al., 2023; 2026a), and is made possible in the continuous case by flow matching. We let $z = (x, y)$, so the flow is defined over $\Omega = \mathbb{R}^d$ and Δ_C with metric,

$$d(z, z')^2 = \|x - x'\|_2^2 + \beta\|y - y'\|_2^2, \tag{20}$$

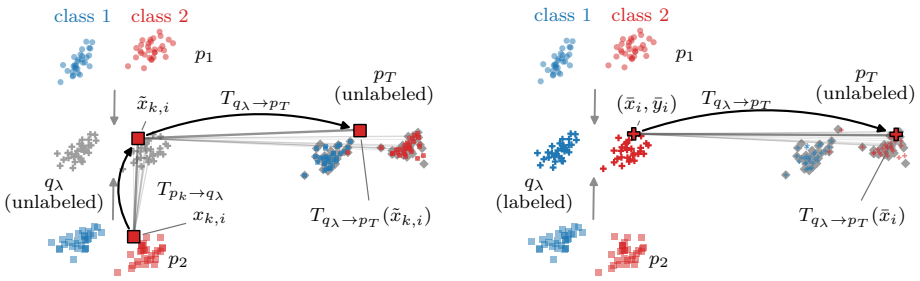
where $\beta > 0$ is an importance factor for the label terms. Furthermore, since $\hat{T}_{\epsilon,\psi}$ is a convex combination, labels stay in the simplex when computing the CFM targets. The only difference comes when solving the ODE in equation 19, where we add a projection step $\pi_\Omega(z_{r,i}+v_\theta(z_{r,i}, rdt, k, \lambda)dt)$, where $\pi_\Omega(z) = \pi_\Omega(x, y) = (x, \pi_{\Delta_C}(y))$, and $\pi_{\Delta_C}(y)$ denotes the orthogonal projection into Δ_C.

4 Experiments

Our experiments compare different barycenter solvers over downstream tasks. In the main paper, we cover domain adaptation (Section 4.1) and generalization (Section 4.2), following the methodologies established in Montesuma & Mboula (2021) and Wang et al. (2020). In our appendix, we extend our experiments to the Swiss roll measures of Korotin et al. (2021) (Section D), Bayesian posterior aggregation (Section E) and algorithmic fairness (Section F).

4.1 Domain Adaptation

In this experiment, we test the ability of Wasserstein barycenter solvers in the context of multi-source domain adaptation (Sun et al., 2015), where a model is learned from multiple source domains $p_{1:K}$, and tested in an unlabeled target domain p_T, hence, the i.i.d. hypothesis underlying Empirical Risk Minimization (ERM) (Vapnik, 2013) does not hold, a case known as distribution or dataset shift (Quiñonero-Candela et al., 2008). We follow the Wasserstein Barycenter Transport (WBT) pipeline (Montesuma & Mboula, 2021), a method that aligns the multiple sources to the target via an intermediate barycentric domain. This pipeline was originally proposed for synthesized, labeled barycenter. For comparing unsupervised barycenters, we use the variant proposed in Montesuma et al. (2026a). An example is shown in Figure 6. Therefore, besides ranking the alignment proposed by barycenter solvers, this task rewards barycenters that respect the underlying class structure.



(a) Unsupervised WBT

(b) Supervised WBT

Figure 6: Unsupervised (a, $\mathcal{X} \times \mathcal{Y} =$✗), and supervised (b, $\mathcal{X} \times \mathcal{Y} =$✓) WBT methodologies.

We test barycenter methods in 10 benchmarks, along computer vision, neuroscience and chemical engineering: Office 31 (Saenko et al., 2010), Office Home (Venkateswara et al., 2017), Domain Net (Peng et al., 2019), BCI-CIV-2a (Brunner et al., 2008), FACED (Chen et al., 2023), SEED-VIG (Zheng & Lu, 2017), SHU-MI (Ma et al., 2022), Mumtaz (Mumtaz, 2016), Physio (Schalk et al., 2004), TEP (Reinartz et al., 2021; Montesuma et al., 2026b). In total, we test 11 methods, with a total of 14 variants and the source-only baseline. Empirical solvers: discrete (Cuturi & Doucet, 2014) and WGF (Montesuma et al., 2026a). Neural solvers: NWB (Fan et al., 2020), CW2B (Korotin et al., 2021), WIN (Korotin et al., 2022a), NOT (Kolesov et al., 2024), U-NOT (Gazdieva et al., 2024) and NormFlow (Visentin & Cheridito, 2025). Diffusion based: TDSB (Noble et al., 2023) and TSBM (Howard et al., 2025). Parametric: GMM (Montesuma et al., 2024b). We show the average rank of barycenter methods on these benchmarks in Figure 7, and report the detailed results in Appendix G.

From Figure 7 (left), we see that `BaryFM` achieves the best average rank over the tested benchmarks, followed by `WGF` (Montesuma et al., 2026a). Our `BaryFM` consistently achieves top 1 in 6 out of 10 benchmarks (c.f. Table 13), being consistent on all benchmarks. In Figure 7 (right), we ablate different components of `BaryFM`, especially each head, and the universal training. We go into

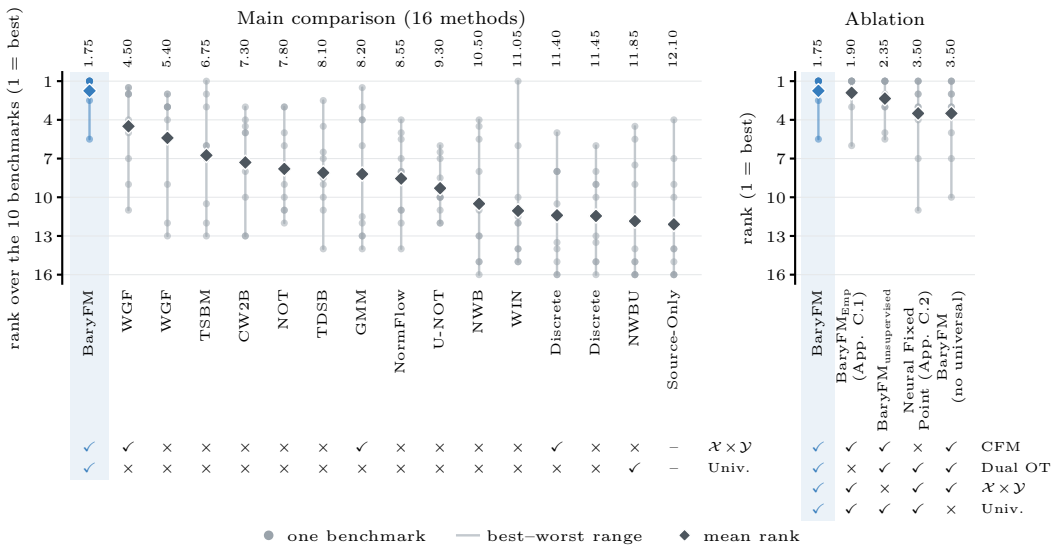


Figure 7: Comparison of BaryFM with 15 methods, and ablation of BaryFM components. We report the average rank over 10 benchmarks. Detailed results are shown in Table 13 and Appendix G.

further detail on these variants in Appendix C. Overall, the components that have the most impact in BaryFM performance are the CFM head, which amortizes the sampling from $q_\lambda^\star$, and universality.

4.2 Domain Generalization

Domain generalization (Gulrajani & Lopez-Paz, 2020) seeks to generalize to an unseen domain from multi-domain source data $p_{1:K}$, with no target samples available. We tackle it through *data augmentation*(Zhong et al., 2025), that is, we generate data blending the source domains, and minimize

$$\theta^\star = \arg\min_{\theta\in\Theta} \frac{1-\rho}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n}\ell(y_{k,i}, f_\theta(x_{k,i})) + \frac{\rho}{mK'}\sum_{k'=1}^{K'}\sum_{j=1}^{m}\ell(\tilde{y}_{k',j}, f_\theta(\tilde{x}_{k',j})), \tag{21}$$

where ℓ is a loss function, f_θ is a neural net, such as a ResNet 50 (He et al., 2016), and $(\tilde{x}_{k',j}, \tilde{y}_{k',j})$ is an augmented sample. In this experiment the augmented samples are obtained either through (1) MixUp (Zhang et al., 2017), (2) Domain MixUp (Wang et al., 2020), (3) BaryFM$_{\text{Random}}$ or (4) BaryFM$_{\text{Vertex}}$ (both ours). In the first case, $\lambda_{k'} \sim \text{Beta}(\alpha, \alpha)$, the 2nd and 3rd cases $\lambda_{k'} \sim \text{Dir}(\alpha)$. The 4th case is different, as we sample from the vertices of $\mathcal{W}(p_{1:K})$, meaning, $\lambda_{k'} = e_{k'}$, $k' = 1, \cdots, K$. More importantly, MixUp and its variants mixes samples linearly, while BaryFM mixes samples nonlinearly through Wasserstein barycenters. Using Wasserstein barycenters for the augmentation task demands 3 properties of solvers: (i) *amortization over samples*, meaning, we must sample $z \sim \hat{q}_\lambda^{(L_{\text{FP}})}$, (ii) *universality*, meaning, we must be able to sample from $\hat{q}_\lambda^{(L_{\text{FP}})}, \forall\lambda \in \Delta_K$ and (iii) joint barycenters, meaning, ours samples must be $z = (x, y)$, $x \in \mathbb{R}^d$, $y \in \Delta_C$. To the best of our knowledge, BaryFM is the only solver satisfying all three (Table 2).

To better capture the semantics of complex data, we perform augmentations in a latent space. For image data, we use a Variational Autoencoder (VAE) (Kingma & Welling, 2013; Rombach et al., 2022), while for EEG we use Principal Component Analysis (PCA) fit on source domains. An illustration of our approach is shown in Figure 8. Further details are available in Appendix H.

Our results are summarized in Table 1, which reports the downstream performance of ResNets (He et al., 2016) and CBraMod (Wang et al., 2025) finetuned on 5 multi-domain benchmarks. On average, BaryFM$_{\text{Random}}$ ranks best (1.40), against BaryFM$_{\text{Vertex}}$ (2.80), Domain MixUp (3.00), MixUp (3.20) and ERM (4.60). Overall, these results illustrate a couple of findings. First, augmentations broadly help generalization to new domains. Second, interpolating under the Wasserstein geometry (our BaryFM vs. MixUp variants) generalizes better. Third, sampling from the Wasserstein simplex

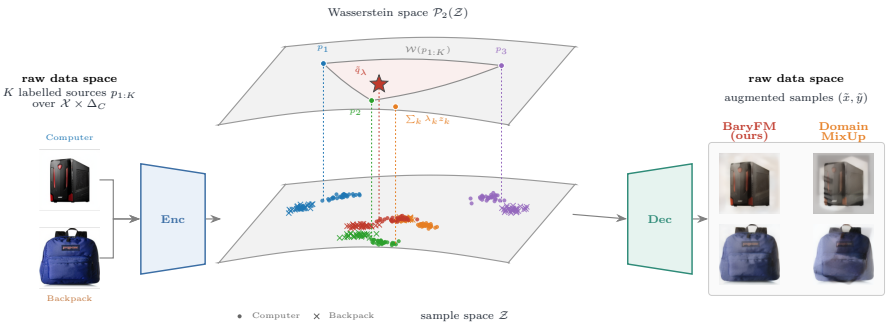


Figure 8: Data augmentation strategy for domain generalization. On the left, samples from different domains are encoded into a common latent space. In the center, synthetic samples are generated in the Wasserstein simplex of source domains. On the right, the latent synthetic samples are decoded back into raw data space.

($\text{BaryFM}_{\text{Random}}$ vs. $\text{BaryFM}_{\text{Vertex}}$) works better overall. These findings illustrate the usefulness of an *universal, amortized, joint* barycenter algorithm for domain generalization.

Method	Office 31	Office Home	DomainNet	BCI-CIV-2a	SEED-VIG	Avg. Rank
Backbone	ResNet 50	ResNet 50	ResNet 101	CBraMod	CBraMod	-
Latent Space		VAE		PCA		-
ERM	$84.38_{\pm 0.53}$	$69.74_{\pm 0.57}$	$49.57_{\pm 0.24}$	$50.63_{\pm 3.45}$	$58.30_{\pm 2.44}$	4.60
MixUp	$85.66_{\pm 0.69}$	$70.43_{\pm 0.34}$	$49.53_{\pm 0.11}$	$52.37_{\pm 3.11}$	$61.11_{\pm 2.57}$	3.20
Domain MixUp	$84.97_{\pm 0.49}$	$68.86_{\pm 0.82}$	$\mathbf{50.63}_{\pm 0.20}$	$52.70_{\pm 2.78}$	$61.04_{\pm 3.15}$	3.00
$\text{BaryFM}_{\text{Vertex}}$	$84.89_{\pm 0.58}$	$70.55_{\pm 0.47}$	$49.69_{\pm 0.34}$	$\mathbf{54.18}_{\pm 3.48}$	$59.25_{\pm 5.63}$	2.80
$\text{BaryFM}_{\text{Random}}$	$\mathbf{85.94}_{\pm 0.66}$	$\mathbf{71.17}_{\pm 0.76}$	$50.19_{\pm 0.21}$	$53.73_{\pm 3.43}$	$\mathbf{62.08}_{\pm 2.38}$	1.40

Table 1: Downstream performance of data augmentation strategies on domain generalization benchmarks. Results denote $\mu_{\pm\sigma}$ over seeds (c.f. Section H).

5 Conclusion

In this paper, we study the *universal Wasserstein barycenter problem*, that is, approximating the family of measures in the *Wasserstein simplex* (c.f. Equation 1), which contains the Wasserstein barycenters $\text{Bar}(\lambda, p_{1:K})$ for general $\lambda \in \Delta_K$. We propose a new method based on a neural network, composed of a shared encoding layer, and 2 heads (c.f. Figure 2): a dual OT head approximating the OT problem between each p_k and the barycenter target $q^\star_\lambda$, and a flow matching (Lipman et al., 2022) head, modeling the flow between the aforementioned measures. To the best of our knowledge, our method is the first universal, neural method that can handle flows over feature-label joint measures.

We validate our proposed `BaryFM` on 4 downstream applications of Wasserstein barycenters, namely, domain adaptation (Montesuma & Mboula, 2021) (c.f. Section 4.1), generalization (c.f. Section 4.2), algorithmic fairness (Gordaliza et al., 2019) (c.f. Appendix F) and Bayesian posterior aggregation (Srivastava et al., 2018) (c.f. Appendix E). Overall, in domain adaptation, our method has the best average rank among 15 tested methods, over 10 benchmarks. Our method enables a new application of barycenter solvers, that is domain generalization. We demonstrate empirically that augmenting the source domain data with Wasserstein barycenters improves performance over the ERM baseline, and is better, or competitive against Domain MixUp (Wang et al., 2020).

Limitations. As we cover in Appendix B, our method computes a *regularized* barycenter, which carries inherent biases with respect the true Wasserstein barycenter. While some of these biases (mini-batching, entropic regularization) serve as a way of making the barycenter problem tractable, further analysis on how the computed barycenter differs from the ground-truth is needed, and is the subject of future work. In restricted cases (e.g., Location-Scatter families in Appendix D) we show that our method remains accurate to a known ground-truth despite these biases.

AI USE STATEMENT

We used generative AI tools solely to aid or polish writing. More specifically, we used these tools to proofread and detect notational inconsistencies and undefined symbols, in the attempt of improving the clarity of the authors' writing. These tools **were not used** for retrieval and discovery of related work, nor for research and ideation, nor for drafting sections of the paper, nor for generating synthetic datasets, nor for proving mathematical claims. All AI-assisted text was reviewed and edited by the authors, who take full responsibility for the content and claims of this paper.

ETHICS STATEMENT

This work does not involve the collection of new human-subject data. Our experiments rely on publicly available datasets, used in the literature for benchmarking domain adaptation, generalization, Bayesian posterior aggregation and algorithmic fairness. Some of these datasets, such as BCI-CIV-2a (Brunner et al., 2008), FACED (Chen et al., 2023), SEED-VIG (Zheng & Lu, 2017), Mumtaz (Mumtaz, 2016), SHU-MI (Ma et al., 2022), Physio (Schalk et al., 2004) include electroencephalography recordings, and we follow the data usage and access conditions associated with each of them. We refer readers to each dataset reference for more information about the ethics behind the studies conducted.

REPRODUCIBILITY STATEMENT

We provide the details for the faithful reproducibility of our method in Section 3, especially, we provide a high level description of our method in Algorithm 1. Our appendix has other descriptions of variants of our method done during ablation, especially Section C, and Algorithms 2 and 3. Furthermore, we provide a complete description of the used hyper-parameters for Bayesian posterior aggregation in Section E, Table 6, domain adaptation in Section G, Tables 11 and 12, domain generalization in Section H, Table 28, and Fairness, in Section F.

REFERENCES

Stefan Aeberhard, Danny Coomans, and Olivier De Vel. Comparative analysis of statistical pattern recognition methods in high dimensional settings. *Pattern Recognition*, 27(8):1065–1077, 1994.

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.

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International conference on machine learning*, pp. 146–155. PMLR, 2017.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *Ethics of data and analytics*, pp. 254–264. Auerbach Publications, 2022.

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):1, 2008.

R.L. Burden and J.D. Faires. *Numerical Analysis*. Cengage Learning, 2010. ISBN 9780538733519. URL `https://books.google.fr/books?id=zXnSxY9G2JgC`.

Bureau of Labor Statistics, U.S. Department of Labor. National longitudinal survey of youth 1979 cohort, 1979–2016 (rounds 1–27). `https://www.bls.gov/nls/`, 2019. Produced and distributed by the Center for Human Resource Research (CHRR), The Ohio State University, Columbus, OH.

Jingjing Chen, Xiaobin Wang, Chen Huang, Xin Hu, Xinke Shen, and Dan Zhang. A large finer-grained affective computing eeg dataset. *Scientific Data*, 10(1):740, 2023.

Yongxin Chen, Tryphon T Georgiou, and Allen Tannenbaum. Optimal transport for gaussian mixture models. *IEEE Access*, 7:6269–6278, 2018.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: Dynamic and kantorovich formulations. *Journal of Functional Analysis*, 274(11): 3090–3123, 2018.

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.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pp. 146–158, 1975.

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.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in neural information processing systems*, 34:17695–17709, 2021.

Nabarun Deb, Promit Ghosal, and Bodhisattva Sen. Rates of estimation of optimal transport maps using plug-in estimators via barycentric projections. *Advances in Neural Information Processing Systems*, 34:29736–29753, 2021.

Julie Delon and Agnes Desolneux. A wasserstein-type distance in the space of gaussian mixture models. *SIAM Journal on Imaging Sciences*, 13(2):936–970, 2020.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

James J Downs and Ernest F Vogel. A plant-wide industrial process control problem. *Computers & chemical engineering*, 17(3):245–255, 1993.

Jiaojiao Fan, Amirhossein Taghvaei, and Yongxin Chen. Scalable computations of wasserstein barycenter via input convex neural networks. *arXiv preprint arXiv:2007.04462*, 2020.

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

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.

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. *Advances in neural information processing systems*, 29, 2016.

Paula Gordaliza, Eustasio Del Barrio, Gamboa Fabrice, and Jean-Michel Loubes. Obtaining fairness using optimal transport theory. In *International conference on machine learning*, pp. 2357–2365. PMLR, 2019.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

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.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Samuel Howard, Peter Potaptchik, and George Deligiannidis. Schrödinger bridge matching for tree-structured costs and entropic wasserstein barycentres. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=DliPKnn6e0.

Leonid V Kantorovich. On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pp. 199–201, 1942.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Ron Kohavi et al. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pp. 202–207, 1996.

Alexander Kolesov, Petr Mokrov, Igor Udovichenko, Milena Gazdieva, Gudmund Pammer, Evgeny Burnaev, and Alexander Korotin. Estimating barycenters of distributions with neural optimal transport. *arXiv preprint arXiv:2402.03828*, 2024.

Alexander Korotin, Lingxiao Li, Justin Solomon, and Evgeny Burnaev. Continuous wasserstein-2 barycenter estimation without minimax optimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3tFAs5E-Pe.

Alexander Korotin, Vage Egiazarian, Lingxiao Li, and Evgeny Burnaev. Wasserstein iterative networks for barycenter estimation. *Advances in Neural Information Processing Systems*, 35:15672–15686, 2022a.

Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Neural optimal transport. *arXiv preprint arXiv:2201.12220*, 2022b.

Matthias Liero, Alexander Mielke, and Giuseppe Savaré. Optimal entropy-transport problems and a new hellinger–kantorovich distance between positive measures. *Inventiones mathematicae*, 211 (3):969–1117, 2018.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jun Ma, Banghua Yang, Wenzheng Qiu, Yunzhe Li, Shouwei Gao, and Xinxing Xia. A large eeg dataset for studying cross-session variability in motor imagery brain-computer interface. *Scientific Data*, 9(1):531, 2022.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pp. 666–704, 1781.

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*, pp. 16785–16793, 2021.

Eduardo Fernandes Montesuma, Fred Ngolè Mboula, and Antoine Souloumiac. Multi-source domain adaptation through dataset dictionary learning in wasserstein space. *arXiv preprint arXiv:2307.14953*, 2023.

Eduardo Fernandes Montesuma, Fred Maurice Ngole Mboula, and Antoine Souloumiac. Recent advances in optimal transport for machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(2):1161–1180, 2024a.

Eduardo Fernandes Montesuma, Fred Ngolè 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*, pp. 21–38. Springer, 2024b.

Eduardo Fernandes Montesuma, Yassir Bendou, and Mike Gartrell. Wasserstein gradient flows for scalable and regularized barycenter computation. In Emilija Perković and Daniel Malinsky (eds.), *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pp. 4595–4606. PMLR, 17–21 Aug 2026a. URL https://proceedings.mlr.press/v337/fernandes-montesuma26a.html.

Eduardo Fernandes Montesuma, Michela Mulas, Fred Ngolè Mboula, Francesco Corona, and Antoine Souloumiac. Benchmarking domain adaptation for chemical processes on the tennessee eastman process. In Mattia Cerrato, Danguolė Kalinauskaitė, Mantas Lukoševičius, Mykola Pechenizkiy, and Kristina Šutienė (eds.), *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pp. 307–322, Cham, 2026b. Springer Nature Switzerland. ISBN 978-3-032-25305-7.

Wajid Mumtaz. Mdd patients and healthy controls eeg data (new). figshare. Dataset, 2016. URL https://doi.org/10.6084/m9.figshare.4244171.v2.

Maxence Noble, Valentin De Bortoli, Arnaud Doucet, and Alain Durmus. Tree-based diffusion schrödinger bridge with applications to wasserstein barycenters. *Advances in Neural Information Processing Systems*, 36:55193–55236, 2023.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

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.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3942–3951, 2018.

Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5–6):355–607, 2019. doi: 10.1561/2200000073.

Aram-Alexandre Pooladian and Jonathan Niles-Weed. Entropic estimation of optimal transport maps. *arXiv preprint arXiv:2109.12004*, 2021.

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

Christopher Reinartz, Murat Kulahci, and Ole Ravn. An extended tennessee eastman simulation dataset for fault-detection and decision support systems. *Computers & chemical engineering*, 149:107281, 2021.

R Tyrrell Rockafellar. *Convex analysis*, volume 28. Princeton university press, 1997.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

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.

Gerwin Schalk, Dennis J McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R Wolpaw. Bci2000: a general-purpose brain-computer interface (bci) system. *IEEE Transactions on biomedical engineering*, 51(6):1034–1043, 2004.

Vivien Seguy, Bharath Bhushan Damodaran, Remi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large scale optimal transport and mapping estimation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1zlp1bRW.

Thibault Séjourné, Gabriel Peyré, and François-Xavier Vialard. Unbalanced optimal transport, from theory to numerics. *Handbook of Numerical Analysis*, 24:407–471, 2023.

Sanvesh Srivastava, Volkan Cevher, Quoc Dinh, and David Dunson. Wasp: Scalable bayes via barycenters of subset posteriors. In *Artificial intelligence and statistics*, pp. 912–920. PMLR, 2015.

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.

Shiliang Sun, Honglei Shi, and Yuanbin Wu. A survey of multi-source domain adaptation. *Information Fusion*, 24:84–92, 2015.

Asuka Takatsu. Wasserstein geometry of Gaussian measures. *Osaka Journal of Mathematics*, 48 (4):1005–1026, 2011.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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*, pp. 5018–5027, 2017.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

Gabriele Visentin and Patrick Cheridito. Computing optimal transport maps and wasserstein barycenters using conditional normalizing flows. *arXiv preprint arXiv:2505.22364*, 2025.

Johannes von Lindheim. Simple approximative algorithms for free-support wasserstein barycenters. *Computational Optimization and Applications*, 85(1):213–246, 2023. ISSN 1573-2894. doi: 10.1007/s10589-023-00458-3. URL https://doi.org/10.1007/s10589-023-00458-3.

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. In *International conference on learning representations*, volume 2025, pp. 75310–75346, 2025.

Yufei Wang, Haoliang Li, and Alex C Kot. Heterogeneous domain generalization via domain mixup. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3622–3626. IEEE, 2020.

I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications*, 36(2): 2473–2480, 2009.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Han Zhao and Geoffrey J Gordon. Inherent tradeoffs in learning fair representations. *Journal of Machine Learning Research*, 23(57):1–26, 2022.

Wei-Long Zheng and Bao-Liang Lu. A multimodal approach to estimating vigilance using eeg and forehead eog. *Journal of neural engineering*, 14(2):026017, 2017.

Yingyi Zhong, Wen'an Zhou, and Zhixian Wang. A survey of data augmentation in domain generalization. *Neural Processing Letters*, 57(2):34, 2025.

Supplementary Materials for Towards Universal Wasserstein Barycenters through Flow Matching

CONTENTS

1 Introduction **1**

2 Background **2**

2.1 Optimal Transport and Barycenters 2

2.2 Conditional Flow Matching 3

3 `BaryFM`: Amortized, Universal Wasserstein Barycenters **4**

4 Experiments **7**

4.1 Domain Adaptation 7

4.2 Domain Generalization 8

5 Conclusion **9**

A Introduction **18**

B Related Work **18**

C Alternative Versions of `BaryFM` **20**

C.1 Empirical `BaryFM` 20

C.2 Neural Fixed-Point 21

C.3 Semi-Unbalanced `BaryFM` 23

C.4 Other ODE Integration Strategies 23

D Swiss Roll Measures **24**

E Bayesian Posterior Aggregation **26**

F Fairness **30**

G Domain Adaptation **33**

G.1 Wasserstein Barycenter Transport 33

G.2 Experimental Details 34

G.3 Detailed Results and Dataset Descriptions 38

G.3.1 Office 31 38

G.3.2 Office Home 39

G.3.3 Domain Net 40
G.3.4 BCI-CIV-2a 41
G.3.5 FACED 42
G.3.6 SEED-VIG 43
G.3.7 Mumtaz 44
G.3.8 Physio 45
G.3.9 SHU-MI 46
G.3.10 Tennessee Eastman Process 47

H Domain Generalization **48**
H.1 Data Augmentation for Domain Generalization 48
H.2 Experimental Details 49
H.3 Detailed Results 52

A Introduction

In this appendix, we provide much more information about the implementation, hyper-parameters, and experiments done in our work. Our appendix is organized as follows. Section B discusses how our algorithm fits within the broader literature of Wasserstein barycenters. Section C provides the description of 2 alternative proposed methods to `BaryFM`, namely *empirical* `BaryFM` and `NeuralFP`. These methods are compared in the domain adaptation ablation in Table 14. Next, we denote 5 sections to additional experimental details. Section D covers the Swiss roll toy example. Section E covers Bayesian Posterior Aggregation. Section F covers algorithmic fairness. Section G covers domain adaptation. Section H covers domain generalization. In each of these sections, we provide details about how barycenters are applied to these tasks, as well as detailed results.

B Related Work

Empirical Barycenter Solvers. Empirical barycenter solvers parametrize the barycentric measure, $q_\lambda^\star$ via the samples in their support. This hinders the possibility of drawing new samples from this measure, as each new drawn requires solving the optimization problem from scratch, which by it self limits their scalability. Besides this issue, early algorithms such as Cuturi & Doucet (2014) and (Montesuma et al., 2023, Algorithm 1) assume access to the complete set of samples in the marginal measures. This issue has been partly solved by Montesuma et al. (2026a) who proposed a gradient flow-based algorithm that supports mini-batching.

Parametric Barycenter Solvers. These include Gaussian and Gaussian Mixture approximations to the concerned measures. In the first case, Agueh & Carlier (2011) and Álvarez-Esteban et al. (2016) show that the barycenter can be fold via an iterative fixed-point algorithm with complexity that depends on the dimensionality $\mathcal{O}(d^3)$. While this complexity is sample-free (thus scalable to large-scale datasets), it scales poorly with the data dimensionality. Montesuma et al. (2024b) presents a different approach based on Gaussian Mixture Model (GMM) OT (Chen et al., 2018; Delon & Desolneux, 2020). In practice, the authors propose a method based on GMMs with diagonal covariances, which scales both with number of samples and dimensionality. Nonetheless, as with all parametric models, these algorithms rely on strong hypothesis which are seldom met in real datasets.

Neural Barycenter Solvers. These methods include a recent wave of works in the intersection of machine learning and OT. As such, they rely on different strategies. We cover mainly 3 families of strategies. The first family relies on multi-level optimization. For instance Fan et al. (2020) propose a 3-loop min-max-min problem with $2K+1$ ICNNs. Korotin et al. (2021) avoid the 3-level optimization problem, by combining congruence and cycle-consistency soft constraints over $2K$ ICNNs.

The second family relies on a neural parametrization of the barycenter measure itself. Korotin et al. (2022a) represents $q_\lambda^{(\ell)}$ as the map from a latent space to the barycentric measure. The optimization problem proposed by the authors is a fixed-point strategy inspired in Álvarez-Esteban et al. (2016). Kolesov et al. (2024) and Gazdieva et al. (2024) learn transport maps paired with potentials, under a bi-level max-min objective, the latter being unbalanced by design.

A third family, more similar to our proposed `BaryFM`, parametrizes the dynamics from a initial measure q_0 towards the barycenter $q_\lambda^\star$. For instance, Noble et al. (2023) and Howard et al. (2025) view the barycenter problem through the lens of a start-shaped tree Schrondinger Bridge Matching (SBM), and Visentin & Cheridito (2025) learn a conditional normalizing flow.

Universal Barycenter Solvers. All of the aforementioned families share a limitation: for each new set of barycentric coordinates $\lambda \in \Delta_K$, the corresponding optimization problems need to be solved *from scratch*. For instance, in Korotin et al. (2022a), λ enters the regression targets. In Noble et al. (2023) and Howard et al. (2025), it enters the SBM problem itself. To the best of our knowledge, the only prior *universal* solver is (Fan et al., 2020, Appendix B), who adds λ as an input to the $2K+1$ ICNNs. The authors validated their approach on low-dimensional, toy examples.

We summarize our discussion in Table 2, which shows that most existing methods do not consider the question of approximating barycenters over the Wasserstein simplex. Based on the limitations

of existing methods, and in order to make the overall architecture scalable with respect number of marginals K, samples n and *universal* we use a single network with 2 heads, whose outputs are conditioned on (k, λ), $k \in \{1, \cdots, K\}$ and $\lambda \in \Delta_K$. The closest precedent in the literature is Visentin & Cheridito (2025), who condition their learned OT maps via a domain-index k. While this is similar in spirit to our architecture, the conditioning maps a latent measure to each marginal p_k. Effectively, this means that while our conditioning expresses the OT between $(p_k, q^\star_\lambda)$, theirs concern (q_0, p_k), where $q_0 \in \mathcal{P}_2(\Omega)$ for a latent space Ω.

Method	Reference	Parametrization	Learned object	# nets	# params	Inner opt.	Amort. sampling	Universal	$\mathcal{X} \times \mathcal{Y}$
Free-support (empirical)									
`Discrete`	(Cuturi & Doucet, 2014)	Particles	support	—	nd	—	✗	✗	✗
`DaDiL`	(Montesuma et al., 2023)	Particles	support (atoms)	—	nd	—	✗	✗	✓
`WGF`	(Montesuma et al., 2026a)	Particles	support	—	nd	—	✗	✗	✓
Restricted parametric family									
`GMM-WBT`	(Montesuma et al., 2024b)	GMM	means, covariances	—	md^2	—	✓	✗	✓
Neural: dual potentials									
`NWB`	(Fan et al., 2020)	ICNNs	potentials, map	$2K+1$	$(2K+1)P$	min-max-min	✓	✗	✗
`NWBU`	(Fan et al., 2020, App. B)	ICNNs	potentials, map	$2K+1$	$(2K+1)P$	min-max-min	✓	✓	✗
`CW2B`	(Korotin et al., 2021)	ICNNs	potentials	$2K$	$2KP$	—	✓	✗	✗
Neural: measure and maps									
`WIN`	(Korotin et al., 2022a)	NNs	generator, maps	$2K+1$	$(2K+1)P$	min-max-min	✓	✗	✗
`NOT`	(Kolesov et al., 2024)	NNs	maps, potentials	$2K$	$2KP$	max-min	✓	✗	✗
`U-NOT`	(Gazdieva et al., 2024)	NNs	maps, potentials	$2K$	$2KP$	max-min	✓	✗	✗
Neural: dynamics and flows									
`TDSB`	(Noble et al., 2023)	NNs	Stochastic Differential Equation (SDE) drifts	$2K$	$2KP$	—	✓	✗	✗
`TSBM`	(Howard et al., 2025)	NNs	vector fields	$2K$	$2KP$	—	✓	✗	✗
`NormFlow`	(Visentin & Cheridito, 2025)	NNs	bijections	1	$P + \mathcal{O}(KH)$	—	✓	✗	✗
Ours									
`NeuralFP`	(ours)	NNs	dual potential	1	$P + \mathcal{O}(KH)$	—	✗	✓	✓
`BaryFM`	(ours)	NNs	dual, velocity	1	$P + \mathcal{O}(KH)$	—	✓	✓	✓

Table 2: Comparison of different approaches in the state-of-the-art along parametrization, objects, number of neural nets and parameters, optimization strategy, amortization, universality, and handling of joint measures.

A Taxonomy of Biases. Given the different existing solvers and their inner workings, we provide, in Table 3, a taxonomy of the different *biases* in the barycenter solvers. We understand as biases, different design choices that make algorithms compute a barycenter $\hat{q}^{(L_{\text{FP}})}_\lambda$ that is different from the ground-truth $q^\star_\lambda$. We divide these biases into 3 categories,

- **Input biases** concern the way the marginal measures are approximated. These include mini-batch OT (Fatras et al., 2019) and parametric OT (e.g., Gaussian or GMMs).
- **Algorithmic biases** include several factors, such as entropic OT (Cuturi, 2013), unbalanced OT (Séjourné et al., 2023), expressing the barycenter as a push-forward of a latent measure (as in Korotin et al. (2022a)), relaxing constraints (as in Korotin et al. (2021)) or fixed-point truncation (most methods based on Álvarez-Esteban et al. (2016)).
- **Output biases** which roughly include the time-discretization of an ODE as in ours and Montesuma et al. (2026a).

		Input		Algorithmic					Output
Method	Reference	Mini-Batch OT	Parametric OT	Entropic OT	Unbalanced OT	Latent Pushforward	Constraint Relaxation	Fixed-Pt. Truncation	Time Discretization
`Discrete`	(Cuturi & Doucet, 2014)	✗	✗	✓(✗ $\epsilon = 0$)	✗	✗	✗	✓	✗
`WGF`	(Montesuma et al., 2026a)	✓	✗	✓(✗ $\epsilon = 0$)	✗	✗	✗	✗	✓
`GMM`	(Montesuma et al., 2024b)	✗	✓	✗	✗	✗	✗	✓	✗
`NWB`	(Fan et al., 2020)	✗	✗	✗	✗	✓	✗	✗	✗
`NWBU`	(Fan et al., 2020, App. B)	✗	✗	✗	✗	✓	✗	✗	✗
`CW2B`	(Korotin et al., 2021)	✗	✗	✗	✗	✗	✓	✗	✗
`WIN`	(Korotin et al., 2022a)	✗	✗	✗	✗	✓	✗	✓	✗
`NOT`	(Kolesov et al., 2024)	✗	✗	✗	✗	✗	✗	✗	✗
`U-NOT`	(Gazdieva et al., 2024)	✗	✗	✗	✓	✗	✗	✗	✗
`TDSB`	(Noble et al., 2023)	✗	✗	✓	✗	✗	✗	✓	✓
`TSBM`	(Howard et al., 2025)	✗	✗	✓	✗	✗	✗	✓	✓
`NormFlow`	(Visentin & Cheridito, 2025)	✗	✗	✗	✗	✗	✓	✗	✗
`NeuralFP`	(ours)	✓	✗	✓	✗	✗	✗	✓	✗
`BaryFM`	(ours)	✓	✗	✓	✗	✗	✗	✓	✓

Table 3: Taxonomy of barycenter solvers biases, along input, algorithmic and output biases.

C ALTERNATIVE VERSIONS OF BARYFM

As described in section 3, our method is based on 2 pieces, a dual OT estimator head, and a flow matching head. In our experiments, we ablate these 2 components. In this section, we briefly describe other 2 methods,

- Empirical BaryFM (Section C.1), where we replace the dual estimation head by an inner OT problem approximating the barycenter,
- NeuralFP (Section C.2), where we remove the flow matching component of our algorithm and use the dual OT head in the fixed-point strategy of Álvarez-Esteban et al. (2016).
- Semi-Unbalanced BaryFM (Section C.3), where we change the semi-dual OT loss with the semi-unbalanced objective developed in Gazdieva et al. (2024).

We summarize the different versions of our method in Table 4. Note that these different formulations make an ablation ladder, starting from the discrete barycenter of Cuturi & Doucet (2014), and progressively converging towards BaryFM. We highlight the advantages of each component,

- The **Dual OT Head** allows us to amortize OT between mini-batches from p_k and the moving target $q_\lambda^\star$. Therefore, instead of computing $K-$OT problem at each optimization step, with complexity $\mathcal{O}(Kn^2)$, we are able to compute the OT variables with a forward pass of a neural network.
- The **Velocity Head** allows us to amortize the *sampling* of $q_\lambda^\star$, for an arbitrary $\lambda \in \Delta_K$. This sampling is efficient, since it relies on solving an ODE. In general, it requires T forward passes of the velocity field (c.f. equation 19).

Besides these components, we explore in Section C.3 a semi-unbalanced version of our method inspired by Gazdieva et al. (2024).

Method	Conditional Flow Matching	Neural Dual Estimation	Balanced
Discrete Barycenter	✗	✗	✓
NeuralFP	✗	✓	✓
Empirical BaryFM	✓	✗	✓
BaryFM	✓	✓	✓
Semi-unbalanced BaryFM	✓	✓	✗

Table 4: Summary of different methods proposed in this paper. The discrete barycenter corresponds to the algorithm of Cuturi & Doucet (2014), which does not use neither CFM nor the dual OT head.

C.1 EMPIRICAL BARYFM

This version of our algorithm estimates the barycenter targets with empirical OT. In a nutshell, we initialize the barycenter measure with $\bar{z}_i^{(0)} = \sum_{k=1}^K \lambda_k z_{k,i}$, where $z_{k,i} \sim p_k$. Note that we sample a fixed number of B samples from each marginal measure. For $\ell = 1, \cdots, L$, we progressively update the support of $q_\lambda^{(\ell)}$ via the fixed-point approach of Álvarez-Esteban et al. (2016),

$$\underbrace{\bar{z}_i^{(\ell+1)} = \sum_{k=1}^K \lambda_k \hat{T}_k(\bar{z}_i^{(\ell)})}_{\text{Barycenter support update}}, \text{ where } \underbrace{\hat{T}_k(\bar{z}_i^{(\ell)}) = \sum_{j=1}^B \frac{\gamma_{k,i,j}^{(\ell)}}{\sum_{j'=1}^B \gamma_{k,i,j'}^{(\ell)}} z_{k,j}}_{\text{Barycentric mapping}}, \text{ and } \gamma_k^{(\ell)} = \text{OT}_\epsilon(p_k, q_\lambda^{(\ell)}), \tag{22}$$

where OT_ϵ denotes the Sinkhorn problem (Cuturi, 2013) between empirical measures p_k with support $\{z_{k,j}\}_{j=1}^B$ and barycenter support $\{\bar{z}_i^{(\ell)}\}_{i=1}^B$. Note that equation 22 is an empirical approximation to the mapping G defined in equation 11. Indeed, the barycentric mapping defines in equation 22 is an approximation to the Monge map $T_k^\star$ (Deb et al., 2021), as long as it exists (e.g., p_k assumed to be absolutely continuous).

The computation of the CFM targets has complexity $\mathcal{O}(KB^2)$ per fixed-point iterations for both the neural and empirical solvers. The main difference is that, within the fixed-point iterations, the empirical solver needs to solve $K-$OT problems, with a complexity that scales with the number of Sinkhorn iterations. Meanwhile, the neural solver requires a single forward pass, being able to estimate in *one-shot* the OT variables. Therefore, complexity generally reduces from $\mathcal{O}(L_{\text{FP}} L_{\text{Sinkhorn}} KB^2)$ to $\mathcal{O}(L_{\text{FP}} KB^2)$. We present the empirical version of `BaryFM` in Algorithm 2.

Algorithm 2 Empirical Barycentric Flow Matching

Input: Measures $\{p_k\}_{k=1}^K$, velocity field v_θ, batch size B, iterations N, entropic regularization ϵ

for iter $= 1, \dots, N$ **do**
  Sample $\lambda \sim \text{Dir}(\alpha)$ and $\{\{z_{k,j}\}_{j=1}^B\}_{k=1}^K$, where $z_{k,j} \sim p_k$
  $\bar{z}_i^{(0)} \leftarrow \sum_{k=1}^K \lambda_k z_{k,i}$
  for $\ell = 1, \cdots, L_{\text{FP}}$ **do**
    $C_{k,i,j} \leftarrow \|\bar{z}_i^{(\ell-1)} - z_{k,j}\|_2^2$
    $\gamma_{1:K}^{(\ell)} \leftarrow \text{Sinkhorn}\,(a_{1:K}, b, C_{1:K}, \epsilon)$ ▷ $a_{k,i} = B^{-1}, b_j = B^{-1}$
    $\bar{z}_i^{(\ell)} \leftarrow \sum_{k=1}^K \lambda_k \sum_{j=1}^B \frac{\gamma_{k,i,j}^{(\ell)}}{\sum_{j'=1}^B \gamma_{k,i,j'}^{(\ell)}} z_{k,j}$
  end for
  $\mathcal{L}_{\text{C}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1], k\sim\text{Cat}(\lambda), (z_0,z_1)\sim\gamma_k}\left[\|v_\theta(z_t, t, k, \lambda) - (z_1 - z_0)\|_2^2\right]$
  $\theta \leftarrow \theta - \text{LR}_{\text{C}}\, \nabla_\theta \mathcal{L}_{\text{C}}(\theta)$
end for
return θ

C.2 Neural Fixed-Point

We recall that the fixed-point operator of Álvarez-Esteban et al. (2016) is,

$$G(q) = \left(\sum_{k=1}^K \lambda_k T_k^\star\right)_\sharp q,$$

where $T_k^\star$ is the Monge map from q to p_k. Since the barycentric map $\hat{T}_{\epsilon,\psi}$ is an approximation of the Monge map, when it exists (Deb et al., 2021; Pooladian & Niles-Weed, 2021), we devise a variant of our `BaryFM` method that relies exclusively on the dual OT head, by approximating G via,

$$\hat{G}_{\epsilon,\psi}(q) = \left(\sum_{k=1}^K \lambda_k \hat{T}_{\epsilon,\psi}(\cdot, k, \lambda)\right)_\sharp q$$

We summarize its learning algorithm in Algorithm 3.

Since this variant does not have a velocity field, inference is done via the fixed-point strategy in equation 22, with $\hat{T}_k$ replaced by the neural barycentric mapping $\hat{T}_{\epsilon,\psi}$,

$$\begin{aligned} \bar{z}_i^{(\ell+1)} &= \sum_{k=1}^K \lambda_k \hat{T}_{\epsilon,\psi}(\bar{z}_i^{(\ell)}, k, \lambda), \\ \hat{T}_{\epsilon,\psi}(\bar{z}_i^{(\ell)}) &= \sum_{j=1}^B \text{softmax}_j\left(\frac{f_{\epsilon,\psi}(z_{k,j}, k, \lambda) - c(z_{k,j}, \bar{z}_i^{(\ell)})}{\epsilon}\right) z_{k,j}. \end{aligned} \tag{23}$$

At inference, we draw samples $z_{k,i} \sim p_k$, initialize $\bar{z}_i^{(0)} = \sum_{k=1}^K \lambda_k z_{k,i}$ then execute the iterations in equation 23. This procedure is shown in Algorithm 4, and can be interpreted as a neural version of the algorithms of Cuturi & Doucet (2014) and Montesuma et al. (2023).

Related Work. The `NeuralFP` version of our method sits within the existing literature of neural solvers for the fixed-point approach of Álvarez-Esteban et al. (2016), such as Korotin et al. (2022a)

Algorithm 3 `NeuralFP`: Neural Fixed-Point Barycenters. Sampling re-uses lines 4–8 with n samples per marginal and any $\lambda \in \Delta_K$, returning $\{\bar{z}_i^{(L_{\text{FP}})}\}_{i=1}^n$.

Input: Measures $p_1, \cdots, p_K$, dual potential f_ψ, batch size B, iterations N, # of fixed-point iterations L_{FP}, # of dual iterations L_{Dual}, entropic regularization ϵ

1: **for** iter $= 1, \ldots, N$ **do**
2: Sample $\lambda \sim \text{Dir}(\alpha)$ and $\{\{z_{k,j}\}_{j=1}^B\}_{k=1}^K$, where $z_{k,j} \sim p_k$
3: $f_{k,j} \leftarrow f_{\epsilon,\psi}(z_{k,j}, k, \lambda)$ ▷ Single pass: $f_{\epsilon,\psi}$ does not depend on $\bar{z}$
4: Initialize $\bar{z}_i^{(0)} \leftarrow \sum_{k=1}^K \lambda_k z_{k,i}$
5: **for** $\ell = 1, \cdots, L_{\text{FP}}$ **do**
6: $C_{k,i,j} \leftarrow c(z_{k,j}, \bar{z}_i^{(\ell-1)})$
7: $\gamma_{j|i,k} \leftarrow \text{softmax}_c\left({}^{(f_{k,j} - C_{k,i,j})}/_{\epsilon}\right)$ ▷ $\gamma_{\epsilon,\psi}(z_{k,j}|\bar{z}_i^{(\ell-1)}, k, \lambda)$, eq. 15
8: $\bar{z}_i^{(\ell)} \leftarrow \text{sg}\left(\sum_{k=1}^K \lambda_k \sum_{j=1}^B \gamma_{j|i,k} z_{k,j}\right)$ ▷ No backprop through ψ
9: **end for**
10: **for** $s = 1, \ldots, L_{\text{Dual}}$ **do**
11: $\mathcal{L}_{\text{D}} \leftarrow \sum_{k=1}^K \lambda_k \Big(\mathbb{E}_z[f_{\epsilon,\psi}(z,k,\lambda)] + \mathbb{E}_{\bar{z}}[f^{(c)}_{\epsilon,\psi}(\bar{z},k,\lambda)] - \epsilon(\mathbb{E}_{z,\bar{z}}[\gamma_{\epsilon,\psi}(z,\bar{z},k,\lambda)] - 1) \Big)$,
12: $\psi \leftarrow \psi + \text{LR}_{\text{D}} \nabla_\psi \mathcal{L}_{\text{D}}$
13: **end for**
14: **end for**
15: **return** ψ

Algorithm 4 `NeuralFP` Inference

Input: Measures $p_{1:K}$, dual potential f_ψ, number of samples n, # of fixed-point iterations L_{FP}, weights λ, entropic regularization ϵ

Sample $\{\{z_{k,j}\}_{j=1}^n\}_{k=1}^K$, where $z_{k,j} \sim p_k$
$f_{k,j} \leftarrow f_{\epsilon,\psi}(z_{k,j}, k, \lambda)$
$\bar{z}_i^{(0)} \leftarrow \sum_{k=1}^K \lambda_k z_{k,i}$
for $\ell = 1, \cdots, L_{\text{FP}}$ **do**
$C_{k,i,j} \leftarrow c(z_{k,j}, \bar{z}_i^{(\ell-1)})$
$\gamma_{j|i,k} \leftarrow \text{softmax}_c\left({}^{(f_{k,j} - C_{k,i,j})}/_{\epsilon}\right)$
$\bar{z}_i^{(\ell)} \leftarrow \sum_{k=1}^K \lambda_k \sum_{j=1}^B \gamma_{j|i,k} z_{k,j}$
end for
return $\{\bar{z}_i^{(L_{\text{FP}})}\}_{i=1}^n$ ▷ Support of $\hat{q}_\lambda^{(L_{\text{FP}})}$

and Howard et al. (2025). The shared goal is approximating the fixed-points of the operator G. We divide these methods into 2 families.

The first family parametrize q and p_k empirically, meaning, they are mixtures of Diracs. As a consequence, the sample-level Monge map $T_k^\star$ may not exist (e.g., when the supports have different size) but they can be approximated via the barycentric mapping in equation 22. This is the case of works such as Cuturi (2013); Montesuma et al. (2023); von Lindheim (2023) and Montesuma et al. (2026a).

The second family parametrize the iterate $q_\lambda^{(\ell)}$ through a neural network. This is the case of `WIN` (Korotin et al., 2022a). These authors propose a generative model for the barycenter iterate, that is, $q_\lambda^{(\ell)} = \xi_\sharp q_0$, where q_0 is a latent measure. Their method works by alternating 2 steps. First, they estimate $K-$transport maps T_{θ_k} from $\xi_\sharp q_0$ to p_k. Second, they regress ξ into the approximated fixed-point operator $\sum_k \lambda_k T_{\theta_k} \circ \xi$.

In comparison with these approaches, our `NeuralFP` parametrizes the dual OT parameters through a single network f_ψ. While this approach computes an entropic barycenter, it only needs $K-$neural networks and its parametrization is independent of the support of $q_\lambda^\star$. In contrast, the 1st family of methods parametrize the barycentric measure directly through their support, and the 2nd family, namely, `WIN`, uses $(2K+1)$ neural networks.

C.3 SEMI-UNBALANCED BARYFM

Taking inspiration from Gazdieva et al. (2024), we compute a semi-unbalanced version of our BaryFM. In a nutshell, unbalanced OT is an optimization over a coupling with relaxed constraints. Recalling equations 3 and 4, the unbalanced OT problem (Chizat et al., 2018; Liero et al., 2018) is,

$$\text{UOT}(p,q) = \underset{\gamma\in\mathcal{M}_+(\Omega\times\Omega)}{\arg\inf} \mathbb{E}_{(z,z')\sim\gamma}[c(z,z')] - \epsilon H(\gamma) + D_\eta(\pi_{1,\sharp}\gamma\|p) + D_\xi(\pi_{2,\sharp}\gamma\|q),$$

where D_η and D_ξ are Csiszár (1975) divergences between measures over Ω (e.g., the Kullback-Leibler divergence), η and ξ are convex functions with $\eta(1) = \xi(1) = 0$, and $\pi_i(z_1, \cdots, z_n) = z_i$ is the projection onto the $i-$th coordinate. This problem takes the hard-constraint OT problem, and regularizes it so γ is simply a non-negative measure, while D_η and D_ξ act as a soft-constraint type of regularization. We are interested in the *semi-unbalanced case* (Gazdieva et al., 2024, Equation 4), in which only one marginal is penalized,

$$\text{SUOT}(p,q) = \underset{\gamma\in\Gamma(q)}{\arg\inf}\, \mathbb{E}_{(z,z')\sim\gamma}[c(z,z')] - \epsilon H(\gamma) + \tau D_\eta(\pi_{1,\sharp}\gamma\|p), \tag{24}$$

where $\Gamma(q) = \{\gamma \in \mathcal{M}_+(\Omega\times\Omega) : \pi_{2,\sharp}\gamma = q\}$ is the set of plans with the 2nd marginal constraint, and $\tau > 0$ is the 1st marginal violation penalty weight. This formulation is adequate for barycenters, as in practice we want to allow the mass constraint over $p_{1:K}$ to be violated. The mass constraint on the barycenter side, $q_\lambda^\star$, must be respected, otherwise we could have degenerate solutions (e.g., barycenters with null mass). In light of equation 24, we consider a modified semi-unbalanced dual loss (Gazdieva et al., 2024, Equation 5),

$$\sum_{k=1}^{K}\lambda_k\Big(\mathbb{E}_z[-\tau\eta^*(-f_{\epsilon,\psi}(z,k,\lambda)/\tau)] + \mathbb{E}_{\bar z}[f^{(c)}_{\epsilon,\psi}(\bar z,k,\lambda)] - \epsilon(\mathbb{E}_{z,\bar z}[\gamma_{\epsilon,\psi}(z,\bar z,k,\lambda)] - 1)\Big). \tag{25}$$

where $\eta^*(u)$ denotes the Fenchel conjugate $\eta^*(u) = \sup_{s\geq 0}\{su - \eta(s)\}$ evaluated at the dual $f_{\epsilon,\psi}(z,k,\lambda) \in \mathbb{R}$.

C.4 OTHER ODE INTEGRATION STRATEGIES

In the main paper (c.f. equation 19), we described the forward Euler strategy for solving the ODE,

$$\partial_t z(t) = v_\theta(z(t), t, k, \lambda),$$

by approximating $\partial_t z(t) = {}^{dz}/_{dt} \approx {}^{(z_{r+1}-z_r)}/_{dt}$, $r = {}^{t}/_{dt}$. Other numerical approximation strategies exist (Burden & Faires, 2010, Chapter 5). In our experiments (c.f. Section D), we ablate the Euler strategy against the midpoint strategy,

$$z_{\text{mid},i} = z_{r,i} + v_\theta(z_{r,i}, rdt, k, \lambda)\frac{dt}{2},$$
$$z_{r+1,i} = z_{r,i} + v_\theta(z_{\text{mid},i}, rdt + {}^{dt}/_2, k, \lambda)dt$$

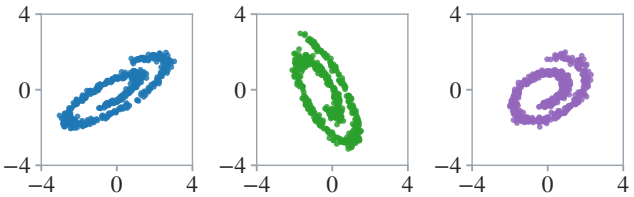
It is a general fact from numerical analysis that the error involved with discretizing the ODE depends on the strategy (e.g., Euler, or midpoint), and that midpoint the solver generally has a smaller error for the same time discretization parameter dt. See Figure 12 for a comparison in our context.

D Swiss Roll Measures

In this experiment, we build a toy example where each member of the family $\mathcal{W}(p_{1:K})$ has a closed-form ground-truth. To that end, we use the theory developed in (Álvarez-Esteban et al., 2016, Section 4), where the authors extend the fixed-point barycenter computation algorithm to location-scatter families. Briefly, let p_0 be a reference measure, such as a Gaussian measure, or, in the context of this experiment, the Swiss roll measure. We assume, without loss of generality, that p_0 is centered with identity covariance. Given $T_1, \cdots, T_K$ affine transformations, we consider the family,

$$\text{Location-Scatter}(p_0) = \{T_{k,\sharp}p_0 : T_k(x) = A_k x + b_k, A_k \in \text{SPD}(d), \text{ and } b_k \in \mathbb{R}^d\}_{k=1}^K. \tag{26}$$

Each measure $p_k \in$ Location-Scatter(p_0) is the image measure of p_0 by the affine transformation T_k. We show an example of the marginal measures in Figure 9, which are used throughout the examples in this paper. We use $K = 3$ for visualizing these measures in the simplex



(a) p_1 (b) p_2 (c) p_3

Figure 9: The $K = 3$ Swiss-roll marginals.

This example is similar to that proposed by Korotin et al. (2021), and is interesting for several reasons. First, the Swiss roll measure itself has a non-linear structure that makes the question of barycenter accuracy interesting. Second, its barycenters have analytical solution in terms of the parameters $\{(A_k, b_k)\}_{k=1}^K$, so methods can be scored against a known ground-truth. Third, and more importantly to us, the family $\mathcal{W}(p_{1:K})$ is also known analytically. For reference, following (Álvarez-Esteban et al., 2016, Section 4), barycenters of $p_{1:K}$ are $q_\lambda^\star = T_{\lambda,\sharp}p_0$, where $T_\lambda(x) = A_\lambda x + b_\lambda$ is affine with coefficients that depend on $\lambda \in \Delta_K$.

In this experiment, we use $K = 3$ measures with the following coefficients,

$$A_1 = \begin{bmatrix} 1.2292 & 0.7474 \\ 0.7474 & 1.4325 \end{bmatrix} \quad b_1 = [-0.3807 \quad -0.2799],$$

$$A_2 = \begin{bmatrix} 1.7729 & 0.6365 \\ 0.6365 & 0.9066 \end{bmatrix} \quad b_2 = [-0.5273 \quad -0.2228],$$

$$A_3 = \begin{bmatrix} 0.5725 & 0.1488 \\ 0.1488 & 2.1387 \end{bmatrix} \quad b_3 = [-0.1675 \quad -0.1474],$$

As mentioned earlier, the main interest of this family is that barycenters have analytical solution in terms of T_λ and p_0,

$$b_\lambda = \sum_{k=1}^K \lambda_k b_k, \;\; A_\lambda^2 = \sum_{k=1}^K \lambda_k \left(A_\lambda A_k^2 A_\lambda\right)^{1/2}. \tag{27}$$

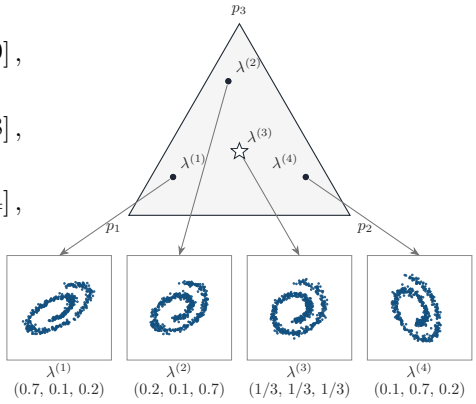


Figure 10: Wasserstein simplex $\mathcal{W}_{p_{1:3}}$ with 4 measures highlighted. Due the location-scatter structure, all measures in $\mathcal{W}(p_{1:3})$ are known in closed-form.

We compare 3 methods over this family of measures: `WGF` (Montesuma et al., 2026a), which computes, from scratch, $q_\lambda^\star$. `NWBU` (Fan et al., 2020, Appendix B), which uses partial ICNNs, accepting λ as input, and `BaryFM` (ours). We rank methods according the BW2-UVP% metric of Korotin et al. (2021), which itself depends on the Wasserstein distance over Gaussian measures, known as the Bures-Wasserstein metric (Takatsu, 2011),

$$\text{BW}_2(p,q)^2 = \|\mu_p - \mu_q\|_2^2 + \text{Tr}\Big(\Sigma_p + \Sigma_q - 2\big(\Sigma_p^{1/2}\Sigma_q\Sigma_p^{1/2}\big)^{1/2}\Big). \tag{28}$$

From the BW metric,

$$\text{BW2-UVP\%}(p|q) = 100\frac{\text{BW2}(p,q)^2}{\frac{1}{2}\text{Var}(q)}\%, \tag{29}$$

where $\text{Var}(q) = \int_\Omega \|z - \mu_q\|^2 dq$ is the variance of the probability measure q.

Main Results. We summarize our main results in Figure 11, which shows the universal approximation of the Swiss roll measures across the simplex. We highlight the barycenter under uniform weights, at the center of the simplex. Overall, `WGF` and `BaryFM` barycenters are more accurate than `NWBU`, by the intensity of colors of the heatmaps across the simplex. Qualitatively, the 2nd row of Figure 11 shows that `NWBU` barycenters are correct, but considerably more noisy. Curiously, `WGF`'s barycenters have a very tight support, which is a consequence of entropic regularization. In contrast, our `BaryFM` fits the underlying Swiss roll manifold very well.

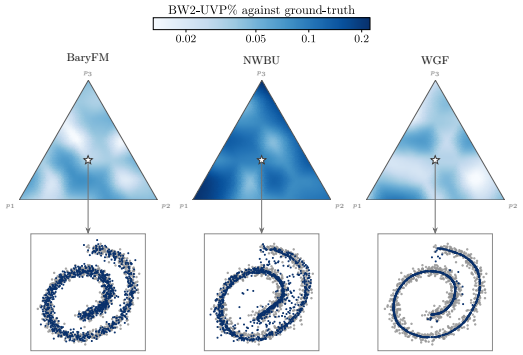


Figure 11: BW2-UVP% over the simplex for `WGF`, `NWBU`, and `BaryFM` (ours) in the Swiss Roll toy example. Blue points represent the calculated barycenter by each method, gray points represent points from the ground-truth.

Sampling Speed. An immediate advantage of amortizing the sampling of the family $\mathcal{W}(p_{1:K})$ is computational efficiency. To that end we explore, in Figure 12, the average accuracy across the simplex as a function of # of velocity-field evaluations, and seconds per barycenter. Our method sits between `NWBU` and `WGF`, while being around 4 order of magnitude faster than the latter, while at parity with respect BW2-UVP%.

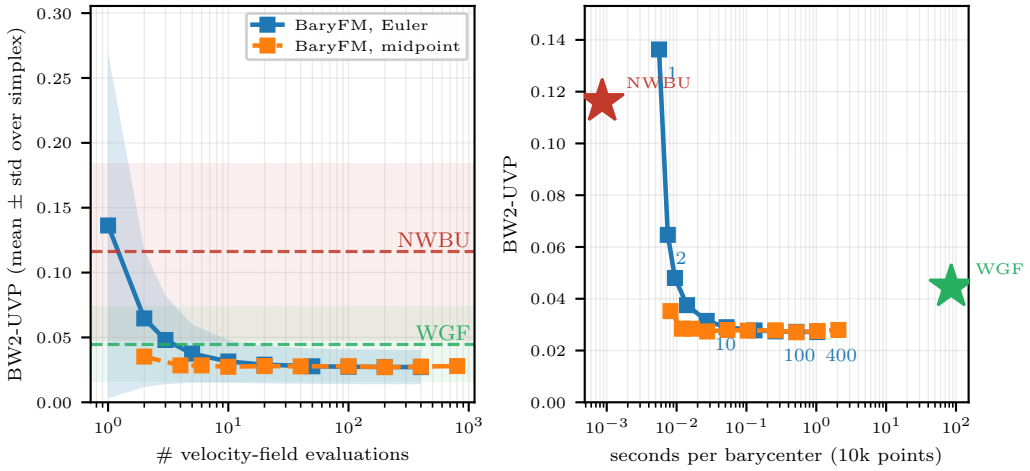


Figure 12: BW2-UVP% as a function of # of velocity field evaluations and inference time.

E BAYESIAN POSTERIOR AGGREGATION

Background. Let $\mathcal{D}$ denote a dataset (e.g., a collection of samples $\{(x_i, y_i)\}_{i=1}^n$). Bayesian inference is concerned with estimating the *posterior measure*, $p(\theta|\mathcal{D})$, where $\theta \in \Theta$ are model parameters (e.g., weights of neural network, or coefficients of a linear regression model) given the dataset $\mathcal{D}$. The naming *posterior*, comes from Bayes rule,

$$p(\theta|\mathcal{D}) = \frac{p(\theta)p(\mathcal{D}|\theta)}{p(\mathcal{D})},$$

where $p(\theta)$ is called the prior, $p(\mathcal{D}|\theta)$ is the likelihood, and $p(\mathcal{D})$ is the probability of observing the dataset $\mathcal{D}$. Therefore, given a set of datasets $\{\mathcal{D}_k\}_{k=1}^K$, Bayesian posterior aggregation seeks to combine the posterior measures $\{p(\theta|\mathcal{D}_k)\}_{k=1}^K$. Samples from these posteriors, $\{\{\theta_{k,j}\}_{j=1}^{n_k}\}_{k=1}^K$ can be obtained, for instance, via Markov Chain Monte Carlo (MCMC).

In this context, Srivastava et al. (2015; 2018) were the first to aggregate posterior via Wasserstein barycenters, that is, via,

$$q^\star = \text{Bar}(\bar{\lambda}, p_{1:K}) = \underset{q \in \mathcal{P}_2(\Omega)}{\arg\min} \frac{1}{K} \sum_{k=1}^K W_2(q, p(\theta|\mathcal{D}_k))^2.$$

In other words, given $p_k = p(\theta|\mathcal{D}_k)$, they propose $q_{\bar{\lambda}}^\star = \text{Bar}(\bar{\lambda}, p_{1:K})$, with $\bar{\lambda} = (1/K, \cdots, 1/K)$. This task has a well defined ground-truth, as the true posterior corresponds to sampling from $q_0 = p(\theta|\mathcal{D})$, for instance, running MCMC on the full dataset. Therefore, methods are ranked with respect their BW2-UVP% against the ground-truth, BW2-UVP%$(\hat{q}_{\bar{\lambda}}, q_0)$. Note, however, that the BW2-UVP compares measures based on their 1st and 2nd moments only. We therefore include in our comparison the $2-$Sliced-Wasserstein distance,

$$\text{SW}_2(p, q)^2 = \int_{\mathbb{S}^{d-1}} W_2(\pi_{u,\sharp}p, \pi_{u,\sharp}q)^2 du \approx \frac{1}{N_{\text{proj}}} \sum_{i=1}^{N_{\text{proj}}} W_2(\pi_{u_i,\sharp}p, \pi_{u_i,\sharp}q)^2, \tag{30}$$

where $\mathbb{S}^{d-1}$ is the unit sphere in $\mathbb{R}^d$, and $u_i \sim \mathcal{U}(\mathbb{S}^{d-1})$ is an unit norm vector. We use $N_{\text{proj}} = 500$ projections. The Sliced Wasserstein distance relies on the Wasserstein distance between 1-dimensional measures, and enjoys a dimension independent statistical complexity. We refer readers to Montesuma et al. (2024a) and Peyré & Cuturi (2019) for further discussion.

Dataset. For this experiment we consider the dataset introduced in Noble et al. (2023). These authors have obtained samples from a logistic regression model on the UCI[1] `Wine` dataset of Aeberhard et al. (1994). More specifically, the authors have used 3 partitions of the original `Wine` dataset for obtaining 9,900 samples from each $p_k = p(\theta|\mathcal{D}_k)$, $k = 1, \cdots, 3$. Likewise, the 9,900 samples are obtained from the true posterior $q_0 = p(\theta|\mathcal{D})$. We refer readers to (Noble et al., 2023, Appendix G.2) for further details on the specifics of their numerical simulations. All methods are evaluated with respect the BW2-UVP% and SW_2 metrics, using all the 29,700 transported marginal samples to the computed barycenter.

Main Results. Our main results are shown in Table 5. We compare our `BaryFM` against several barycenter solvers. Every method is run on 3 seeds (42, 43 and 44), and we report their $\mu_{\pm\sigma}$. Overall, our semi-unbalanced variant (`BaryFM-U`) performs better than the its balanced version (`BaryFM`). For instance, in the homogeneous benchmark, `BaryFM-U` ranks 5th, while `BaryFM` ranks 7th. We are able to improve performance with Classifier Free Guidance (CFG) (see analysis below), matching the best previous method on the homogeneous benchmark, and surpassing it on the hterogeneous one.

In the homogeneous benchmark, with $k-$CFG, our `BaryFM-U` achieves $8.02_{\pm 0.07}$ against `U-NOT` $8.08_{\pm 0.10}$ (2nd best, also unbalanced method), thus being statistically tied. In the heterogeneous benchmark, we our `BaryFM-U` achieves $9.93_{\pm 0.17}$, against $10.26_{\pm 0.08}$ of `U-NOT` (3rd best, also unbalanced method). Overall, the main drivers of performance in this benchmark are (i) universal training, (ii) CFG, and (iii) semi-unbalanced OT. Universal training is our particular proposal, CFG is made possible by CFM, and semi-unbalanced OT is made possible via the dual OT head.

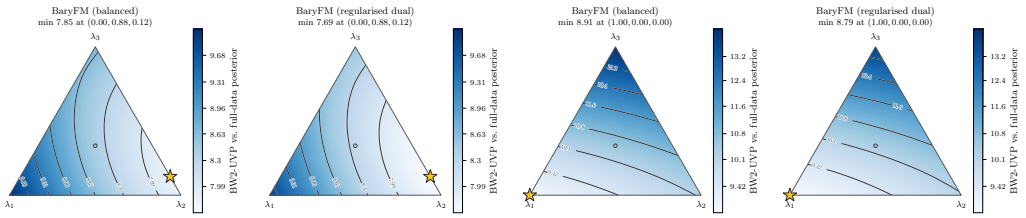
[1] `https://archive.ics.uci.edu/dataset/109/wine`

Method	Universal	Homogeneous				Heterogeneous			
		BW2-UVP%	Rank	SW_2	Rank	BW2-UVP%	Rank	SW_2	Rank
Discrete	✗	$17.70_{\pm 0.02}$	16	$0.310_{\pm 0.004}$	17	$18.92_{\pm 0.08}$	17	$0.317_{\pm 0.007}$	17
WGF	✗	$20.42_{\pm 0.02}$	18	$0.333_{\pm 0.004}$	18	$20.93_{\pm 0.00}$	18	$0.338_{\pm 0.004}$	18
TDSB	✗	$14.22_{\pm 1.73}$	14	$0.234_{\pm 0.019}$	15	$17.03_{\pm 1.46}$	15	$0.256_{\pm 0.011}$	14
TSBM	✗	$8.09_{\pm 0.03}$	3	$0.179_{\pm 0.007}$	8	$10.34_{\pm 0.07}$	5	$0.202_{\pm 0.007}$	9
CW2B	✗	$11.58_{\pm 0.69}$	12	$0.222_{\pm 0.012}$	14	$16.27_{\pm 1.54}$	14	$0.264_{\pm 0.011}$	15
WIN	✗	$12.79_{\pm 0.72}$	13	$0.207_{\pm 0.010}$	12	$15.23_{\pm 1.67}$	12	$0.227_{\pm 0.022}$	13
NOT	✗	$8.54_{\pm 0.18}$	9	$0.183_{\pm 0.008}$	10	$11.10_{\pm 0.20}$	11	$0.209_{\pm 0.009}$	11
U-NOT	✗	$8.08_{\pm 0.10}$	2	$0.176_{\pm 0.007}$	5	$10.26_{\pm 0.08}$	3	$0.197_{\pm 0.008}$	6
NF	✗	$8.41_{\pm 0.02}$	7	$0.183_{\pm 0.007}$	10	$10.79_{\pm 0.03}$	9	$0.206_{\pm 0.007}$	10
NWB	✗	$15.65_{\pm 4.10}$	15	$0.217_{\pm 0.045}$	13	$15.34_{\pm 1.90}$	13	$0.219_{\pm 0.032}$	12
GMM	✗	$8.28_{\pm 0.15}$	6	$0.178_{\pm 0.008}$	7	$10.69_{\pm 0.15}$	8	$0.201_{\pm 0.009}$	8
`BaryFM` (ours)	✗	$9.31_{\pm 0.17}$	11	$0.182_{\pm 0.007}$	9	$10.80_{\pm 0.08}$	10	$0.199_{\pm 0.008}$	7
`BaryFM-U` (ours)	✗	$8.97_{\pm 0.10}$	10	$0.175_{\pm 0.007}$	4	$10.25_{\pm 0.17}$	2	$0.191_{\pm 0.007}$	2
NWBU	✓	$20.30_{\pm 2.36}$	17	$0.265_{\pm 0.028}$	16	$18.40_{\pm 1.28}$	16	$0.268_{\pm 0.019}$	16
`BaryFM` (ours)	✓	$8.41_{\pm 0.06}$	7	$0.176_{\pm 0.006}$	5	$10.51_{\pm 0.23}$	7	$0.195_{\pm 0.009}$	5
`BaryFM-U` (ours)	✓	$8.27_{\pm 0.06}$	5	$0.169_{\pm 0.005}$	1	$10.35_{\pm 0.25}$	6	$0.192_{\pm 0.006}$	3
`BaryFM` (ours) + k-CFG	✓	$8.19_{\pm 0.07}$	4	$0.174_{\pm 0.009}$	3	$10.31_{\pm 0.46}$	4	$0.194_{\pm 0.012}$	4
`BaryFM-U` (ours) + k-CFG	✓	$8.02_{\pm 0.07}$	1	$0.170_{\pm 0.008}$	2	$9.93_{\pm 0.17}$	1	$0.189_{\pm 0.009}$	1

Table 5: Bayesian Posterior Aggregation results. We report BW2-UVP% and SW_2 over seeds 42, 43 and 44 for each method.

On the benefits of universality. One result that stands out from Table 5 is the effect of universality. Comparing `BaryFM` universal versus non-universal, its ranks improve from 11th in the Homogeneous setting to 7th, and from 10th to 7th in the Heterogeneous setting. In the semi-unbalanced version, the gains are more mixed. In the Homogeneous benchmark rank improves from 10th to 5th, and in the Heterogeneous benchmark rank degrades from 2nd to 6th. Except for this degradation, universality tends to improve the downstream score in posterior aggregation (i.e., similarity with the posterior).

Besides the ranking improvement, there is a second advantage of approximating the Wasserstein simplex $\mathcal{W}(p_{1:K})$, as we can actually explore $\lambda \mapsto \text{BW2-UVP\%}(\hat{q}_\lambda^{(L_{\text{FP}})}, q_0)$. We show this exploration in Figure 13, which shows that the minimum BW2-UVP% is not actually in the uniform barycenter $\bar{\lambda}$. Indeed, our algorithm is able to obtain much better metrics in the homogeneous (best BW2-UVP% 7.69) and heterogeneous settings (best BW2-UVP% 8.79).



(a) Homogeneous. (b) Homogeneous. (c) Heterogeneous. (d) Heterogeneous.

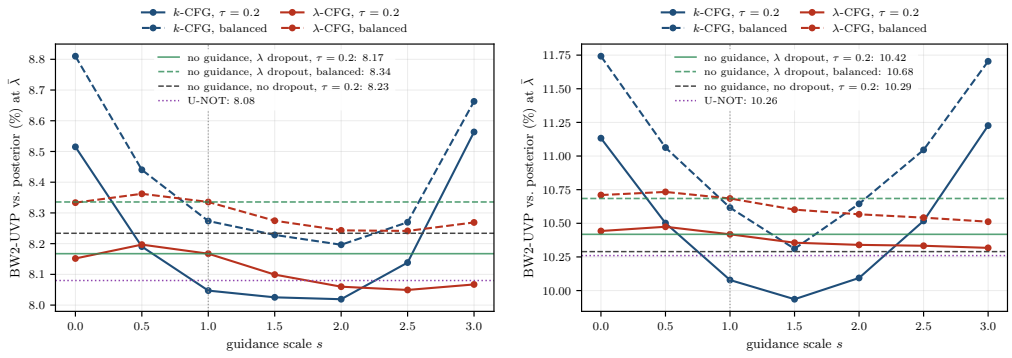
Figure 13: BW2-UVP% between $\hat{q}_\lambda^{(L_{\text{FP}})}$, the λ-weighted Wasserstein barycenter of marginals, against the ground truth full posterior q_0 over the simplex Δ_3.

Classifier Free Guidance. A nice possibility made available by flow matching is CFG (Ho & Salimans, 2022), based on dropping out the conditioning factors of $v_\theta(z, t, k, \lambda)$. We experiment with the dropout of 2 conditioning variables, the domain index k, and the barycentric coordinates $\lambda \in \Delta$,

$$v(z, t, k, \lambda, s) = v_\theta(z, t, \emptyset, \lambda) + s(v_\theta(z, t, k, \lambda) - v_\theta(z, t, \emptyset, \lambda)), \tag{31}$$

$$v(z, t, k, \lambda, s) = v_\theta(z, t, k, \emptyset) + s(v_\theta(z, t, k, \lambda) - v_\theta(z, t, k, \emptyset)), \tag{32}$$

where s is the so-called guidance scale. As commonly done in the flow matching and diffusion literature (Ho & Salimans, 2022), we define the embedding vectors $E_k(\emptyset)$ and $E_\lambda(\emptyset)$, and train our `BaryFM` with a dropout probability on the conditionings k and λ.



(a) Homogeneous (b) Heterogeneous

Figure 14: Different CFG strategies (k-CFG and λ-CFG) for Bayesian Posterior Aggregation. Through CFG ($s \neq 1$, especially $s \in (1, 2]$ our proposed `BaryFM` improves over the best previous baseline, `U-NOT` (Gazdieva et al., 2024).

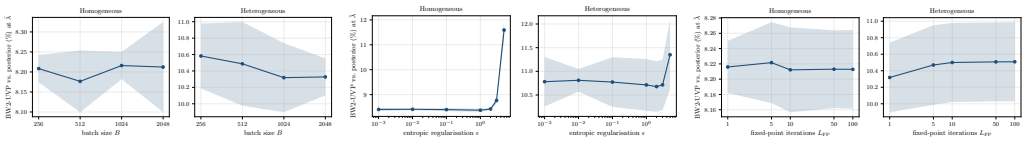
We call equation 31 $k-$conditioning, and equation 32 $\lambda-$conditioning. From Figure 14, CFG on the $k-$conditioning has a clear $U-$shape curve. Performance improves with a reasonable $s > 1.0$, thus with CFG active, achieving its best at $s = 2.0$ (homogeneous, both variants) and $s = 1.5$ (heterogeneous, both variants). Performance improvements from $\lambda-$dropout are somewhat more mixed. On the one hand, it is able to improve to similar levels of $k-$CFG in the Homogeneous benchmark. In the heterogeneous, gains are almost flat as a function of s. Interestingly, this CFG strategy is monotonic in s.

	`BaryFM`		`BaryFM-U`	
	–	$+k$-CFG	–	$+k$-CFG
Semi-unbalanced penalty τ	–	–	0.2	0.2
Batch size B	1 024	4 096	1 024	4 096
$k-$dropout (Homogeneous/Heterogeneous)	0.0	0.8 / 0.1	0.0	0.7 / 0.2
Guidance scale s (Homogeneous/Heterogeneous)	1.0	2.0 / 1.5	1.0	2.0 / 1.5
Entropic regularisation ϵ	2			
Fixed-point iterations L_{FP}	1			
Dual iterations L_{Dual}	1			
Divergence η	KL			
Iterations N	3 000			
Optimizer (field and dual)	Adam, LR $= 10^{-5}$			
Learning-rate schedule	cosine, $\text{LR}_{\text{min}} = 10^{-7}$			
EMA decay	0.999			
Gradient clipping	1.0			
Network	densely connected, 2 layers, width 512			
Embeddings (feature / time / domain / weight)	128 / 32 / 32 / 32			
n_{gen}	29 700 (full support, $3 \times 9\,900$)			
ODE solver	Midpoint, 100 steps			

Table 6: Hyperparameters of the universal `BaryFM` rows in Table 5.

Hyperparameters and their sensitivity. We present, in Table 6, the hyperparameters used in our method, and its sensitivity to hyperparameter choice. We perform this study on the following axis: entropic regularization $\epsilon \in \{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 2 \times 10^{0}, 3 \times 10^{0}, 5 \times 10^{0}\}$, $B \in \{256, 512, 1024, 2048\}$, $L_{\text{FP}} \in \{1, 5, 10, 50, 100\}$. Our results are summarized in Figure 15. Overall, our algorithm is robust to both B and L_{FP}, having a smaller variation than seed standard

deviation. Sensitivity to ϵ is more pronounced, and our algorithm tends to prefer smaller ϵ (i.e., $\epsilon \leq 2$).



(a) Sensitivity to B. (b) Sensitivity to ϵ. (c) Sensitivity to L_{FP}.

Figure 15: Sensitivity to batch size B, entropic regularization ϵ and # of fixed point iterations L_{FP}.

Conclusion. Overall, our results demonstrate 2 interesting effects. First, the best posterior approximation is not attained at the center of simplex, meaning $\hat{q}_{\bar{\lambda}}$. Indeed, exploring the whole Wasserstein simplex $\mathcal{W}(p_{1:K})$ allows us to improve the BW2-UVP% against the true posterior from 10.51 to 8.91 (heterogeneous, balanced `BaryFM`), with similar gains of our method in other settings. As a result, we argue that, at least in the context of this benchmark, the quality of barycenter approximation is only one axis of exploration, as there are other measures within the family of barycenters (i.e., the Wasserstein simplex) that do this task better. Second, CFG is an important mechanism for posterior approximation, although is bias the calculated barycenter.

F FAIRNESS

Background. In the context of machine learning, fairness (Mehrabi et al., 2021) studies the biases of developed models or algorithms against sub-populations in the data. Mathematically, this task is formalized by a protected variable, S, taking values among a discrete set $\{1, \cdots, K\}$. For instance, a protected variable can concern gender, in which case $S \in \{1, 2\}$, where 1 corresponds to *male*, and 2 corresponds to *female*. Let $h : \mathcal{X} \to \mathcal{Y}$ denote a predictor, mapping a feature space $\mathcal{X}$ to a label space $\mathcal{Y}$. Among the different notions of fairness, we consider Disparate Impact (DI),

$$\text{DI}(h, X, S) = \frac{1}{K-1} \sum_{k=2}^{K} \frac{\text{Pr}(h(X) = 1 | S = k)}{\text{Pr}(h(X) = 1 | S = 1)}, \tag{33}$$

where Pr denotes the probability of the event in its argument. $h(X) = 1$ denotes the *positive outcome* (e.g., obtaining a loan or a scholarship), and $S = 1$ is the privileged group (e.g., *male*). In a nutshell, DI measures the ratio, in probability, of getting a positive outcome from the rule h. Intuitively, a rule h is *fairer* as DI $\to 1$.

Wasserstein barycenters were first proposed to achieve fairness in machine learning by Gordaliza et al. (2019). These authors proposed to *repair* the measures of each sub-population into the Wasserstein barycenter weighted by the population proportions. In these experiments, we repair into the uniform barycenter,

$$q^\star_{\bar{\lambda}} = \text{Bar}(\bar{\lambda}, p_{1:K}), \text{ where } \bar{\lambda} = (1/K, \cdots, 1/K).$$

The underlying idea is that the barycenter will correct the potential shifts within the protected variable-conditioned measures p_k. As a result, trained on the repaired measure (called total repair by Gordaliza et al. (2019)), the model will become insensitive to the protected variable. Uniform weights are used to ensure all groups are treated equally.

Experimental Protocol. All barycenter algorithms are compared along 2 axis. First, we measure how fair models trained under the estimated barycenter $\hat{q}_{\bar{\lambda}}$. We measure fairness via equation 33. Second, we measure how performance is degraded by repairing the probability measures. This is a consequence of repairing, and downstream classification accuracy is usually at a trade-off with respect fairness (Zhao & Gordon, 2022). In that regard a we fit a logistic regression model to the repaired data obtained via each barycenter solver.

Datasets. We consider 4 datasets fore evaluating fairness: Adult Income (Kohavi et al., 1996), COMPAS (Angwin et al., 2022), Credit (Yeh & Lien, 2009) and the National Longitudinal Survey of Youth (NLSY) (Bureau of Labor Statistics, U.S. Department of Labor, 2019). We summarize the information about each dataset in Table 7, among which the positive outcome $h(X) = 1$, the considered protected variables, groups and the favored group.

Dataset	Task (Y=1)	# Samples	# Features	Protected Variable	Groups	# Samples per group	%	Pr(Y=1 | S=s)
Adult	income > 50K	45,222	5	gender	**Male**	30,527	67.5	0.312
					Female	14,695	32.5	0.114
COMPAS	two-year **non-recidivism**	5,278	8	race	Afr.-Am.	3,175	60.2	0.477
					Caucasian	2,103	39.8	0.609
Credit	**no** payment default	30,000	22	gender	**Female**	18,112	60.4	0.792
					Male	11,888	39.6	0.758
NLSY	income ≥ median	209,863	5	gender × race	Female-Non-Bl.	78,554	37.4	0.452
					Male-Non-Bl.	76,561	36.5	0.576
					Male-Black	26,979	12.9	0.503
					Female-Black	27,769	13.2	0.426

Table 7: Fairness datasets. On each dataset, we highlight the task according to the positive outcome $Y = 1$. We further list the demographic groups considered on each dataset, and denote in bold the **favored group**.

Main Results. We summarize our results in Table 8, which reports the classification accuracy (Acc.) and DI per dataset and barycenter solver, from which we draw a few observations. First, fairness is improved across the board for most solvers. For instance, the logistic regression fit on `Adult` has DI $= 0.509$, whereas a fit on repaired data through `BaryFM` gets DI $= 1.013$ at a cost

of 1.16% in classification accuracy. Second, the fairness axis makes a clear split between empirical (DWB, WGF) and neural solvers (e.g., NormFlow). Indeed, empirical solvers tend to compensate on the fairness axis while leaving classification accuracy behind, whereas neural solvers tend to preserve accuracy at the cost of a residual fairness gap. One notable example is NWBU, that preserves classification performance on Adult while degrading DI.

Method	Adult		COMPAS		Credit		NLSY	
	Acc	DI	Acc	DI	Acc	DI	Acc	DI
Baseline	$80.84_{\pm 0.19}$	$0.509_{\pm 0.014}$	$69.19_{\pm 2.25}$	$0.662_{\pm 0.007}$	$80.77_{\pm 0.28}$	$0.974_{\pm 0.010}$	$83.69_{\pm 0.12}$	$0.840_{\pm 0.003}$
DWB	$78.82_{\pm 0.33}$	$1.019_{\pm 0.013}$	$67.02_{\pm 0.33}$	$1.013_{\pm 0.008}$	$80.02_{\pm 0.03}$	$0.998_{\pm 0.009}$	$81.27_{\pm 0.13}$	$0.960_{\pm 0.006}$
WGF	$78.67_{\pm 0.32}$	$1.030_{\pm 0.029}$	$65.94_{\pm 1.26}$	$0.981_{\pm 0.022}$	$77.03_{\pm 0.70}$	$0.990_{\pm 0.004}$	$81.00_{\pm 0.10}$	$0.998_{\pm 0.004}$
NormFlow	$80.54_{\pm 0.19}$	$0.652_{\pm 0.071}$	$67.42_{\pm 2.32}$	$0.871_{\pm 0.036}$	$80.72_{\pm 0.31}$	$0.984_{\pm 0.014}$	$83.05_{\pm 0.26}$	$0.942_{\pm 0.025}$
NOT	$79.52_{\pm 0.33}$	$1.074_{\pm 0.180}$	$68.09_{\pm 0.33}$	$0.916_{\pm 0.083}$	$80.78_{\pm 0.30}$	$1.002_{\pm 0.014}$	$82.34_{\pm 0.22}$	$1.014_{\pm 0.011}$
NWB	$78.68_{\pm 0.70}$	$0.499_{\pm 0.037}$	$61.65_{\pm 3.53}$	$1.505_{\pm 1.041}$	$79.36_{\pm 0.36}$	$0.944_{\pm 0.051}$	$74.60_{\pm 3.56}$	$0.944_{\pm 0.164}$
NWBU	$80.87_{\pm 0.81}$	$0.357_{\pm 0.222}$	$68.40_{\pm 1.26}$	$0.692_{\pm 0.047}$	$80.52_{\pm 0.52}$	$0.966_{\pm 0.042}$	$81.91_{\pm 1.38}$	$0.754_{\pm 0.138}$
TDSB	$78.82_{\pm 0.18}$	$0.830_{\pm 0.121}$	$66.16_{\pm 2.52}$	$0.759_{\pm 0.060}$	$80.22_{\pm 0.13}$	$0.983_{\pm 0.038}$	$79.88_{\pm 0.20}$	$0.877_{\pm 0.078}$
TSBM	$79.67_{\pm 0.09}$	$0.898_{\pm 0.016}$	$66.51_{\pm 2.03}$	$0.920_{\pm 0.006}$	$80.52_{\pm 0.22}$	$0.993_{\pm 0.015}$	$80.52_{\pm 0.04}$	$0.984_{\pm 0.009}$
UNOT	$80.23_{\pm 0.07}$	$0.780_{\pm 0.029}$	$68.05_{\pm 0.98}$	$0.853_{\pm 0.029}$	$80.81_{\pm 0.38}$	$0.987_{\pm 0.013}$	$83.00_{\pm 0.18}$	$0.947_{\pm 0.017}$
CW2B	$79.50_{\pm 0.08}$	$1.142_{\pm 0.035}$	$67.17_{\pm 1.12}$	$0.967_{\pm 0.036}$	$80.72_{\pm 0.34}$	$0.997_{\pm 0.010}$	$83.19_{\pm 0.10}$	$0.936_{\pm 0.012}$
WIN	$78.86_{\pm 1.68}$	$0.859_{\pm 0.484}$	$68.18_{\pm 1.16}$	$0.862_{\pm 0.079}$	$80.70_{\pm 0.36}$	$0.975_{\pm 0.005}$	$81.68_{\pm 0.13}$	$1.001_{\pm 0.010}$
GMM	$80.67_{\pm 0.10}$	$0.599_{\pm 0.023}$	$67.33_{\pm 1.55}$	$1.047_{\pm 0.073}$	$80.77_{\pm 0.28}$	$1.108_{\pm 0.175}$	$83.56_{\pm 0.11}$	$0.881_{\pm 0.004}$
BaryFM	$79.68_{\pm 0.76}$	$1.013_{\pm 0.168}$	$67.55_{\pm 2.08}$	$1.023_{\pm 0.013}$	$80.75_{\pm 0.18}$	$0.992_{\pm 0.012}$	$82.39_{\pm 0.16}$	$1.001_{\pm 0.003}$

Table 8: Performance of total repair, per barycenter solver, across 4 fairness datasets.

Fairness vs. Accuracy Tradeoff. Figure 16 summarizes the results of Table 8 by computing the ranking of each metric obtained by of each method on each dataset, then averaging out over datasets. This averaged ranking gives a global view of how methods perform over the 4 tested datasets. Intuitively, methods are better by ranking smaller on both accuracy and fairness ranks, that is, being on the upper right corner of Figure 16 (both axes are **decreasing**). This figure provides another view of our previous claim. Especially, empirical solvers such as DWB and WGF occupy the upper left corner of the figure (better fairness, worse downstream accuracy), while neural methods are more scattered over the plane, with a few (NormFlow, UNOT and NWBU) occupying the bottom right corner (better downstream accuracy, worse fairness). Our proposed BaryFM occupies the upper right corner, having strong fairness metric and downstream accuracy performance.

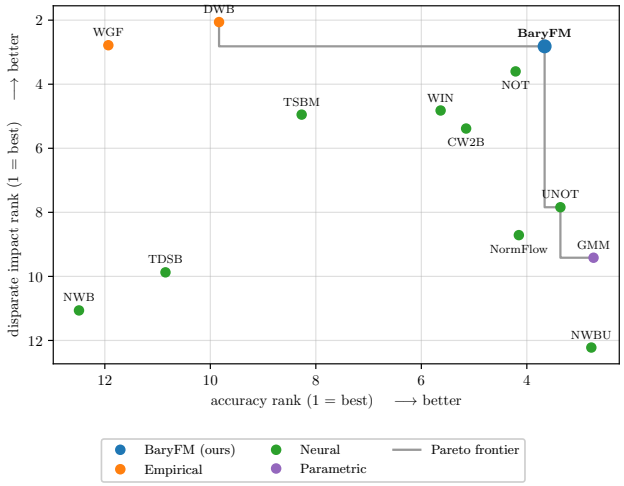
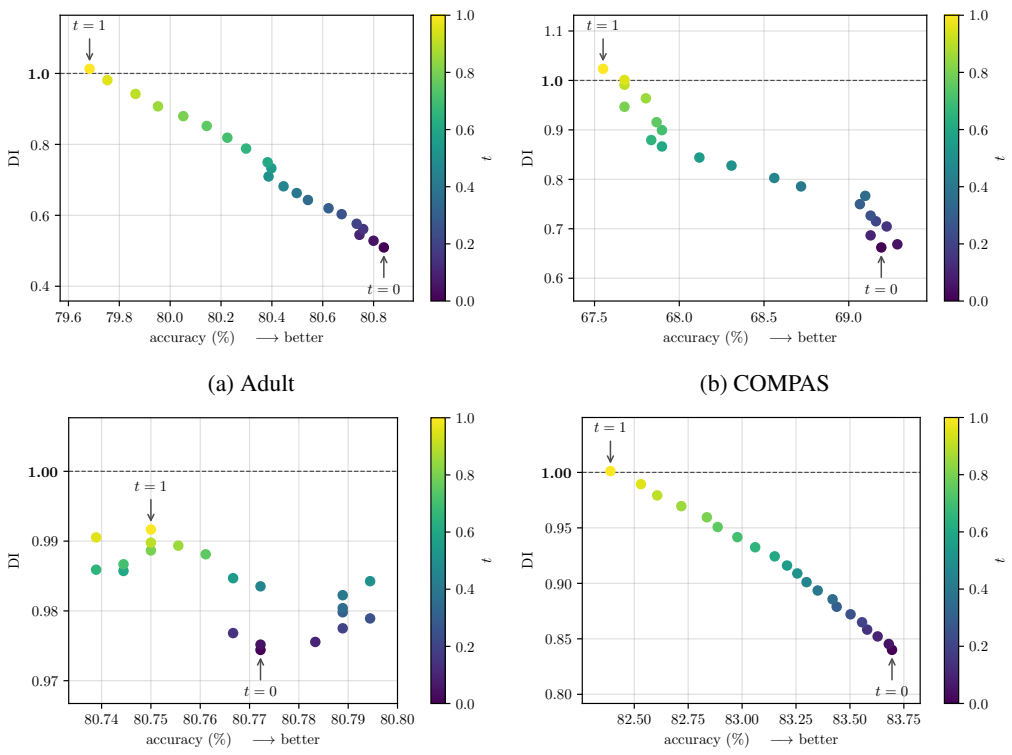


Figure 16: Fairness vs. classification accuracy tradeoff. For having a single view of methods performance across different datasets, we use the average rank on each axis of methods across datasets.

Fairness vs. Accuracy across CFM trajectories. An interesting consequence of computing a barycenter through CFM is that our algorithm is inherently dynamical. Since in our methodology we move each p_k to $\hat{q}_\lambda^{(L_{\text{FP}})}$ through a flow for $t \in [0, 1]$, we can measure, at each step of the ODE trajectory, the DI and classification accuracy of partially repaired (in the transport sense). We show these trajectories in Figure 17. At $t = 0$, we retrieve the baseline performance (i.e., no repair). as time progresses, we see a progressive overall improvement of DI towards values close to 1.0, and a degradation of downstream accuracy, again, highlighting the fairness-vs-task performance tradeoff (Zhao & Gordon, 2022).



(a) Adult (b) COMPAS

(c) Credit (d) NLSY

Figure 17: Disparate Impact (DI) vs. classification accuracy as a function of flow time. At $t = 0$, we have the starting point of samples at each p_k. As time evolves, these samples are progressively transported to the barycenter, which usually improves DI (models become more fair) at the cost of accuracy.

Hyperparameters and their sensitivity. We use the same training and inference configurations for each dataset, namely: $B = 1024$ (except for COMPAS, $B = 512$), $N = 30\,000$, $L_{\text{FP}} = L_{\text{D}} = 1$, Adam (Kingma & Ba, 2014) optimizer with $\text{LR}_{\text{C}} = \text{LR}_{\text{D}} = 10^{-3}$, entropic regularization $\epsilon = 10^{-3}$, # of ODE steps $T = 100$, and an Euler solver. Every dataset uses an Multilayer Perceptron (MLP) architecture with $\dim_h = 256$ and 4 layers in each head. These result in 645 062, 646 601, 653 783 and 645 318 parameters for Adult, COMPAS, Credit and NLSY respectively.

We study the parameter sensitivity of our `BaryFM` across 2 axis: universal vs. non-universal solver, and number of fixed point iterations L_{FP}. We show an overview in Figure 18, which has 6 variants, named $\{\text{U,Univ}\} - L_{\text{FP}}$, for $L_{\text{FP}} \in \{1, 5, 10\}$. Overall, 4 of the 6 variants rank consistently above the pareto-frontier defined by `DWB` (best fairness rank) – `NOT` – `CW2B` – `U-NOT` – `NWBU` (best accuracy rank). Especially, our Univ $-$ 10 (universal solver, $L_{\text{FP}} = 10$) has the best geometric mean fairness rank (1.86) against `DWB` (2.21), and is close to the methods with the best geometric mean accuracy ranks (ours 3.94 vs. `U-NOT` and `NWBU` with 3.87 and 3.53 respectively). In the end, we report our Univ $-$ 5 variant in Table 8, as it strikes the best balanced between the fair-

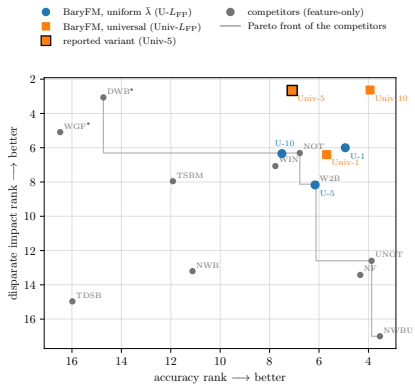


Figure 18: Fairness vs. classification accuracy tradeoff for `BaryFM` under different hyperparameters.

ness metric (DI) and the downstream task metric (classification accuracy).

G Domain Adaptation

We evaluate barycenter algorithms in terms of their *domain alignment performance* of extracted embeddings. In a nutshell, we fine-tune a model with the available source-domain data. This is either a pre-trained foundation model, such as CBraMod (Wang et al., 2025) for EEG data, ResNets (He et al., 2016) pretrained on ImageNet for image data, or a convolutional neural net from scratch for chemical engineering data. This initial pre-training minimizes the risk over the pooled source domain data, which can be written as,

$$\theta^\star = \arg\min_{\theta\in\Theta} \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\ell(y_{k,i}, h(x_{k,i})), \tag{34}$$

where $x_{k,i}$ is the $i-$th embedding from the $k-$th domain (resp. y for labels), and Θ is the space of neural net weights. This first fine-tuning stage serves the purpose of separating the classes in the different datasets. As explored in (Montesuma et al., 2026a, Appendix E), the domain alignment performed by barycenter algorithms happens in 2 different ways, depending on whether the barycenter measure is labeled or not.

G.1 Wasserstein Barycenter Transport

Originally proposed by Montesuma & Mboula (2021), the WBT algorithm solves multi-source domain adaptation in a 2 stage procedure. First, source domains are aggregated into the Wasserstein barycenter. Second, the Wasserstein barycenter is transported into the target domain. The original method was proposed with a labeled variant of the discrete algorithm of Cuturi & Doucet (2014), later formalized as a barycentric transport over the joint continuous variable $z = (x, y)$, where $x \in \mathbb{R}^d$ and $y \in \Delta_C$, for $C-$classes by Montesuma et al. (2023).

In Montesuma et al. (2026a), the authors have extended the WBT methodology for other barycenter solvers, allowing the comparison of barycenter solvers for the downstream task of domain adaptation. Therefore, the comparison along this application judge barycenter in terms of (1) their ability to align heterogeneous domains, and (2) their ability to respect class structure. There is a supplementary, 3rd axis that is how well the algorithms approximate the mappings from the marginal measures p_k to the barycenter support $q_\lambda^\star$. An illustration of the unsupervised and supervised mechanisms is shown in Figure 6.

In both unsupervised and supervised cases, the barycenter points must be aligned with the target domain. We isolate the quality of the calculated barycenter by subjecting all methods to the same alignment mechanism. Here, note that WBT is effectively reducing a multi-source domain adaptation problem to a single-source one. Therefore, we use Courty et al. (2016), which transports the support of the estimated barycenter $\hat{q}_\lambda$ to that of p_T,

$$T_{q_\lambda\to p_T}(\tilde{x}_{k,i}) = n_T\sum_{j=1}^{n_T}\gamma_{ij}^\star x_{T,j}. \qquad \gamma^\star = \text{OT}(q_\lambda^\star, p_T). \tag{35}$$

However, this method has limited scalability. For instance, in the context of DomainNet, domains such as `quickdraw` and `real` have around $170,000$ points, making the required OT solve unfeasible. We take a way around this issue by solving a CFM problem for aligning the barycenter with the target domain, meaning,

$$\theta_T^\star = \arg\min_{\theta\in\Theta} \mathbb{E}_{t,z_0,z_1}[\|v_\theta(x_t,t) - (x_1 - x_0)\|_2^2], \text{ where } t\sim\mathcal{U}[0,1] \text{ and } (x_0,x_1)\sim\text{OT}(\hat{q}_\lambda, p_T),$$

where $\text{OT}(\hat{q}_\lambda, p_T)$ is an OT solve between minibatches from the estimated barycenter $\hat{q}_\lambda$ and from p_T. Once trained, the barycenter samples are transported into the target domain via the same ODE defined in equation 19.

G.2 Experimental Details

Adaptation Protocol. Algorithms are evaluated in 2 protocols, depending on the established literature. For instance, in computer vision, datasets have a small number of domains (e.g., Office 31 has 3 domains), and thus, algorithms are evaluated in the Leave-One-Domain-Out (LODO) setting. Taking Office 31 as an example, this dataset has 3 domains: Amazon (A), dSLR (D) and Webcam (W). In the LODO protocol, one has 3 adaptation tasks: $\{\mathtt{D}, \mathtt{W}\} \to \mathtt{A}$, $\{\mathtt{A}, \mathtt{W}\} \to \mathtt{D}$, and $\{\mathtt{A}, \mathtt{W}\} \to \mathtt{W}$. Performance is evaluated and ranked in terms of *average target domain performance*.

The second adaptation protocol is *fixed target*, which commonly appears in cross-subject adaptation studies. In that case, there is a fixed partition of source subjects (i.e., subjects from which we have available labeled data and from which the barycenter is calculated) and target subjects (i.e., subjects to which adaptation is done). To keep results comparable to those reported by the backbone (Wang et al., 2025), we report *pooled target domain performance*.

Dataset	Adaptation Protocol	Backbone	# Features	# Classes	# Domains	# Samples	Seeds
Office 31	LODO	ResNet-50	2048	31	3	4,110	42, 43, 44
Office-Home	LODO	ResNet-101	2048	65	4	15,500	42, 43, 44
DomainNet	LODO	ResNet-101	2048	345	6	586,575	42, 43, 44
BCI-CIV-2a	Fixed Target	CBraMod	200	4	9	5,184	42, 43, 44
FACED	Fixed Target	CBraMod	200	9	123	10,332	42, 43, 44
SEED-VIG	Fixed Target	CBraMod	200	–	21	20,355	42, 43, 44
Mumtaz	Fixed Target	CBraMod	200	2	5	7,143	42, 43, 44
Physio	Fixed Target	CBraMod	200	4	109	9,837	42, 43, 44
SHU-MI	Fixed Target	CBraMod	200	2	25	11,988	42, 43, 44
TEP	LODO	CNN	128	29	6	17,289	42 (pre-defined 5 folds)

Table 9: Summary of datasets used in our experiments.

Architecture. All domain adaptation experiments use the same network architecture, which follows the design presented in Section 3. Our networks are composed of a shared encoder, that different layers to map x, y, k, λ into a shared latent space $\mathbb{R}^{\dim_h}$. In a nutshell, continuous inputs such as $x \in \mathbb{R}^{\dim_x}, y \in \Delta_C, \lambda \in \Delta_K$ are mapped into $\mathbb{R}^{\dim_h}$ through a dense layer. Since $k \in \{1, \cdots, K\}$ is inherently discrete, we use a lookup table with K vectors in $\mathbb{R}^{\dim_h}$.

On top of the latent vector $h = \text{Cat}(E_x, E_y, E_k, E_\lambda)$, we use 2 heads, a CFM and a dual OT head. The CFM head receives, additionally $h_t = \text{Cat}(h, E_t)$, where E_t is a sinusoidal position encoding of the time variable $t \in [0, 1]$. Both heads implement the same architecture, that is, a MLP with 4 layers, and SiLU activations. We summarize this discussion in Table 10.

Dataset	Hidden width	MLP embedding	x	t	y	k	λ	# Params
Office 31	512	✓	64	64	64	64	64	4.09M
Office-Home	512	✗	128	32	32	64	64	3.24M
DomainNet	512	✗	128	64	128	64	64	3.54M
BCI-CIV-2a	512	✓	64	64	64	64	64	2.18M
FACED	512	✓	128	64	64	64	64	2.36M
SEED-VIG	512	✓	128	64	32	128	64	2.38M
Mumtaz	512	✓	64	64	64	64	64	2.18M
Physio	512	✓	64	64	64	64	64	2.19M
SHU-MI	512	✓	64	64	64	64	64	2.18M
TEP	512	✓	64	64	64	64	64	2.12M

Table 10: Architecture of the network used in domain adaptation. Columns x, t, y, k and λ give the embedding dimensions of the features, time, labels, domain index and barycentric weights. The last column gives the total number of trainable parameters, including the dual OT head.

Hyperparameters. We divide the hyperparameters of our algorithm in 2 families. First, there are the training hyperparameters, which are: batch size B, number of iterations N, number of fixed

point iterations L_{FP}, number of dual iterations L_{D}, learning rate $\text{LR} = \text{LR}_C = \text{LR}_D$ (that is, we use a single learning rate for the whole network), and entropic regularization ϵ. We use the same architecture on all benchmarks, consisting of a densely connected multi-layer perceptron. We train this network using the AdamW (Kingma & Ba, 2014; Loshchilov & Hutter, 2017) optimizer with weight decay $\omega = 10^{-2}$. We show an overview in Table 11.

Benchmark	B	N	L_{Dual}	L_{FP}	LR	Weight Decay	ε
Office 31	310	10 000	1	1	10^{-3}	0.01	10^{-3}
Office Home	1300	10 000	1	1	10^{-4}	0	10^{-3}
DomainNet	2048	30 000	1	1	10^{-4}	0	10^{-3}
BCI-CIV-2a	400	10 000	1	1	10^{-3}	0.01	10^{-3}
FACED	256	10 000	1	1	10^{-4}	0.01	10^{-3}
SEED-VIG	1024	10 000	5	5	10^{-4}	0.01	10^{-3}
Mumtaz	1024	10 000	1	1	10^{-4}	0.01	10^{-3}
Physio	1024	10 000	1	1	10^{-4}	0.01	10^{-3}
SHU-MI	1024	10 000	1	1	10^{-4}	0.01	10^{-3}
TEP	400	10 000	1	1	10^{-3}	0.01	10^{-2}

Table 11: Training hyperparameters of `BaryFM` across benchmarks.

Second, there are inference hyper-parameters, which comprise number of generated samples, number of ODE steps, and ODE solver. In general, ODE steps > 10 suffice for obtaining the best performance, and there is not much difference in the ODE solver choice. We ablate these results in Figure 19. For simplicity, we use 100 steps with an Euler solver. The number of generated points, on the other hand, has more leverage on performance, as sampling more points in the barycenter support increases performance. For the sake of a fair comparison, we use the same number of points in each barycenter solver in Table 13.

	Office 31	Office Home	DomainNet	BCI-CIV-2a	FACED	SEED-VIG	Mumtaz	Physio	SHU-MI	TEP
n_{gen}	4 650	39 000	103 500	1 000	9 000	53 100	10 000	10 000	10 000	5 800
ODE steps	100	100	100	100	100	100	100	100	100	100
ODE Solver	Euler	Euler	Euler	Euler	Euler	Euler	Euler	Euler	Euler	Euler
Transport	`emd`	`emd`	`cfm`	`emd`	`emd`	`emd`	`emd`	`emd`	`emd`	`emd`

Table 12: Inference hyperparameters of `BaryFM` across benchmarks.

Main Results. Our main results are shown in Table 13, which comprehend the aggregate performance on each benchmark. Note that Office 31, Office Home, DomainNet and TEP report average target domain performance, and neuroscience datasets (BCI-CIV-2a, FACED, SEED-VIG, Mumtaz, Physio, SHU-MI) report pooled performance over target subjects. This reporting choices are done to conform to usual practice. See, for instance, how Wang et al. (2025) report results on multiple held-out subjects. As we mentioned in the main paper, our `BaryFM` achieves an average rank of 1.75, surpassing the previous state-of-the-art barycenter solver for domain adaptation, `WGF` (Montesuma et al., 2026a). We provide detailed descriptions of these benchmarks and their results per domain in Sections G.3.1 through G.3.10.

Ablations. In Table 14, we ablate the components of our `BaryFM` method, meaning: CFM, dual OT, joint flow ($\mathcal{X} \times \mathcal{Y}$) and universality. We provide further information on each variant in Section C. Our methodology is as follows: each ablation stage replaces `BaryFM` in Table 13 and is thus ranked independently (i.e., one variant is not ranked against the others). There are 2 components that stand out in Table 14, and degrade performance the most. The first component is the CFM head, which validates that learning the flow from each p_k to $\hat{q}_\lambda^{(L_{\text{FP}})}$ is necessary. The second is universality, which means that learning universal barycenters leads to a better estimation of the uniform barycenter. A similar result was obtained in the context of Bayesian Posterior Aggregation (c.f. Section E, Table 5). Other components (e.g., $\mathcal{X} \times \mathcal{Y}$ and the dual OT head) do improve performance, but not as much. In addition to these remarks, note that the worst variants (non-universal `BaryFM`, and `NeuralFP`) rank 3.50, which is still better than the 2nd best (`WGF`).

Method	$\mathcal{X} \times \mathcal{Y}$	Universal	Office31	OfficeHome	DomainNet	BCI-CIV-2a	FACED	SEED-VIG	Mumtaz	Physio	SHU-MI	TEP	Avg. Rank
Backbone	-	-	ResNet-50	ResNet-101				CBraMod				CNN	-
Source-Only	-	-	$86.58_{\pm 0.07}$	$76.93_{\pm 0.09}$	$51.23_{\pm 0.01}$	$52.72_{\pm 0.05}$	$54.87_{\pm 0.27}$	$44.72_{\pm 2.15}$	$91.25_{\pm 0.36}$	$64.10_{\pm 0.08}$	$62.68_{\pm 0.41}$	$79.18_{\pm 0.18}$	12.10
Discrete ✗	✗	✗	$85.76_{\pm 0.56}$	$75.95_{\pm 0.05}$	$50.46_{\pm 0.20}$	$62.62_{\pm 0.41}$	$54.47_{\pm 0.32}$	$34.66_{\pm 0.50}$	$91.30_{\pm 0.13}$	$64.95_{\pm 0.29}$	$55.89_{\pm 3.26}$	$84.54_{\pm 1.47}$	11.45
Discrete ✓	✓	✗	$85.46_{\pm 0.36}$	$76.09_{\pm 0.36}$	$51.01_{\pm 0.10}$	$62.50_{\pm 0.63}$	$54.43_{\pm 0.50}$	$34.76_{\pm 0.60}$	$91.30_{\pm 0.13}$	$64.97_{\pm 0.27}$	$53.80_{\pm 4.03}$	$85.23_{\pm 1.26}$	11.40
WGF ✗	✗	✗	$86.69_{\pm 0.33}$	$77.09_{\pm 0.28}$	$49.48_{\pm 0.04}$	$63.02_{\pm 0.23}$	$58.06_{\pm 0.24}$	$52.29_{\pm 0.62}$	$92.79_{\pm 0.41}$	$65.12_{\pm 0.40}$	$62.87_{\pm 0.90}$	$85.52_{\pm 1.48}$	5.40
WGF ✓	✓	✗	$88.90_{\pm 0.74}$	$78.03_{\pm 0.13}$	$52.08_{\pm 0.14}$	$62.56_{\pm 0.13}$	$56.99_{\pm 0.24}$	$51.74_{\pm 0.44}$	$91.77_{\pm 0.56}$	$65.54_{\pm 0.23}$	$63.70_{\pm 0.11}$	$86.87_{\pm 1.27}$	4.50
CW2B	✗	✗	$87.12_{\pm 1.27}$	$75.93_{\pm 0.19}$	$49.14_{\pm 0.69}$	$62.70_{\pm 0.35}$	$58.07_{\pm 0.18}$	$51.06_{\pm 0.74}$	$91.88_{\pm 0.19}$	$64.97_{\pm 0.14}$	$63.05_{\pm 0.20}$	$83.00_{\pm 1.55}$	7.30
WIN	✗	✗	$86.21_{\pm 0.53}$	$75.88_{\pm 0.17}$	$46.89_{\pm 0.49}$	$59.95_{\pm 3.69}$	$58.83_{\pm 0.43}$	$49.78_{\pm 2.47}$	$91.58_{\pm 0.08}$	$64.60_{\pm 0.03}$	$62.96_{\pm 0.32}$	$82.58_{\pm 1.95}$	11.05
NOT	✗	✗	$87.38_{\pm 0.85}$	$76.35_{\pm 0.54}$	$49.74_{\pm 0.07}$	$62.73_{\pm 0.10}$	$57.94_{\pm 0.11}$	$42.23_{\pm 2.74}$	$91.63_{\pm 0.13}$	$64.97_{\pm 0.28}$	$62.99_{\pm 0.52}$	$83.35_{\pm 2.10}$	7.80
U-NOT	✗	✗	$86.42_{\pm 0.67}$	$76.21_{\pm 0.12}$	$49.84_{\pm 0.15}$	$62.64_{\pm 0.81}$	$57.85_{\pm 0.26}$	$43.98_{\pm 1.59}$	$91.74_{\pm 0.22}$	$64.84_{\pm 0.17}$	$62.92_{\pm 0.21}$	$83.45_{\pm 1.92}$	9.30
TDSB	✗	✗	$86.74_{\pm 0.19}$	$74.97_{\pm 0.15}$	$52.56_{\pm 0.13}$	$62.64_{\pm 0.18}$	$57.04_{\pm 0.09}$	$49.58_{\pm 0.87}$	$91.74_{\pm 0.00}$	$64.87_{\pm 0.08}$	$63.14_{\pm 0.21}$	$83.76_{\pm 1.50}$	8.10
TSBM	✗	✗	$86.96_{\pm 0.04}$	$76.11_{\pm 0.30}$	$52.59_{\pm 0.02}$	$62.50_{\pm 0.31}$	$55.99_{\pm 0.65}$	$55.03_{\pm 0.35}$	$92.07_{\pm 0.22}$	$64.78_{\pm 0.34}$	$63.06_{\pm 0.70}$	$84.42_{\pm 1.24}$	6.75
NormFlow	✗	✗	$86.88_{\pm 0.28}$	$75.91_{\pm 0.08}$	$52.28_{\pm 0.07}$	$62.44_{\pm 0.22}$	$58.01_{\pm 0.13}$	$46.13_{\pm 1.83}$	$91.80_{\pm 0.05}$	$64.74_{\pm 0.03}$	$62.95_{\pm 0.26}$	$83.92_{\pm 1.30}$	8.55
GMM	✓	✗	$87.18_{\pm 0.99}$	$75.62_{\pm 0.57}$	$51.72_{\pm 0.02}$	$61.95_{\pm 0.52}$	$55.49_{\pm 0.07}$	$41.25_{\pm 1.87}$	$91.58_{\pm 0.08}$	$65.54_{\pm 0.08}$	$63.49_{\pm 0.46}$	$85.34_{\pm 0.63}$	8.20
NWB	✗	✗	$85.07_{\pm 0.67}$	$73.31_{\pm 0.37}$	$38.47_{\pm 4.23}$	$62.41_{\pm 0.54}$	$57.38_{\pm 0.08}$	$39.60_{\pm 2.25}$	$91.80_{\pm 0.17}$	$65.04_{\pm 0.19}$	$63.14_{\pm 0.60}$	$83.36_{\pm 1.65}$	10.50
NWBU	✗	✓	$84.71_{\pm 0.81}$	$74.29_{\pm 0.36}$	$37.41_{\pm 1.82}$	$62.70_{\pm 0.88}$	$57.06_{\pm 0.21}$	$37.15_{\pm 2.92}$	$91.14_{\pm 0.75}$	$65.00_{\pm 0.06}$	$63.05_{\pm 0.12}$	$80.57_{\pm 1.50}$	11.85
BaryFM	✓	✓	$89.74_{\pm 0.23}$	$78.72_{\pm 0.12}$	$52.56_{\pm 0.07}$	$64.53_{\pm 0.65}$	$58.25_{\pm 0.03}$	$57.42_{\pm 0.94}$	$93.12_{\pm 0.59}$	$65.00_{\pm 0.11}$	$63.70_{\pm 0.15}$	$87.30_{\pm 1.39}$	1.75

Table 13: Downstream performance of WBT per barycenter solver. Results report $\mu_{\pm\sigma}$ over 3 seeds. Avg. Rank averages the per-dataset ranks.

				Office31	OfficeHome	DomainNet	BCI-CIV-2a	FACED	SEED-VIG	Mumtaz	Physio	SHU-MI	TEP	Avg. Rank
CFM	Dual OT	$\mathcal{X} \times \mathcal{Y}$	Universal	ResNet-50	ResNet-101				CBraMod				CNN	-
✗	✓	✓	✓	$88.53_{\pm 0.11}$	$78.61_{\pm 0.02}$	$52.18_{\pm 0.09}$	$64.93_{\pm 0.19}$	$56.78_{\pm 0.10}$	$47.65_{\pm 3.33}$	$91.94_{\pm 0.17}$	$\mathbf{65.15}_{\pm 0.09}$	$63.63_{\pm 0.49}$	$87.07_{\pm 1.32}$	3.50
✓	✗	✓	✓	$89.09_{\pm 0.33}$	$78.68_{\pm 0.04}$	$52.54_{\pm 0.12}$	$64.47_{\pm 0.28}$	$\mathbf{59.27}_{\pm 0.16}$	$53.52_{\pm 0.49}$	$93.12_{\pm 0.27}$	$64.98_{\pm 0.21}$	$63.61_{\pm 0.66}$	$87.26_{\pm 1.15}$	1.90
✓	✓	✗	✓	$89.03_{\pm 0.18}$	$78.60_{\pm 0.07}$	$\mathbf{52.70}_{\pm 0.04}$	$\mathbf{65.22}_{\pm 0.53}$	$57.95_{\pm 0.39}$	$52.15_{\pm 0.65}$	$92.05_{\pm 0.48}$	$65.00_{\pm 0.17}$	$63.61_{\pm 0.30}$	$\mathbf{87.32}_{\pm 1.20}$	2.35
✓	✓	✓	✗	$87.88_{\pm 0.27}$	$78.65_{\pm 0.09}$	$52.45_{\pm 0.04}$	$64.79_{\pm 0.18}$	$57.82_{\pm 0.23}$	$49.99_{\pm 0.06}$	$92.35_{\pm 0.99}$	$64.89_{\pm 0.17}$	$63.47_{\pm 0.28}$	$87.14_{\pm 1.22}$	3.50
✓	✓	✓	✓	$\mathbf{89.74}_{\pm 0.23}$	$\mathbf{78.72}_{\pm 0.12}$	$52.56_{\pm 0.07}$	$64.53_{\pm 0.65}$	$58.25_{\pm 0.03}$	$\mathbf{57.42}_{\pm 0.94}$	$\mathbf{93.12}_{\pm 0.59}$	$65.00_{\pm 0.11}$	$\mathbf{63.70}_{\pm 0.15}$	$87.30_{\pm 1.39}$	1.75

Table 14: Architectural ablation of `BaryFM`. Results report $\mu_{\pm\sigma}$ over 3 seeds. The rank of each method is computed by replacing `BaryFM` in Table 13; Avg. Rank averages the per-dataset ranks.

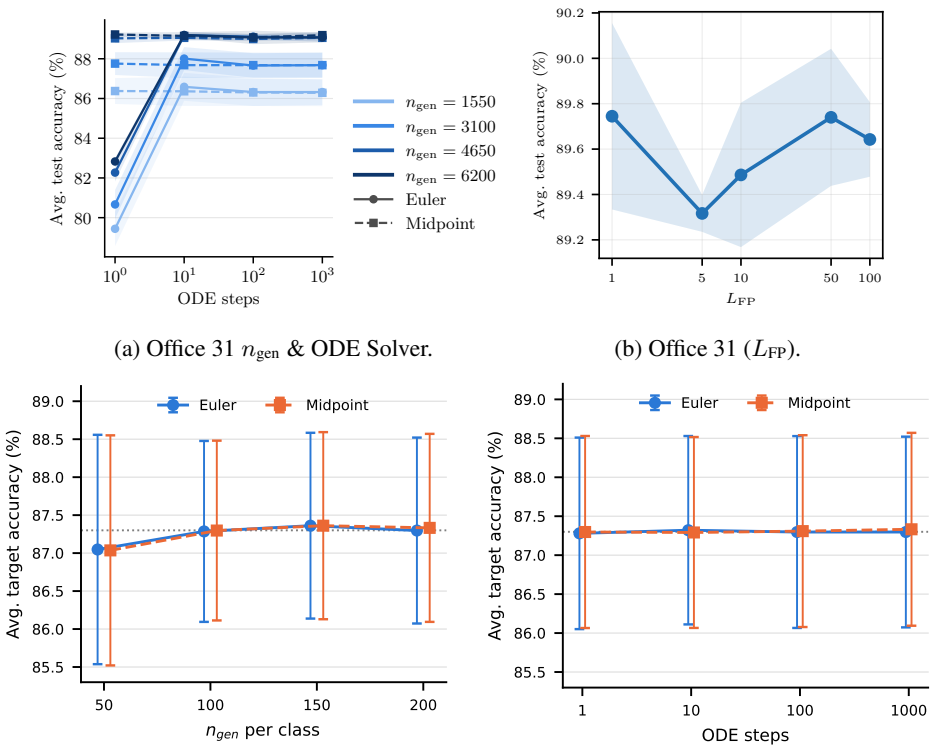
Hyperparameter Sensitivity. Besides the specific components of our `BaryFM`, we investigate how our method is sensitive to inference hyper-parameters, namely n_{gen}, ODE solver, number of ODE steps T, and number of fixed-point iterations L_{FP}. Furthermore, L_{FP} does not have much effect on the downstream performance, as it oscillates between an average domain performance of 89.2 and 89.7. For simplicity, we use $L_{\text{FP}} = 1$ on most benchmarks. From the curves in Figure 19 (a), we see that the main driver of performance in Office 31 is the number of generated points, while for TEP, performance is globally stable across n_{gen}. The choice of ODE solver is also less important, except for extremely small T on Office 31 (c.f. Figure 19 (a), ODE steps = 10^0). Overall, we conclude that our algorithm is generally robust to the ODE hyper-parameters, and that n_{gen} sensitivity depends on the tested dataset.

Number of Parameters. We count the number of parameters in each tested neural method across the 10 benchmarks. We summarize the results in Table 15. Furthermore, we note that FACED and Physio demand significantly more parameters of neural networks that use $\mathcal{O}(K)$ networks to approximate Wasserstein barycenters, such as `CW2B` (Korotin et al., 2021) or `TSBM` (Howard et al., 2025). For instance, BCI-IV-2a (5 sources) demands $\approx 10^6$ parameters for `CW2B` and $\approx 6.9 \times 10^6$ for `TSBM`, while FACED demand $\approx 1.6 \times 10^7$ parameters and $\approx 1.11 \times 10^8$, respectively.

Method	Office31	OfficeHome	DomainNet	BCI-IV-2a	FACED	SEED-VIG	Mumtaz	Physio	SHU-MI	TEP
CW2B	3,957,504	5,936,256	9,893,760	1,023,360	16,373,760	2,660,736	1,023,360	14,327,040	3,070,080	677,760
WIN	3,979,140	5,688,966	9,108,618	1,518,882	23,033,832	3,813,810	1,518,882	20,165,172	4,387,542	1,223,178
NOT	1,643,778	2,465,667	4,109,445	552,045	8,832,720	1,435,317	552,045	7,728,630	1,656,135	413,445
U-NOT	1,643,778	2,465,667	4,109,445	552,045	8,832,720	1,435,317	552,045	7,728,630	1,656,135	413,445
TDSB	2,453,760	3,680,640	6,134,400	2,567,760	41,084,160	6,676,176	2,567,760	35,948,640	7,703,280	2,428,800
TSBM	8,946,688	13,420,032	22,366,720	6,941,520	111,064,320	18,047,952	6,941,520	97,181,280	20,824,560	3,168,000
NormFlow	3,658,176	3,696,560	1,723,904	201,152	359,552	218,048	201,152	338,432	222,272	463,760
NWB	2,428,356	3,506,246	5,662,026	674,274	10,275,624	1,698,418	674,274	8,995,444	1,232,424	277,984
NWBU	1,959,300	2,801,350	4,486,986	681,954	11,872,104	1,738,354	681,954	10,213,684	2,007,574	533,706
`NeuralFP`	2,004,545	1,201,441	1,293,121	1,057,025	1,132,545	1,140,641	1,056,897	1,065,345	1,058,177	1,021,761
$\texttt{BaryFM}_{\text{Emp}}$	3,169,439	2,301,281	2,552,985	1,260,044	1,403,729	1,407,721	1,258,890	1,268,364	1,260,170	1,200,669
$\texttt{BaryFM}_{\text{unsup}}$	4,005,953	3,169,473	3,186,113	2,112,137	2,285,705	2,343,561	2,112,137	2,120,457	2,113,417	2,038,337
`BaryFM` (no Universal)	4,089,504	3,237,730	3,538,586	2,180,109	2,356,562	2,376,938	2,178,955	2,188,429	2,180,235	2,120,734
`BaryFM`	4,089,504	3,237,730	3,538,586	2,180,109	2,356,562	2,376,938	2,178,955	2,188,429	2,180,235	2,120,734

Table 15: Number of trainable parameters of neural barycenter solvers.

In comparison, our proposed architectures remain bounded as a function of the number of marginal measures (e.g., $\approx 2.1 \times 10^6$ for BCI-IV-2a, and $\approx 2.3 \times 10^6$ for FACED). From Table 15, the only other method that is able to do this is `NormFlow` (Visentin & Cheridito, 2025), who relies on the multi-scale architecture of Dinh et al. (2016).



(a) Office 31 n_{gen} & ODE Solver.

(b) Office 31 (L_{FP}).

(c) TEP (n_{gen})

(d) TEP (T)

Figure 19: Hyper-parameter sensitivity to n_{gen}, ODE solver, # of fixed-point iterations (L_{FP}), and # of ODE steps (T) on Office 31 and TEP benchmarks.

Overall, Table 15 shows a gap in the architecture design of neural network solvers, as there is no current consensus. Future work could focus on several computational questions around this issue, such as (i) scaling laws, and (ii) theoretically justified architectures.

G.3 Detailed Results and Dataset Descriptions

G.3.1 Office 31

Proposed by Saenko et al. (2010), this dataset consists of images from 3 domains, Amazon, dSLR and Webcam. Images from Amazon were scraped from the web, while dSLR and Webcam are taken from a high-resolution camera and a webcam, respectively. As a result, there are at least these factors of variation across domains: quality, pose, background, and illumination. These factors induce different shifts in the underlying data, motivating the domain adaptation scenario. There is a grand total of 4652 images in this benchmark, with 2817 on Amazon, 498 in dSLR, and 795 in Webcam. We show the detailed results in Table 16. We use ResNets (He et al., 2016) (50 variant) as the backbone for extracting embeddings. We finetune the backbone with the available labeled source domain data, then adapt to each target using unlabeled data.

Method	CFM	Dual OT	$\mathcal{X} \times \mathcal{Y}$	Universal	Amazon	dSLR	Webcam	Avg.	Rank
Source-Only	-	-	-	-	$64.23_{\pm 0.36}$	$99.41_{\pm 0.51}$	$96.10_{\pm 0.68}$	$86.58_{\pm 0.07}$	10
Discrete ✗	-	-	✗	✗	$70.67_{\pm 0.40}$	$92.26_{\pm 0.52}$	$94.35_{\pm 0.89}$	$85.76_{\pm 0.56}$	13
Discrete ✓	-	-	✓	✗	$69.98_{\pm 0.61}$	$91.67_{\pm 1.36}$	$94.74_{\pm 1.01}$	$85.46_{\pm 0.36}$	14
WGF ✗	-	-	✗	✗	$69.63_{\pm 0.56}$	$93.75_{\pm 0.89}$	$96.69_{\pm 0.34}$	$86.69_{\pm 0.33}$	9
WGF ✓	-	-	✓	✗	$70.33_{\pm 0.44}$	$98.51_{\pm 1.36}$	$97.86_{\pm 0.68}$	$88.90_{\pm 0.74}$	2
CW2B	-	-	✗	✗	$69.05_{\pm 1.22}$	$95.83_{\pm 2.73}$	$96.49_{\pm 0.58}$	$87.12_{\pm 1.27}$	5
WIN	-	-	✗	✗	$69.57_{\pm 1.92}$	$93.15_{\pm 1.36}$	$95.91_{\pm 1.01}$	$86.21_{\pm 0.53}$	12
NOT	-	-	✗	✗	$70.03_{\pm 0.46}$	$95.24_{\pm 2.25}$	$96.88_{\pm 0.68}$	$87.38_{\pm 0.85}$	3
U-NOT	-	-	✗	✗	$70.38_{\pm 0.63}$	$93.75_{\pm 2.36}$	$95.13_{\pm 0.89}$	$86.42_{\pm 0.67}$	11
TDSB	-	-	✗	✗	$70.85_{\pm 0.10}$	$94.05_{\pm 0.52}$	$95.32_{\pm 0.00}$	$86.74_{\pm 0.19}$	8
TSBM	-	-	✗	✗	$70.03_{\pm 0.46}$	$94.94_{\pm 1.36}$	$95.91_{\pm 1.01}$	$86.96_{\pm 0.04}$	6
NormFlow	-	-	✗	✗	$70.09_{\pm 0.36}$	$94.05_{\pm 1.03}$	$96.49_{\pm 0.58}$	$86.88_{\pm 0.28}$	7
GMM	-	-	✓	✗	$71.25_{\pm 0.91}$	$95.54_{\pm 1.55}$	$94.74_{\pm 0.58}$	$87.18_{\pm 0.99}$	4
NWB	-	-	✗	✗	$69.57_{\pm 0.56}$	$92.26_{\pm 2.06}$	$93.37_{\pm 0.34}$	$85.07_{\pm 0.67}$	15
NWBU	-	-	✗	✓	$67.83_{\pm 1.48}$	$91.37_{\pm 2.25}$	$94.93_{\pm 1.79}$	$84.71_{\pm 0.81}$	16
NeuralFP	✗	✓	✓	✓	$68.93_{\pm 0.44}$	$98.81_{\pm 0.52}$	$97.86_{\pm 0.34}$	$88.53_{\pm 0.11}$	2
BaryFM$_{\text{Emp}}$	✓	✗	✓	✓	$69.63_{\pm 0.53}$	$98.81_{\pm 0.52}$	$98.83_{\pm 0.00}$	$89.09_{\pm 0.33}$	1
BaryFM$_{\text{unsup}}$	✓	✓	✗	✓	$69.05_{\pm 0.40}$	$100.00_{\pm 0.00}$	$98.05_{\pm 0.89}$	$89.03_{\pm 0.18}$	1
BaryFM (no Universal)	✓	✓	✓	✗	$67.36_{\pm 0.36}$	$98.81_{\pm 0.52}$	$97.47_{\pm 0.34}$	$87.88_{\pm 0.27}$	2
BaryFM	✓	✓	✓	✓	$70.21_{\pm 0.76}$	$100.00_{\pm 0.00}$	$99.03_{\pm 0.89}$	$89.74_{\pm 0.23}$	1

Table 16: Per-domain Office 31 results, along average over domains, and rank.

From Table 16, all variants of `BaryFM` attain either the 1st, or 2nd place among methods, with 2nd best being `WGF` of Montesuma et al. (2026a). Note that, naturally, the transfer towards dSLR and Webcam are easier, since they have, in the source domains, a closely resembling labeled domain (Webcam for dSLR, and dSLR for Webcam), however, most unlabeled solvers degrade performance in these domains (e.g., `NWBU` has $91.37_{\pm 2.25}$ on dSLR, significantly lower than the source-only baseline at $99.41_{\pm 0.51}$). Our `BaryFM` is the only method among the compared solvers that manages to improve performance on the most difficult domain (Amazon), while also improving performance on the other domains (dSLR and Webcam).

G.3.2 OFFICE HOME

Proposed by Venkateswara et al. (2017), this benchmark is a visual adaptation benchmark with 4 domains, Art, Clipart, Product, and Real World, with 2427, 4365, 4439 and 4357 images each. Images in this benchmark have a different kind of shift with respect to Office 31. Indeed, the different domains in Office Home account for *stylistic shifts*, that is, a spoon in art is a work of art while in real world, it is a photo of a spoon. This generally induces a stronger shift between domains, as well as some domains (e.g., Clipart) have less separated classes. Our obtained results are shown in Table 17. Like Office 31, we use ResNets (He et al., 2016) (101 variant) as the backbone for extracting embeddings. We finetune the backbone with the available labeled source domain data, then adapt to each target using unlabeled data.

Method	CFM	Dual OT	$\mathcal{X} \times \mathcal{Y}$	Universal	Art	Clipart	Product	Real World	Avg.	Rank
Source-Only	-	-	-	-	$75.55_{\pm 0.25}$	$63.21_{\pm 0.15}$	$83.57_{\pm 0.01}$	$85.40_{\pm 0.07}$	$76.93_{\pm 0.09}$	4
Discrete ✗	-	-	✗	✗	$69.63_{\pm 0.86}$	$65.29_{\pm 0.51}$	$84.12_{\pm 0.64}$	$84.78_{\pm 0.50}$	$75.95_{\pm 0.05}$	9
Discrete ✓	-	-	✓	✗	$70.15_{\pm 1.28}$	$65.73_{\pm 0.22}$	$83.28_{\pm 0.11}$	$85.19_{\pm 0.39}$	$76.09_{\pm 0.36}$	8
WGF ✗	-	-	✗	✗	$75.24_{\pm 1.24}$	$64.76_{\pm 0.89}$	$83.33_{\pm 0.23}$	$85.02_{\pm 0.99}$	$77.09_{\pm 0.28}$	3
WGF ✓	-	-	✓	✗	$75.45_{\pm 0.43}$	$65.06_{\pm 0.11}$	$85.32_{\pm 0.33}$	$86.28_{\pm 0.24}$	$78.03_{\pm 0.13}$	2
CW2B	-	-	✗	✗	$71.07_{\pm 0.82}$	$64.95_{\pm 0.62}$	$82.81_{\pm 1.02}$	$84.89_{\pm 0.63}$	$75.93_{\pm 0.19}$	10
WIN	-	-	✗	✗	$70.74_{\pm 1.19}$	$65.02_{\pm 0.39}$	$82.62_{\pm 0.29}$	$85.12_{\pm 0.74}$	$75.88_{\pm 0.17}$	12
NOT	-	-	✗	✗	$71.99_{\pm 1.38}$	$64.99_{\pm 0.66}$	$82.81_{\pm 0.66}$	$85.63_{\pm 0.49}$	$76.35_{\pm 0.54}$	5
U-NOT	-	-	✗	✗	$71.40_{\pm 0.20}$	$64.50_{\pm 0.30}$	$83.61_{\pm 0.44}$	$85.34_{\pm 0.17}$	$76.21_{\pm 0.12}$	6
TDSB	-	-	✗	✗	$68.18_{\pm 0.30}$	$63.94_{\pm 0.22}$	$83.10_{\pm 0.13}$	$84.67_{\pm 0.17}$	$74.97_{\pm 0.15}$	14
TSBM	-	-	✗	✗	$71.60_{\pm 1.10}$	$65.02_{\pm 0.28}$	$82.66_{\pm 0.33}$	$85.15_{\pm 0.13}$	$76.11_{\pm 0.30}$	7
NormFlow	-	-	✗	✗	$71.60_{\pm 0.20}$	$64.87_{\pm 0.45}$	$82.48_{\pm 0.33}$	$84.71_{\pm 0.45}$	$75.91_{\pm 0.08}$	11
GMM	-	-	✓	✗	$69.63_{\pm 0.90}$	$63.49_{\pm 1.16}$	$85.25_{\pm 0.33}$	$84.11_{\pm 0.68}$	$75.62_{\pm 0.57}$	13
NWB	-	-	✗	✗	$70.22_{\pm 0.79}$	$59.84_{\pm 4.03}$	$82.51_{\pm 1.14}$	$80.66_{\pm 2.71}$	$73.31_{\pm 0.37}$	16
NWBU	-	-	✗	✓	$67.85_{\pm 0.99}$	$63.31_{\pm 0.36}$	$82.15_{\pm 1.05}$	$83.85_{\pm 0.29}$	$74.29_{\pm 0.36}$	15
NeuralFP	✗	✓	✓	✓	$76.65_{\pm 0.06}$	$65.84_{\pm 0.02}$	$85.51_{\pm 0.17}$	$86.46_{\pm 0.07}$	$78.61_{\pm 0.02}$	1
BaryFM$_{\text{Emp}}$	✓	✗	✓	✓	$77.24_{\pm 0.13}$	$65.79_{\pm 0.11}$	$85.45_{\pm 0.00}$	$86.24_{\pm 0.14}$	$78.68_{\pm 0.04}$	1
BaryFM$_{\text{unsup}}$	✓	✓	✗	✓	$76.90_{\pm 0.13}$	$65.76_{\pm 0.11}$	$85.29_{\pm 0.10}$	$86.47_{\pm 0.07}$	$78.60_{\pm 0.07}$	1
BaryFM (no Universal)	✓	✓	✓	✗	$76.91_{\pm 0.29}$	$66.04_{\pm 0.12}$	$85.39_{\pm 0.12}$	$86.25_{\pm 0.11}$	$78.65_{\pm 0.09}$	1
BaryFM	✓	✓	✓	✓	$77.12_{\pm 0.20}$	$65.84_{\pm 0.29}$	$85.52_{\pm 0.07}$	$86.38_{\pm 0.12}$	$78.72_{\pm 0.12}$	1

Table 17: Per-domain Office-Home results, along average over domains, and rank.

In this benchmark, `BaryFM` and all its variants achieve the best performance, being very close to each other (e.g., worst `BaryFM`$_{\text{unsup}}$ with $78.60_{\pm 0.07}$ vs. `BaryFM` $78.72_{\pm 0.12}$). All of these results cluster well above the 2nd placed method, `WGF`, at $78.03_{\pm 0.13}$. One important point in this comparison, the source-only baseline ranks 4th, surpassing many methods. Indeed, only `WGF` variants and ours manage to improve over source-only.

G.3.3 DOMAIN NET

Proposed by (Peng et al., 2019), this benchmark is the largest, in number of samples, of the vision adaptation benchmarks. It comprises 6 domains, Clipart, Infograph, Painting, Quickdraw, Real, Sketch, with 48129, 51605, 72266, 172500, 172947, and 69128 images, in a grand total of 586575 images divided across 345 classes. This benchmark pushes the limits of visual adaptation, having a higher diversity of images, classes, and styles. For instance, the most difficult domain, Quickdraw, consists of drawings collected from players of the game *Quick, Draw!* Like other benchmarks, we use ResNets (He et al., 2016) (101 variant) as the backbone for extracting embeddings. We finetune the backbone with the available labeled source domain data, then adapt to each target using unlabeled data.

Method	CFM	Dual OT	$\mathcal{X} \times \mathcal{Y}$	Universal	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Avg.	Rank
Source-Only	-	-	-	-	$71.75_{\pm 0.02}$	$28.63_{\pm 0.03}$	$58.73_{\pm 0.00}$	$17.34_{\pm 0.01}$	$71.07_{\pm 0.00}$	$59.86_{\pm 0.00}$	$51.23_{\pm 0.01}$	7
Discrete ✗	-	-	✗	✗	$69.61_{\pm 0.38}$	$27.33_{\pm 0.32}$	$55.97_{\pm 0.32}$	$21.33_{\pm 0.63}$	$70.84_{\pm 0.29}$	$57.72_{\pm 0.36}$	$50.46_{\pm 0.20}$	9
Discrete ✓	-	-	✓	✗	$70.81_{\pm 0.58}$	$27.19_{\pm 0.23}$	$56.64_{\pm 0.41}$	$21.28_{\pm 0.41}$	$72.06_{\pm 0.17}$	$58.06_{\pm 0.21}$	$51.01_{\pm 0.10}$	8
WGF ✗	-	-	✗	✗	$67.88_{\pm 0.25}$	$27.44_{\pm 0.06}$	$55.28_{\pm 0.46}$	$20.70_{\pm 0.23}$	$68.72_{\pm 0.04}$	$56.85_{\pm 0.19}$	$49.48_{\pm 0.04}$	12
WGF ✓	-	-	✓	✗	$71.28_{\pm 0.29}$	$29.46_{\pm 0.42}$	$58.02_{\pm 0.45}$	$22.41_{\pm 0.36}$	$72.46_{\pm 0.19}$	$58.88_{\pm 0.09}$	$52.08_{\pm 0.14}$	5
CW2B	-	-	✗	✗	$69.01_{\pm 1.52}$	$26.10_{\pm 0.36}$	$55.56_{\pm 1.17}$	$16.98_{\pm 0.79}$	$69.63_{\pm 0.35}$	$57.56_{\pm 0.74}$	$49.14_{\pm 0.69}$	13
WIN	-	-	✗	✗	$66.80_{\pm 0.50}$	$23.76_{\pm 0.12}$	$53.60_{\pm 0.63}$	$14.14_{\pm 1.08}$	$67.67_{\pm 0.55}$	$55.39_{\pm 0.93}$	$46.89_{\pm 0.49}$	14
NOT	-	-	✗	✗	$69.62_{\pm 0.36}$	$26.75_{\pm 0.38}$	$55.71_{\pm 0.14}$	$17.66_{\pm 0.57}$	$70.95_{\pm 0.08}$	$57.77_{\pm 0.07}$	$49.74_{\pm 0.07}$	11
U-NOT	-	-	✗	✗	$69.19_{\pm 0.07}$	$27.43_{\pm 0.13}$	$56.20_{\pm 0.09}$	$18.14_{\pm 0.19}$	$70.28_{\pm 0.38}$	$57.78_{\pm 0.27}$	$49.84_{\pm 0.15}$	10
TDSB	-	-	✗	✗	$71.34_{\pm 0.34}$	$30.64_{\pm 0.28}$	$58.28_{\pm 0.14}$	$23.22_{\pm 0.09}$	$72.04_{\pm 0.12}$	$59.82_{\pm 0.13}$	$52.56_{\pm 0.13}$	2
TSBM	-	-	✗	✗	$71.54_{\pm 0.12}$	$30.11_{\pm 0.08}$	$58.45_{\pm 0.16}$	$23.16_{\pm 0.11}$	$72.14_{\pm 0.20}$	$60.14_{\pm 0.10}$	$52.59_{\pm 0.02}$	1
NormFlow	-	-	✗	✗	$71.06_{\pm 0.21}$	$29.96_{\pm 0.19}$	$58.01_{\pm 0.17}$	$23.09_{\pm 0.19}$	$71.98_{\pm 0.13}$	$59.60_{\pm 0.11}$	$52.28_{\pm 0.07}$	4
GMM	-	-	✓	✗	$70.44_{\pm 0.12}$	$28.05_{\pm 0.03}$	$56.17_{\pm 0.05}$	$24.90_{\pm 0.10}$	$72.98_{\pm 0.09}$	$57.76_{\pm 0.10}$	$51.72_{\pm 0.02}$	6
NWB	-	-	✗	✗	$59.79_{\pm 2.90}$	$20.70_{\pm 6.09}$	$41.58_{\pm 11.44}$	$8.54_{\pm 3.53}$	$62.93_{\pm 1.34}$	$37.28_{\pm 12.42}$	$38.47_{\pm 4.23}$	15
NWBU	-	-	✗	✓	$50.44_{\pm 6.13}$	$18.55_{\pm 2.44}$	$38.02_{\pm 10.76}$	$11.20_{\pm 3.05}$	$55.64_{\pm 3.73}$	$50.61_{\pm 1.36}$	$37.41_{\pm 1.82}$	16
NeuralFP	✗	✓	✓	✓	$72.16_{\pm 0.03}$	$27.99_{\pm 0.21}$	$57.67_{\pm 0.08}$	$22.83_{\pm 0.44}$	$73.39_{\pm 0.12}$	$59.04_{\pm 0.26}$	$52.18_{\pm 0.09}$	4
BaryFM$_{\text{Emp}}$	✓	✗	✓	✓	$72.41_{\pm 0.28}$	$28.69_{\pm 0.12}$	$57.95_{\pm 0.18}$	$23.24_{\pm 0.15}$	$73.23_{\pm 0.07}$	$59.71_{\pm 0.11}$	$52.54_{\pm 0.12}$	3
BaryFM$_{\text{unsup}}$	✓	✓	✗	✓	$72.38_{\pm 0.04}$	$29.15_{\pm 0.14}$	$57.94_{\pm 0.18}$	$23.44_{\pm 0.07}$	$73.48_{\pm 0.01}$	$59.84_{\pm 0.02}$	$52.70_{\pm 0.04}$	1
BaryFM (no Universal)	✓	✓	✓	✗	$72.34_{\pm 0.24}$	$28.66_{\pm 0.09}$	$57.78_{\pm 0.15}$	$23.15_{\pm 0.22}$	$73.22_{\pm 0.12}$	$59.56_{\pm 0.10}$	$52.45_{\pm 0.04}$	3
BaryFM	✓	✓	✓	✓	$72.36_{\pm 0.26}$	$28.81_{\pm 0.23}$	$57.89_{\pm 0.09}$	$23.33_{\pm 0.16}$	$73.22_{\pm 0.09}$	$59.74_{\pm 0.05}$	$52.56_{\pm 0.07}$	3

Table 18: Per-domain DomainNet results, along average over domains, and rank.

The difficulty of this benchmark makes the gains overall marginal. For instance, only one empirical solver manages to improve over the source-only baseline, `WGF`, achieving $52.08_{\pm 0.14}$ against $51.23_{\pm 0.01}$. Furthermore, unsupervised solvers perform best: `TSBM` ranks 1st with $52.59_{\pm 0.02}$, and `TDSB` ties with our full `BaryFM` at $52.56_{\pm 0.13}$ and $52.56_{\pm 0.07}$, respectively. Consistently with this remark, our `BaryFM`$_{\text{unsup}}$ variant is the best among our pool of variants ($52.70_{\pm 0.04}$, c.f. Table 14), which overall suggests an edge of unsupervised barycenter solvers in this benchmark.

G.3.4 BCI-CIV-2A

This benchmark was proposed by Brunner et al. (2008) as the dataset A within the BCI competition IV. This benchmark contains electroencephalogram (EEG) recordings, that is, time series of the form $x \in \mathbb{R}^{V \times T}$, of V variates per T time steps. Each EEG recording is associated with a *motor imagery task*, namely, imagining a movement. There are 4 movements in this benchmark: left hand, right hand, both feet and tongue. Recordings were acquired with $V = 22$ electrodes at 250Hz over 2 sessions on separate days, with 288 trials per session. We refer readers to Wang et al. (2025) for further details on the preprocessing and overall backbone design.

The main driver of heterogeneity in this benchmark, as with other EEG datasets, is *cross-subject variability*, that is, the difference in the measured signals due to the different morphology or state-of-mind of subjects at the moment of data collection. Following Wang et al. (2025), we train using data from subjects $1, 2, \cdots, 5$, use subjects $6, 7$ for checkpoint selection. Subjects 8 and 9 are reserved for test, and are never seen during the fine-tuning of the backbone. We report poled accuracy, which makes Table 19 comparable to the results reported in Wang et al. (2025).

Method	CFM	Dual OT	$\mathcal{X} \times \mathcal{Y}$	Universal	S8	S9	Pool.	Rank
Source-Only	-	-	-	-	$54.98_{\pm 0.10}$	$50.46_{\pm 0.10}$	$52.72_{\pm 0.05}$	16
Discrete ✗	-	-	✗	✗	$66.72_{\pm 0.44}$	$58.51_{\pm 0.46}$	$62.62_{\pm 0.41}$	8
Discrete ✓	-	-	✓	✗	$66.67_{\pm 0.76}$	$58.33_{\pm 0.97}$	$62.50_{\pm 0.63}$	10
WGF ✗	-	-	✗	✗	$66.90_{\pm 0.20}$	$59.14_{\pm 0.66}$	$63.02_{\pm 0.23}$	2
WGF ✓	-	-	✓	✗	$66.15_{\pm 0.17}$	$58.97_{\pm 0.20}$	$62.56_{\pm 0.13}$	9
CW2B	-	-	✗	✗	$66.38_{\pm 0.27}$	$59.03_{\pm 0.63}$	$62.70_{\pm 0.35}$	4
WIN	-	-	✗	✗	$65.51_{\pm 1.52}$	$54.40_{\pm 5.99}$	$59.95_{\pm 3.69}$	15
NOT	-	-	✗	✗	$66.61_{\pm 0.78}$	$58.85_{\pm 0.97}$	$62.73_{\pm 0.10}$	3
U-NOT	-	-	✗	✗	$66.38_{\pm 0.80}$	$58.91_{\pm 1.18}$	$62.64_{\pm 0.81}$	6
TDSB	-	-	✗	✗	$66.67_{\pm 0.17}$	$58.62_{\pm 0.20}$	$62.64_{\pm 0.18}$	6
TSBM	-	-	✗	✗	$66.55_{\pm 0.36}$	$58.45_{\pm 0.36}$	$62.50_{\pm 0.31}$	10
NormFlow	-	-	✗	✗	$66.72_{\pm 0.10}$	$58.16_{\pm 0.46}$	$62.44_{\pm 0.22}$	12
GMM	-	-	✓	✗	$65.45_{\pm 0.52}$	$58.45_{\pm 0.86}$	$61.95_{\pm 0.52}$	14
NWB	-	-	✗	✗	$66.49_{\pm 0.30}$	$58.33_{\pm 1.04}$	$62.41_{\pm 0.54}$	13
NWBU	-	-	✗	✓	$66.15_{\pm 0.17}$	$59.26_{\pm 1.60}$	$62.70_{\pm 0.88}$	4
NeuralFP	✗	✓	✓	✓	$68.52_{\pm 0.30}$	$61.34_{\pm 0.16}$	$64.93_{\pm 0.19}$	1
$\text{BaryFM}_{\text{Emp}}$	✓	✗	✓	✓	$68.00_{\pm 0.53}$	$60.94_{\pm 0.30}$	$64.47_{\pm 0.28}$	1
$\text{BaryFM}_{\text{unsup}}$	✓	✓	✗	✓	$67.77_{\pm 0.20}$	$62.67_{\pm 0.92}$	$65.22_{\pm 0.53}$	1
BaryFM (no Universal)	✓	✓	✓	✗	$68.52_{\pm 0.20}$	$61.05_{\pm 0.27}$	$64.79_{\pm 0.18}$	1
BaryFM	✓	✓	✓	✓	$68.34_{\pm 0.40}$	$60.71_{\pm 0.99}$	$64.53_{\pm 0.65}$	1

Table 19: Per-domain BCI-IV-2a cross-subject results, along pooled performance, and rank.

Overall, all methods improve over the baseline. This is natural, since the finetuning of the baseline does not have access to the target domain data, and the domain shift is large. Our `Baryfm` methods all achieve the best results in this benchmark, followed by `WGF` (unsupervised). The best results are achieved with $\texttt{BaryFM}_{\text{unsup}}$.

G.3.5 FACED

This benchmark was proposed by Chen et al. (2023), and is composed of EEGs with $V = 32$ electrodes, at 250Hz, collected from 123 subjects, which makes it the benchmark with the most number of domains. Each EEG corresponds to one of 9 classes: amusement, inspiration, joy, tenderness, anger, fear, disgust, sadness and a neutral state in a balanced coverage of positive and negative emotions. We refer readers to Wang et al. (2025) for the detailed description on the data preprocessing and the backbone for this benchmark. Like these authors, we use subjects 1-80 for training, 81-100 for checkpoint selection, and 101-123 for testing. Table 20 shows the per subject results alongside the pooled performance of each method.

As with other EEG benchmarks (e.g. SEED-VIG and SHU-MI), there is a significant variation in the performance on each individual. In the case of FACED, as discussed in Chen et al. (2023), this variation is induced by differences in people's emotional experiences.

In this benchmark, most empirical approaches, except for unsupervised `WGF` (Montesuma et al., 2026a), perform poorly. In contrast, neural methods tend to perform better, especially `WIN` (Korotin et al., 2022a). In comparison, our `BaryFM` is a close 2nd competitor ($58.25_{\pm 0.03}$ ours vs. $58.83_{\pm 0.43}$ theirs), and our empirical variant surpasses `WIN` ($59.27_{\pm 0.16}$ ours vs. $58.83_{\pm 0.43}$ theirs). Overall, solvers improve over the source-only baseline for most subjects, except for a few subjects (e.g., 103, 119), see Figure 20.

Method	CFM	Dual OT	$\mathcal{X} \times \mathcal{Y}$	Universal	S101	S102	S103	S104	S105	S106	S107	S108	S109	S110	S111	S112	S113	S114	S115	S116	S117	S118	S119	S120	S121	S122	S123	Pool	Rank
Source-Only	-	-	-	-	$38.89_{\pm 0.69}$	$40.87_{\pm 0.69}$	$57.94_{\pm 0.69}$	$51.98_{\pm 1.37}$	$37.30_{\pm 0.69}$	$65.08_{\pm 0.69}$	$48.02_{\pm 0.69}$	$66.27_{\pm 0.69}$	$51.59_{\pm 0.69}$	$60.71_{\pm 0.00}$	$50.79_{\pm 1.37}$	$51.59_{\pm 0.69}$	$78.97_{\pm 0.69}$	$63.10_{\pm 0.00}$	$58.33_{\pm 0.00}$	$72.62_{\pm 0.00}$	$64.68_{\pm 0.69}$	$44.44_{\pm 0.69}$	$38.10_{\pm 0.69}$	$54.37_{\pm 0.69}$	$47.62_{\pm 0.00}$	$65.08_{\pm 0.69}$	$53.57_{\pm 0.00}$	$54.87_{\pm 0.27}$	14
Discrete ✗	-	-	✗	✗	$38.10_{\pm 1.19}$	$38.51_{\pm 0.69}$	$54.37_{\pm 0.00}$	$51.59_{\pm 0.69}$	$39.68_{\pm 0.69}$	$66.67_{\pm 2.06}$	$47.22_{\pm 3.00}$	$65.48_{\pm 2.38}$	$51.98_{\pm 3.15}$	$57.94_{\pm 1.37}$	$50.40_{\pm 1.37}$	$53.57_{\pm 2.38}$	$78.17_{\pm 1.37}$	$59.52_{\pm 4.12}$	$63.49_{\pm 0.69}$	$75.40_{\pm 0.69}$	$66.27_{\pm 0.69}$	$46.43_{\pm 2.06}$	$33.73_{\pm 3.83}$	$54.37_{\pm 1.37}$	$47.22_{\pm 1.37}$	$61.11_{\pm 1.37}$	$53.17_{\pm 1.82}$	$54.47_{\pm 0.32}$	15
Discrete ✓	-	-	✓	✗	$37.70_{\pm 1.44}$	$34.81_{\pm 1.19}$	$54.37_{\pm 1.37}$	$50.46_{\pm 0.69}$	$40.08_{\pm 1.44}$	$67.46_{\pm 1.19}$	$47.22_{\pm 1.37}$	$67.86_{\pm 1.19}$	$50.79_{\pm 1.37}$	$55.95_{\pm 1.37}$	$51.59_{\pm 1.37}$	$55.16_{\pm 1.23}$	$79.37_{\pm 1.37}$	$61.11_{\pm 1.82}$	$63.49_{\pm 1.19}$	$75.00_{\pm 1.19}$	$66.67_{\pm 1.19}$	$46.43_{\pm 1.19}$	$34.13_{\pm 0.69}$	$54.37_{\pm 1.37}$	$48.41_{\pm 1.37}$	$60.71_{\pm 1.19}$	$53.57_{\pm 1.19}$	$54.64_{\pm 0.30}$	16
WGF ✗	-	-	✗	✗	$42.06_{\pm 2.48}$	$36.90_{\pm 1.19}$	$59.92_{\pm 2.48}$	$57.54_{\pm 1.37}$	$40.48_{\pm 1.19}$	$71.43_{\pm 1.19}$	$51.98_{\pm 3.15}$	$71.03_{\pm 1.82}$	$61.90_{\pm 3.03}$	$57.94_{\pm 1.82}$	$55.95_{\pm 1.19}$	$57.94_{\pm 1.82}$	$80.56_{\pm 1.37}$	$67.06_{\pm 1.37}$	$64.68_{\pm 1.19}$	$79.37_{\pm 2.17}$	$71.03_{\pm 1.19}$	$46.83_{\pm 0.69}$	$38.49_{\pm 2.48}$	$57.94_{\pm 2.75}$	$48.41_{\pm 1.37}$	$65.08_{\pm 1.19}$	$54.76_{\pm 1.19}$	$58.06_{\pm 0.54}$	4
WGF ✓	-	-	✓	✗	$40.48_{\pm 0.00}$	$36.51_{\pm 0.69}$	$58.33_{\pm 1.19}$	$57.54_{\pm 1.82}$	$39.68_{\pm 1.82}$	$70.24_{\pm 0.00}$	$47.62_{\pm 1.19}$	$66.27_{\pm 0.69}$	$53.57_{\pm 1.19}$	$62.30_{\pm 0.69}$	$53.57_{\pm 2.06}$	$56.75_{\pm 1.37}$	$80.56_{\pm 0.69}$	$65.87_{\pm 1.37}$	$65.48_{\pm 1.19}$	$76.19_{\pm 1.19}$	$69.44_{\pm 0.69}$	$47.62_{\pm 0.00}$	$37.30_{\pm 1.82}$	$54.37_{\pm 0.69}$	$50.00_{\pm 1.19}$	$65.48_{\pm 2.38}$	$55.56_{\pm 0.69}$	$56.99_{\pm 0.24}$	11
CW2B	-	-	✗	✓	$42.86_{\pm 0.00}$	$38.10_{\pm 0.00}$	$58.73_{\pm 1.82}$	$59.13_{\pm 0.69}$	$37.70_{\pm 0.69}$	$69.05_{\pm 0.00}$	$50.79_{\pm 0.69}$	$62.70_{\pm 0.69}$	$53.97_{\pm 0.69}$	$57.54_{\pm 1.82}$	$83.73_{\pm 0.69}$	$66.67_{\pm 0.00}$	$68.25_{\pm 0.69}$	$79.76_{\pm 1.19}$	$71.03_{\pm 0.69}$	$47.62_{\pm 0.00}$	$36.90_{\pm 1.19}$	$55.95_{\pm 1.19}$	$49.60_{\pm 0.69}$	$65.08_{\pm 0.00}$	$55.56_{\pm 0.69}$	$58.07_{\pm 0.18}$	3
WIN	-	-	✗	✗	$40.48_{\pm 1.19}$	$43.25_{\pm 2.48}$	$59.92_{\pm 1.37}$	$59.92_{\pm 2.69}$	$41.27_{\pm 1.82}$	$71.03_{\pm 0.69}$	$50.79_{\pm 1.82}$	$69.84_{\pm 0.69}$	$56.75_{\pm 0.69}$	$62.70_{\pm 2.48}$	$56.75_{\pm 1.37}$	$57.94_{\pm 0.69}$	$83.33_{\pm 0.00}$	$67.06_{\pm 1.82}$	$65.48_{\pm 0.00}$	$76.98_{\pm 0.69}$	$72.62_{\pm 1.19}$	$48.41_{\pm 1.37}$	$36.90_{\pm 2.06}$	$59.52_{\pm 0.00}$	$51.98_{\pm 0.69}$	$65.08_{\pm 0.00}$	$55.16_{\pm 1.37}$	$58.83_{\pm 0.43}$	1
NOT	-	-	✗	✗	$41.27_{\pm 0.69}$	$37.70_{\pm 0.69}$	$58.33_{\pm 0.00}$	$59.92_{\pm 2.69}$	$37.70_{\pm 1.19}$	$70.24_{\pm 0.00}$	$50.79_{\pm 0.69}$	$68.65_{\pm 1.37}$	$55.95_{\pm 0.00}$	$61.51_{\pm 0.69}$	$52.78_{\pm 1.37}$	$58.33_{\pm 1.19}$	$82.14_{\pm 0.00}$	$66.27_{\pm 0.69}$	$68.65_{\pm 0.69}$	$78.57_{\pm 1.19}$	$71.43_{\pm 1.19}$	$46.83_{\pm 0.69}$	$38.10_{\pm 0.00}$	$55.95_{\pm 2.38}$	$50.79_{\pm 0.69}$	$65.48_{\pm 1.19}$	$55.95_{\pm 0.69}$	$57.94_{\pm 0.11}$	6
UDA-OT	-	-	✗	✗	$42.06_{\pm 1.37}$	$36.90_{\pm 0.00}$	$58.33_{\pm 1.82}$	$59.13_{\pm 1.82}$	$39.68_{\pm 1.82}$	$69.05_{\pm 1.19}$	$51.98_{\pm 0.00}$	$67.06_{\pm 1.82}$	$54.37_{\pm 0.00}$	$62.70_{\pm 0.69}$	$53.97_{\pm 0.69}$	$57.14_{\pm 1.19}$	$82.14_{\pm 0.69}$	$65.87_{\pm 1.37}$	$65.48_{\pm 0.69}$	$76.19_{\pm 1.19}$	$71.43_{\pm 0.00}$	$46.43_{\pm 0.69}$	$36.11_{\pm 0.69}$	$55.56_{\pm 0.69}$	$49.60_{\pm 0.69}$	$64.29_{\pm 1.37}$	$55.56_{\pm 0.00}$	$57.04_{\pm 0.00}$	10
TDSB	-	-	✗	✗	$40.08_{\pm 1.37}$	$38.49_{\pm 0.00}$	$55.56_{\pm 0.69}$	$57.14_{\pm 0.69}$	$39.68_{\pm 1.82}$	$69.05_{\pm 1.19}$	$51.98_{\pm 0.00}$	$67.06_{\pm 1.82}$	$54.37_{\pm 0.69}$	$62.70_{\pm 0.69}$	$53.97_{\pm 0.69}$	$57.14_{\pm 1.19}$	$82.14_{\pm 0.69}$	$65.08_{\pm 1.37}$	$65.48_{\pm 0.69}$	$75.00_{\pm 1.19}$	$71.43_{\pm 0.00}$	$46.43_{\pm 0.69}$	$36.11_{\pm 0.69}$	$55.56_{\pm 0.00}$	$49.60_{\pm 0.69}$	$64.29_{\pm 1.37}$	$55.56_{\pm 0.00}$	$57.04_{\pm 0.00}$	10
TSBM	-	-	✗	✗	$40.48_{\pm 0.00}$	$41.27_{\pm 2.48}$	$58.73_{\pm 1.37}$	$51.98_{\pm 1.37}$	$37.30_{\pm 3.00}$	$69.44_{\pm 2.48}$	$48.41_{\pm 1.82}$	$65.48_{\pm 1.19}$	$51.59_{\pm 0.69}$	$60.71_{\pm 2.28}$	$51.98_{\pm 3.64}$	$55.95_{\pm 0.00}$	$78.97_{\pm 1.82}$	$62.30_{\pm 4.18}$	$64.29_{\pm 1.19}$	$76.19_{\pm 3.15}$	$70.63_{\pm 1.37}$	$45.24_{\pm 2.38}$	$48.81_{\pm 2.38}$	$56.75_{\pm 1.82}$	$62.70_{\pm 2.75}$	$54.76_{\pm 2.06}$	$55.99_{\pm 0.65}$	12
NormPkw	-	-	✗	✗	$42.46_{\pm 0.69}$	$36.90_{\pm 0.00}$	$58.73_{\pm 0.69}$	$59.13_{\pm 1.82}$	$37.30_{\pm 1.37}$	$70.63_{\pm 0.69}$	$50.40_{\pm 0.69}$	$68.65_{\pm 0.69}$	$55.95_{\pm 0.00}$	$63.10_{\pm 1.19}$	$54.37_{\pm 0.69}$	$57.94_{\pm 0.69}$	$82.94_{\pm 1.37}$	$66.27_{\pm 0.69}$	$67.06_{\pm 0.69}$	$77.78_{\pm 0.69}$	$71.83_{\pm 0.69}$	$47.62_{\pm 0.00}$	$38.10_{\pm 0.00}$	$54.76_{\pm 0.00}$	$50.40_{\pm 0.69}$	$65.87_{\pm 0.69}$	$55.95_{\pm 1.19}$	$58.01_{\pm 0.13}$	5
GWM	-	-	✓	✗	$39.68_{\pm 0.84}$	$36.90_{\pm 1.68}$	$54.37_{\pm 2.53}$	$58.33_{\pm 1.37}$	$37.70_{\pm 1.68}$	$66.07_{\pm 2.53}$	$47.62_{\pm 1.68}$	$67.06_{\pm 2.53}$	$51.79_{\pm 0.00}$	$60.71_{\pm 0.00}$	$51.57_{\pm 1.68}$	$78.57_{\pm 1.68}$	$61.90_{\pm 0.00}$	$61.71_{\pm 0.00}$	$72.02_{\pm 0.84}$	$69.05_{\pm 0.00}$	$48.21_{\pm 0.84}$	$33.57_{\pm 1.68}$	$53.37_{\pm 1.68}$	$47.62_{\pm 0.00}$	$63.49_{\pm 1.68}$	$53.17_{\pm 0.00}$	$57.49_{\pm 0.07}$	13
NWA	-	-	✓	✗	$40.08_{\pm 0.00}$	$36.90_{\pm 1.68}$	$58.73_{\pm 1.37}$	$57.54_{\pm 0.00}$	$38.89_{\pm 1.82}$	$67.86_{\pm 0.69}$	$47.62_{\pm 0.69}$	$67.06_{\pm 1.82}$	$53.97_{\pm 0.69}$	$61.51_{\pm 0.69}$	$51.59_{\pm 0.69}$	$57.14_{\pm 0.69}$	$81.75_{\pm 0.69}$	$64.68_{\pm 0.00}$	$65.48_{\pm 1.37}$	$77.78_{\pm 0.69}$	$69.05_{\pm 0.00}$	$46.83_{\pm 1.37}$	$36.90_{\pm 0.00}$	$55.16_{\pm 0.00}$	$47.62_{\pm 0.00}$	$65.48_{\pm 0.00}$	$55.16_{\pm 0.00}$	$57.06_{\pm 0.08}$	8
NWBU	-	-	✗	✓	$40.48_{\pm 2.06}$	$36.11_{\pm 1.37}$	$58.33_{\pm 1.19}$	$55.95_{\pm 1.19}$	$39.29_{\pm 1.19}$	$68.65_{\pm 1.82}$	$50.40_{\pm 2.75}$	$65.87_{\pm 0.69}$	$53.97_{\pm 0.69}$	$62.30_{\pm 1.82}$	$53.97_{\pm 0.69}$	$57.94_{\pm 0.69}$	$81.75_{\pm 0.69}$	$65.48_{\pm 0.00}$	$65.48_{\pm 0.00}$	$75.79_{\pm 0.69}$	$70.63_{\pm 0.00}$	$46.83_{\pm 1.82}$	$36.51_{\pm 0.69}$	$55.56_{\pm 2.75}$	$49.21_{\pm 1.37}$	$64.68_{\pm 0.69}$	$55.56_{\pm 0.69}$	$57.06_{\pm 0.21}$	
NoratIPP	✗	✓	✓	✓	$42.06_{\pm 0.69}$	$35.71_{\pm 2.19}$	$51.19_{\pm 2.19}$	$55.56_{\pm 1.37}$	$38.10_{\pm 0.00}$	$68.65_{\pm 0.69}$	$52.38_{\pm 0.00}$	$68.25_{\pm 0.69}$	$51.98_{\pm 1.37}$	$61.90_{\pm 0.00}$	$53.57_{\pm 0.00}$	$56.35_{\pm 0.69}$	$84.92_{\pm 1.37}$	$64.29_{\pm 0.00}$	$64.68_{\pm 0.69}$	$76.98_{\pm 1.37}$	$69.84_{\pm 0.69}$	$47.62_{\pm 1.19}$	$37.30_{\pm 0.00}$	$57.94_{\pm 1.82}$	$48.81_{\pm 1.19}$	$63.49_{\pm 1.37}$	$54.37_{\pm 0.69}$	$56.78_{\pm 0.10}$	11
BaryFM$_{\text{step}}$	✓	✓	✓	✓	$42.86_{\pm 0.00}$	$40.08_{\pm 0.69}$	$54.76_{\pm 0.00}$	$58.33_{\pm 0.00}$	$40.48_{\pm 0.00}$	$70.63_{\pm 0.69}$	$53.97_{\pm 0.69}$	$68.65_{\pm 0.69}$	$55.56_{\pm 0.69}$	$63.49_{\pm 1.37}$	$56.35_{\pm 0.00}$	$57.14_{\pm 0.00}$	$86.51_{\pm 0.69}$	$66.67_{\pm 0.00}$	$68.25_{\pm 0.69}$	$80.56_{\pm 0.69}$	$72.62_{\pm 0.00}$	$53.17_{\pm 0.69}$	$37.70_{\pm 0.00}$	$61.11_{\pm 0.00}$	$51.19_{\pm 0.00}$	$63.89_{\pm 0.00}$	$59.13_{\pm 0.09}$	$59.27_{\pm 0.16}$	1
BaryFM$_{\text{drop}}$	✓	✓	✓	✓	$41.27_{\pm 0.69}$	$37.30_{\pm 1.37}$	$55.56_{\pm 0.69}$	$56.75_{\pm 0.69}$	$39.29_{\pm 1.19}$	$69.84_{\pm 0.69}$	$52.78_{\pm 1.82}$	$68.65_{\pm 1.82}$	$51.59_{\pm 0.69}$	$61.90_{\pm 0.00}$	$55.95_{\pm 1.19}$	$56.35_{\pm 0.69}$	$84.13_{\pm 0.69}$	$66.67_{\pm 0.00}$	$68.25_{\pm 2.48}$	$79.37_{\pm 0.69}$	$71.83_{\pm 1.37}$	$50.40_{\pm 1.37}$	$37.70_{\pm 0.69}$	$59.13_{\pm 0.69}$	$48.41_{\pm 0.69}$	$64.29_{\pm 1.19}$	$55.56_{\pm 0.69}$	$57.95_{\pm 0.39}$	5
BaryFM (no Universal)	✓	✓	✓	✗	$42.06_{\pm 0.69}$	$38.90_{\pm 1.19}$	$55.56_{\pm 1.82}$	$56.75_{\pm 0.69}$	$39.68_{\pm 0.69}$	$70.24_{\pm 0.00}$	$53.17_{\pm 1.37}$	$68.25_{\pm 1.82}$	$51.98_{\pm 1.37}$	$61.90_{\pm 0.00}$	$55.16_{\pm 0.69}$	$57.14_{\pm 0.00}$	$84.52_{\pm 0.00}$	$66.27_{\pm 1.37}$	$65.48_{\pm 1.19}$	$78.57_{\pm 1.19}$	$71.43_{\pm 1.19}$	$51.19_{\pm 0.00}$	$36.90_{\pm 1.19}$	$58.73_{\pm 0.00}$	$49.21_{\pm 0.00}$	$63.89_{\pm 0.00}$	$57.82_{\pm 0.23}$	7
BaryFM	✓	✓	✓	✓	$41.67_{\pm 0.00}$	$38.49_{\pm 0.00}$	$53.97_{\pm 0.69}$	$57.14_{\pm 0.00}$	$40.48_{\pm 1.19}$	$69.84_{\pm 0.69}$	$53.17_{\pm 0.69}$	$69.84_{\pm 0.69}$	$53.57_{\pm 0.00}$	$61.90_{\pm 0.00}$	$55.56_{\pm 0.00}$	$56.35_{\pm 0.69}$	$84.13_{\pm 0.69}$	$66.67_{\pm 0.00}$	$67.46_{\pm 0.69}$	$79.37_{\pm 1.37}$	$72.22_{\pm 0.69}$	$51.19_{\pm 0.00}$	$36.51_{\pm 0.69}$	$59.52_{\pm 0.00}$	$49.60_{\pm 0.69}$	$64.29_{\pm 1.19}$	$56.75_{\pm 1.37}$	$58.25_{\pm 0.03}$	2

Table 20: Detailed results on the FACED benchmark.

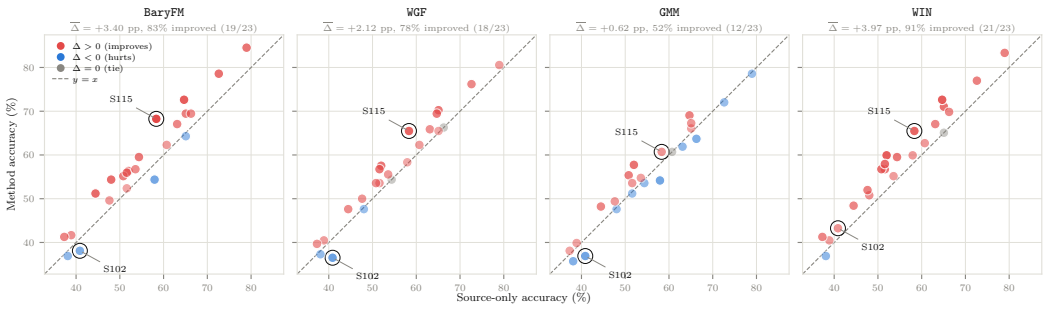


Figure 20: Source-only accuracy vs. method accuracy, for $\texttt{BaryFM}_{\text{Emp}}$, `BaryFM`, `WIN` and `CW2B`, per test subject. The bisecting line defines the no-improvement region. Points under this line (e.g., 119), in blue, show subjects with no improved performance. Subjects on top of that line (e.g., 115) show subjects with improved performance.

G.3.6 SEED-VIG

Introduced in Zheng & Lu (2017), SEED-VIG is a dataset for measuring vigilance estimation in a simulated driving task. Subjects drive a virtual system on a real car with a large screen while EEG is recorded. Labels are continuous, PERCLOS index, that is, the proportion of time during which eyes are closed, measures via eye tracking glasses. Compared to the other tested benchmarks, SEED-VIG is the only regression downstream task, hence we report the Pearson correlation (in %) between the ground-truth index and the predicted one. EEGs in this benchmark are measured with $V = 17$ electrodes at 200Hz, with a total of 20355 recordings across 21 subjects. We use subjects 1-13 for training, 14-17 for checkpoint selection, and the last 4 subjects (18, 19, 20, 21) for test. Our results are shown in Table 21.

Method	CFM	Dual OT	$\mathcal{X} \times \mathcal{Y}$	Universal	S18	S19	S20	S21	Pool.	Rank
Source-Only	-	-	-	-	$87.19_{\pm 0.56}$	$7.13_{\pm 7.81}$	$27.26_{\pm 9.24}$	$70.33_{\pm 2.16}$	$44.72_{\pm 2.15}$	9
Discrete ✗	-	-	✗	✗	$82.93_{\pm 0.48}$	$-4.43_{\pm 0.11}$	$40.98_{\pm 0.68}$	$64.49_{\pm 0.29}$	$34.66_{\pm 0.50}$	16
Discrete ✓	-	-	✓	✗	$82.39_{\pm 0.22}$	$-8.97_{\pm 0.44}$	$39.70_{\pm 0.81}$	$64.94_{\pm 0.85}$	$34.76_{\pm 0.60}$	15
WGF ✗	-	-	✗	✗	$78.83_{\pm 0.56}$	$13.64_{\pm 1.34}$	$41.61_{\pm 1.42}$	$70.10_{\pm 0.09}$	$52.29_{\pm 0.62}$	3
WGF ✓	-	-	✓	✗	$78.09_{\pm 1.52}$	$12.28_{\pm 0.14}$	$38.78_{\pm 2.26}$	$70.89_{\pm 1.00}$	$51.74_{\pm 0.44}$	4
CW2B	-	-	✗	✗	$83.97_{\pm 0.48}$	$-20.12_{\pm 1.32}$	$43.87_{\pm 0.50}$	$72.42_{\pm 0.38}$	$51.06_{\pm 0.74}$	5
WIN	-	-	✗	✗	$81.56_{\pm 1.16}$	$-11.79_{\pm 9.07}$	$44.73_{\pm 2.63}$	$68.11_{\pm 3.49}$	$49.78_{\pm 2.47}$	6
NOT	-	-	✗	✗	$79.62_{\pm 2.71}$	$-15.07_{\pm 2.65}$	$43.98_{\pm 5.34}$	$65.58_{\pm 3.03}$	$42.23_{\pm 2.74}$	11
U-NOT	-	-	✗	✗	$80.12_{\pm 0.23}$	$-21.65_{\pm 0.04}$	$41.10_{\pm 0.96}$	$68.45_{\pm 0.49}$	$43.98_{\pm 1.59}$	10
TDSB	-	-	✗	✗	$77.82_{\pm 0.28}$	$-18.89_{\pm 0.54}$	$40.27_{\pm 0.92}$	$67.31_{\pm 0.54}$	$49.58_{\pm 0.87}$	7
TSBM	-	-	✗	✗	$81.98_{\pm 0.59}$	$-18.29_{\pm 0.39}$	$42.45_{\pm 0.80}$	$70.34_{\pm 0.54}$	$55.03_{\pm 0.35}$	2
NormFlow	-	-	✗	✗	$78.32_{\pm 0.52}$	$-19.47_{\pm 0.88}$	$40.05_{\pm 0.25}$	$66.79_{\pm 0.80}$	$46.13_{\pm 1.83}$	8
GMM	-	-	✓	✗	$84.81_{\pm 4.45}$	$-12.21_{\pm 5.66}$	$55.62_{\pm 2.03}$	$64.52_{\pm 2.55}$	$41.25_{\pm 1.87}$	12
NWB	-	-	✗	✗	$80.51_{\pm 1.58}$	$-17.46_{\pm 3.94}$	$39.21_{\pm 9.60}$	$64.62_{\pm 4.97}$	$39.60_{\pm 2.25}$	13
NWBU	-	-	✗	✓	$80.24_{\pm 1.62}$	$-16.12_{\pm 6.67}$	$38.81_{\pm 0.63}$	$60.60_{\pm 7.84}$	$37.15_{\pm 2.92}$	14
NeuralFP	✗	✓	✓	✓	$83.42_{\pm 2.57}$	$8.56_{\pm 3.26}$	$38.60_{\pm 2.18}$	$66.76_{\pm 2.34}$	$47.65_{\pm 3.33}$	7
$\text{BaryFM}_{\text{Emp}}$	✓	✗	✓	✓	$82.85_{\pm 0.46}$	$13.60_{\pm 0.64}$	$39.00_{\pm 0.63}$	$72.26_{\pm 0.40}$	$53.52_{\pm 0.49}$	2
$\text{BaryFM}_{\text{unsup}}$	✓	✓	✗	✓	$80.44_{\pm 1.82}$	$11.73_{\pm 1.52}$	$46.26_{\pm 1.36}$	$68.76_{\pm 3.35}$	$52.15_{\pm 0.65}$	3
BaryFM (no Universal)	✓	✓	✓	✗	$83.14_{\pm 0.34}$	$8.76_{\pm 0.40}$	$38.53_{\pm 1.62}$	$68.92_{\pm 1.19}$	$49.99_{\pm 0.06}$	5
BaryFM	✓	✓	✓	✓	$87.02_{\pm 0.61}$	$11.52_{\pm 2.30}$	$45.94_{\pm 1.72}$	$70.48_{\pm 0.96}$	$57.42_{\pm 0.94}$	1

Table 21: Per-domain SEED-VIG results, along pooled performance, and rank.

In this benchmark, our `BaryFM` has a wider performance gap to the 2nd best, `TSBM` ($57.42_{\pm 0.94}$ ours vs. $55.03_{\pm 0.35}$, theirs).

G.3.7 MUMTAZ

This dataset was publicly released by Mumtaz (2016), and consists of a public record of resting-state EEG for the diagnosis of major depressive disorder. Following the description of Wang et al. (2025), the EEGs in this benchmark consists of $V = 19$ electrodes at 256Hz from 34 patients diagnosed with major depressive disorder, and 30 healthy control patients, with a total of 7143 samples, and 10 patients held out for test (5 healthy, and 5 patients with major depressive disorder).

Unlike other benchmarks, Mumtaz subjects contain a single class, which makes the Wasserstein geometry, in general, not well suited for coping with heterogeneity. For instance, computing the barycenter of a healthy patient and a patient with major depressive disorder would lead to fuzzy labels that, by design, do not respect the class structure in the embedding space. To fix that issue, we group source patients into 5 distinct cohorts. We apply the same methodology for each barycenter solver. Likewise, we group all target subjects into a single cohort. We show the grouping in Table 22.

Domain	#H	#MDD	# Samples	Healthy subjects	MDD subjects
p_1	5	5	1135	H 21, 24, 25, 27, 28	MDD 15, 17, 18, 27, 28
p_2	4	5	990	H 13, 16, 19, 23	MDD 10, 11, 12, 22, 25
p_3	4	4	906	H 1, 17, 22, 26	MDD 1, 2, 16, 19
p_4	4	4	941	H 2, 12, 14, 15	MDD 13, 21, 23, 26
p_5	4	4	919	H 10, 11, 18, 20	MDD 14, 20, 24, 29
Sources (total)	21	22	4891		
Target (pooled)	5	6	1211	H 5, 6, 7, 8, 9	MDD 4, 5, 6, 7, 9, 34

Table 22: Grouping of source domain subjects into different subjects.

Our results are shown in Table 23. Overall, our `BaryFM` obtains the best performance in this benchmark, followed by unsupervised `WGF` (Montesuma et al., 2026a), and `TSBM` (Howard et al., 2025).

Method	CFM	Dual OT	$\mathcal{X} \times \mathcal{Y}$	Universal	Pool.	Rank
Source-Only	-	-	-	-	$91.25_{\pm 0.36}$	15
Discrete ✗	-	-	✗	✗	$91.30_{\pm 0.13}$	13
Discrete ✓	-	-	✓	✗	$91.30_{\pm 0.13}$	13
WGF ✗	-	-	✗	✗	$92.79_{\pm 0.41}$	2
WGF ✓	-	-	✓	✗	$91.77_{\pm 0.56}$	7
CW2B	-	-	✗	✗	$91.88_{\pm 0.19}$	4
WIN	-	-	✗	✗	$91.58_{\pm 0.08}$	11
NOT	-	-	✗	✗	$91.63_{\pm 0.13}$	10
U-NOT	-	-	✗	✗	$91.74_{\pm 0.22}$	8
TDSB	-	-	✗	✗	$91.74_{\pm 0.00}$	8
TSBM	-	-	✗	✗	$92.07_{\pm 0.22}$	3
NormFlow	-	-	✗	✗	$91.80_{\pm 0.05}$	5
GMM	-	-	✓	✗	$91.58_{\pm 0.08}$	11
NWB	-	-	✗	✗	$91.80_{\pm 0.17}$	5
NWBU	-	-	✗	✓	$91.14_{\pm 0.75}$	16
NeuralFP	✗	✓	✓	✓	$91.94_{\pm 0.17}$	3
$\text{BaryFM}_{\text{Emp}}$	✓	✗	✓	✓	$93.12_{\pm 0.27}$	1
$\text{BaryFM}_{\text{unsup}}$	✓	✓	✗	✓	$92.05_{\pm 0.48}$	3
BaryFM (no Universal)	✓	✓	✓	✗	$92.35_{\pm 0.99}$	2
BaryFM	✓	✓	✓	✓	$93.12_{\pm 0.59}$	1

Table 23: Mumtaz2016 results, along pooled performance, and rank.

G.3.8 PHYSIO

Similarly to BCI-CIV-2a, the Physio benchmark, proposed by Schalk et al. (2004), is a motor imagery dataset. This dataset constains a collection of 9837 EEGs, with $V = 64$ electrodes, collected over 109 subjects. Subjects were asked to imagine the movement of the left fist, right fist, both fist and both feet, resulting in a 4-class classification problem. We refer readers to Wang et al. (2025) for further information on preprocessing.

In this benchmark, most methods have marginal improvement over the baseline (e.g., `WGF` best with $65.54_{\pm 0.23}$ vs. Source-only 16th with $64.10_{\pm 0.08}$, where $\Delta = 1.44$). Our `BaryFM` achieves $65.00_{\pm 0.11}$, behind a few methods. Similarly to FACED (c.f. Table 20), this benchmark also has large inter-subject variability. We provide in Figure 21 a per-subject analysis on improvement over the source-only baseline. We see that methods improve performance on most subjects (e.g., 60% of subjects for `BaryFM` and `NWB`, 75% of subjects for `WGF` and `GMM`), but the overall performance improvement remains relatively narrow due to some subjects (e.g., Subject S98) having negative transfer. As mentioned, this seems to be a limitation of barycenter-base adaptation rather than the limitation of any specific method.

Method	CFM	Dual OT	$\mathcal{X} \times \mathcal{Y}$	Universal	S100	S101	S102	S103	S104	S105	S106	S107	S108	S109	S90	S91	S92	S93	S94	S95	S96	S97	S98	S99	Pool.	Rank
Source-Only	-	-	-	-	$60.19_{\pm 0.80}$	$92.22_{\pm 0.00}$	$62.22_{\pm 0.00}$	$78.89_{\pm 0.00}$	$62.45_{\pm 0.06}$	$79.63_{\pm 0.64}$	$55.56_{\pm 0.00}$	$47.78_{\pm 0.00}$	$66.67_{\pm 0.00}$	$47.78_{\pm 0.00}$	$73.33_{\pm 0.00}$	$66.67_{\pm 0.00}$	$62.87_{\pm 0.51}$	$54.44_{\pm 0.00}$	$64.44_{\pm 0.00}$	$60.00_{\pm 0.00}$	$82.22_{\pm 0.00}$	$64.44_{\pm 0.00}$	$60.00_{\pm 0.00}$	$39.63_{\pm 0.64}$	$64.10_{\pm 0.08}$	16
Discrete ✗	-	-	✗	✗	$63.43_{\pm 0.80}$	$92.59_{\pm 0.64}$	$67.41_{\pm 1.70}$	$77.78_{\pm 0.00}$	$64.75_{\pm 0.06}$	$82.96_{\pm 0.64}$	$54.81_{\pm 0.64}$	$48.52_{\pm 0.64}$	$68.52_{\pm 0.64}$	$49.63_{\pm 3.21}$	$77.04_{\pm 1.28}$	$64.81_{\pm 0.64}$	$65.20_{\pm 1.34}$	$53.70_{\pm 0.64}$	$65.56_{\pm 0.00}$	$60.00_{\pm 0.00}$	$82.59_{\pm 0.64}$	$64.07_{\pm 1.70}$	$56.30_{\pm 1.70}$	$38.89_{\pm 0.00}$	$64.95_{\pm 0.29}$	10
Discrete ✓	-	-	✓	✗	$64.35_{\pm 1.60}$	$92.59_{\pm 0.64}$	$67.78_{\pm 2.22}$	$78.15_{\pm 0.64}$	$65.13_{\pm 0.06}$	$83.33_{\pm 0.00}$	$54.81_{\pm 0.64}$	$48.15_{\pm 1.28}$	$68.52_{\pm 0.64}$	$49.63_{\pm 1.70}$	$75.93_{\pm 1.70}$	$65.56_{\pm 1.92}$	$65.20_{\pm 1.34}$	$53.33_{\pm 0.00}$	$65.56_{\pm 0.00}$	$60.37_{\pm 0.64}$	$81.48_{\pm 1.70}$	$64.81_{\pm 2.31}$	$55.56_{\pm 1.92}$	$38.89_{\pm 0.00}$	$64.97_{\pm 0.27}$	7
WGF ✗	-	-	✗	✗	$64.35_{\pm 2.12}$	$93.06_{\pm 0.00}$	$63.60_{\pm 0.67}$	$78.10_{\pm 1.31}$	$61.01_{\pm 1.19}$	$83.51_{\pm 1.09}$	$52.44_{\pm 0.63}$	$46.46_{\pm 1.89}$	$68.40_{\pm 0.66}$	$49.64_{\pm 1.26}$	$78.08_{\pm 3.42}$	$64.56_{\pm 1.06}$	$66.36_{\pm 0.51}$	$57.07_{\pm 0.64}$	$65.30_{\pm 0.63}$	$59.26_{\pm 1.71}$	$82.58_{\pm 1.06}$	$66.77_{\pm 1.94}$	$58.06_{\pm 1.24}$	$43.15_{\pm 1.96}$	$65.12_{\pm 0.40}$	3
WGF ✓	-	-	✓	✗	$65.74_{\pm 1.60}$	$92.96_{\pm 0.00}$	$67.95_{\pm 0.63}$	$78.85_{\pm 1.14}$	$64.94_{\pm 2.35}$	$82.10_{\pm 0.69}$	$54.58_{\pm 0.06}$	$49.99_{\pm 1.66}$	$69.05_{\pm 0.66}$	$50.77_{\pm 1.66}$	$76.94_{\pm 1.26}$	$66.47_{\pm 0.66}$	$65.19_{\pm 0.66}$	$54.91_{\pm 0.63}$	$65.71_{\pm 0.00}$	$60.03_{\pm 1.14}$	$81.90_{\pm 0.58}$	$66.11_{\pm 1.07}$	$57.29_{\pm 1.14}$	$39.98_{\pm 0.00}$	$65.54_{\pm 0.23}$	1
CW2B	-	-	✗	✗	$67.59_{\pm 0.80}$	$94.44_{\pm 0.00}$	$65.19_{\pm 0.64}$	$79.63_{\pm 0.64}$	$59.77_{\pm 0.00}$	$85.19_{\pm 0.64}$	$53.33_{\pm 1.11}$	$48.52_{\pm 0.64}$	$68.89_{\pm 0.00}$	$45.56_{\pm 0.00}$	$77.41_{\pm 0.64}$	$64.81_{\pm 0.64}$	$65.20_{\pm 0.51}$	$56.67_{\pm 0.00}$	$65.56_{\pm 0.00}$	$60.37_{\pm 0.64}$	$81.11_{\pm 1.11}$	$62.96_{\pm 0.64}$	$58.52_{\pm 0.64}$	$38.89_{\pm 0.00}$	$64.97_{\pm 0.14}$	7
WIN	-	-	✗	✗	$67.59_{\pm 0.80}$	$94.44_{\pm 0.00}$	$64.81_{\pm 0.64}$	$78.89_{\pm 0.00}$	$58.02_{\pm 0.00}$	$83.70_{\pm 1.28}$	$54.07_{\pm 1.28}$	$48.52_{\pm 1.70}$	$66.67_{\pm 0.00}$	$47.78_{\pm 1.92}$	$77.41_{\pm 0.64}$	$62.96_{\pm 0.64}$	$64.62_{\pm 1.01}$	$57.04_{\pm 0.64}$	$65.56_{\pm 0.00}$	$60.00_{\pm 0.00}$	$79.26_{\pm 0.64}$	$62.96_{\pm 2.31}$	$58.52_{\pm 0.64}$	$38.89_{\pm 0.00}$	$64.69_{\pm 0.03}$	15
NWB	-	-	✗	✗	$67.59_{\pm 0.80}$	$94.07_{\pm 0.64}$	$67.78_{\pm 0.00}$	$78.89_{\pm 1.11}$	$59.77_{\pm 0.00}$	$86.30_{\pm 0.64}$	$52.96_{\pm 0.64}$	$48.52_{\pm 0.64}$	$68.89_{\pm 0.00}$	$46.67_{\pm 0.00}$	$77.41_{\pm 0.64}$	$64.44_{\pm 0.00}$	$65.20_{\pm 0.51}$	$56.30_{\pm 0.64}$	$65.19_{\pm 0.64}$	$60.00_{\pm 0.00}$	$80.74_{\pm 0.64}$	$64.07_{\pm 0.64}$	$58.89_{\pm 0.00}$	$38.89_{\pm 0.00}$	$64.97_{\pm 0.28}$	7
LUNA-QOT	-	-	✗	✗	$67.13_{\pm 0.80}$	$94.07_{\pm 0.64}$	$64.44_{\pm 1.11}$	$78.63_{\pm 0.64}$	$59.77_{\pm 0.00}$	$85.56_{\pm 0.00}$	$52.96_{\pm 0.64}$	$48.52_{\pm 1.28}$	$68.15_{\pm 0.64}$	$46.67_{\pm 0.00}$	$77.78_{\pm 0.00}$	$64.44_{\pm 0.00}$	$65.20_{\pm 0.51}$	$56.30_{\pm 0.64}$	$65.19_{\pm 0.64}$	$60.00_{\pm 0.00}$	$81.48_{\pm 0.64}$	$62.96_{\pm 0.64}$	$58.52_{\pm 0.64}$	$38.89_{\pm 0.00}$	$64.84_{\pm 0.17}$	12
TDSB	-	-	✗	✗	$66.67_{\pm 0.80}$	$94.44_{\pm 0.00}$	$65.19_{\pm 0.64}$	$78.89_{\pm 0.64}$	$59.77_{\pm 0.00}$	$85.19_{\pm 0.00}$	$52.59_{\pm 0.64}$	$48.15_{\pm 0.64}$	$67.41_{\pm 0.64}$	$46.67_{\pm 0.00}$	$77.78_{\pm 0.00}$	$65.19_{\pm 1.28}$	$65.50_{\pm 0.51}$	$56.67_{\pm 0.00}$	$65.56_{\pm 0.00}$	$59.63_{\pm 0.64}$	$81.85_{\pm 0.64}$	$62.59_{\pm 0.64}$	$58.89_{\pm 0.00}$	$38.89_{\pm 0.00}$	$64.87_{\pm 0.17}$	11
TSBM	-	-	✗	✗	$65.74_{\pm 1.60}$	$94.44_{\pm 1.11}$	$66.67_{\pm 1.11}$	$81.48_{\pm 0.64}$	$58.02_{\pm 1.96}$	$81.85_{\pm 1.70}$	$53.70_{\pm 1.28}$	$48.89_{\pm 1.11}$	$66.67_{\pm 1.11}$	$47.78_{\pm 0.00}$	$77.78_{\pm 1.11}$	$63.33_{\pm 1.11}$	$64.91_{\pm 0.00}$	$56.30_{\pm 1.70}$	$65.56_{\pm 1.11}$	$59.26_{\pm 1.28}$	$80.37_{\pm 0.64}$	$65.19_{\pm 1.70}$	$57.78_{\pm 0.00}$	$39.26_{\pm 0.64}$	$64.78_{\pm 0.34}$	13
NormFlow	-	-	✗	✗	$68.06_{\pm 0.00}$	$94.07_{\pm 0.64}$	$64.44_{\pm 1.11}$	$80.00_{\pm 0.00}$	$59.77_{\pm 0.00}$	$85.56_{\pm 0.00}$	$52.96_{\pm 0.64}$	$47.78_{\pm 0.00}$	$68.15_{\pm 0.64}$	$45.56_{\pm 0.00}$	$77.41_{\pm 0.64}$	$63.70_{\pm 0.64}$	$64.91_{\pm 0.00}$	$56.30_{\pm 0.64}$	$65.56_{\pm 0.00}$	$60.00_{\pm 0.00}$	$80.74_{\pm 0.64}$	$62.59_{\pm 0.64}$	$58.89_{\pm 0.00}$	$38.89_{\pm 0.00}$	$64.74_{\pm 0.03}$	14
GMM	-	-	✓	✗	$62.04_{\pm 0.80}$	$93.33_{\pm 0.00}$	$66.67_{\pm 0.00}$	$78.89_{\pm 0.00}$	$67.82_{\pm 0.00}$	$86.67_{\pm 1.11}$	$53.33_{\pm 0.00}$	$48.89_{\pm 0.00}$	$67.78_{\pm 0.00}$	$50.74_{\pm 0.64}$	$75.56_{\pm 0.00}$	$65.56_{\pm 0.00}$	$64.62_{\pm 0.51}$	$56.67_{\pm 0.00}$	$68.15_{\pm 0.64}$	$61.85_{\pm 0.64}$	$83.33_{\pm 0.00}$	$62.22_{\pm 0.00}$	$54.44_{\pm 0.00}$	$41.85_{\pm 0.64}$	$65.54_{\pm 0.08}$	1
NWB	-	-	✗	✗	$67.13_{\pm 0.80}$	$93.70_{\pm 0.64}$	$65.93_{\pm 0.64}$	$79.63_{\pm 0.64}$	$59.77_{\pm 0.00}$	$84.81_{\pm 0.64}$	$53.33_{\pm 0.00}$	$48.52_{\pm 0.64}$	$68.89_{\pm 0.00}$	$47.04_{\pm 0.64}$	$78.52_{\pm 0.64}$	$64.81_{\pm 0.64}$	$64.91_{\pm 0.00}$	$57.41_{\pm 0.64}$	$64.81_{\pm 0.64}$	$59.26_{\pm 0.64}$	$81.11_{\pm 1.11}$	$63.70_{\pm 1.28}$	$58.89_{\pm 0.00}$	$38.89_{\pm 0.00}$	$65.04_{\pm 0.19}$	4
NWBU	-	-	✗	✓	$67.13_{\pm 0.80}$	$94.07_{\pm 0.64}$	$65.19_{\pm 0.64}$	$79.26_{\pm 1.28}$	$59.77_{\pm 0.00}$	$85.19_{\pm 0.64}$	$53.33_{\pm 0.00}$	$48.15_{\pm 0.64}$	$68.52_{\pm 0.64}$	$47.41_{\pm 0.64}$	$78.15_{\pm 0.64}$	$65.19_{\pm 0.64}$	$65.50_{\pm 0.51}$	$56.67_{\pm 1.11}$	$65.19_{\pm 0.64}$	$59.63_{\pm 0.64}$	$81.11_{\pm 0.00}$	$63.70_{\pm 1.28}$	$58.15_{\pm 0.64}$	$38.89_{\pm 0.00}$	$65.00_{\pm 0.06}$	6
NeuralFP	✗	✓	✓	✓	$68.06_{\pm 0.00}$	$96.44_{\pm 0.49}$	$66.60_{\pm 0.00}$	$78.85_{\pm 0.00}$	$60.06_{\pm 1.05}$	$86.86_{\pm 0.91}$	$49.85_{\pm 0.00}$	$46.90_{\pm 0.49}$	$68.41_{\pm 0.51}$	$48.90_{\pm 2.39}$	$72.79_{\pm 1.07}$	$64.79_{\pm 0.51}$	$65.40_{\pm 0.83}$	$57.79_{\pm 0.89}$	$66.75_{\pm 0.00}$	$57.38_{\pm 1.07}$	$81.67_{\pm 0.56}$	$66.70_{\pm 0.00}$	$56.63_{\pm 0.00}$	$42.02_{\pm 0.00}$	$65.15_{\pm 0.09}$	3
BaryFM$_{\text{Emp}}$	✓	✓	✓	✓	$70.83_{\pm 1.39}$	$95.19_{\pm 0.64}$	$65.19_{\pm 0.64}$	$80.74_{\pm 0.64}$	$59.00_{\pm 0.06}$	$86.67_{\pm 0.00}$	$48.52_{\pm 0.64}$	$49.26_{\pm 0.64}$	$67.78_{\pm 0.00}$	$48.89_{\pm 1.92}$	$77.41_{\pm 0.64}$	$65.56_{\pm 0.00}$	$64.04_{\pm 0.00}$	$56.67_{\pm 0.64}$	$66.30_{\pm 0.64}$	$78.15_{\pm 0.64}$	$64.07_{\pm 0.64}$	$64.07_{\pm 0.64}$	$57.78_{\pm 0.00}$	$38.89_{\pm 0.00}$	$64.98_{\pm 0.21}$	6
BaryFM$_{\text{Mmp}}$	✓	✓	✓	✓	$69.91_{\pm 0.80}$	$94.81_{\pm 0.64}$	$65.93_{\pm 1.28}$	$78.15_{\pm 0.64}$	$99.39_{\pm 1.33}$	$86.67_{\pm 0.00}$	$49.26_{\pm 0.64}$	$48.89_{\pm 0.00}$	$68.89_{\pm 0.00}$	$47.04_{\pm 0.64}$	$75.19_{\pm 0.64}$	$65.19_{\pm 0.64}$	$64.62_{\pm 0.51}$	$58.89_{\pm 0.00}$	$66.67_{\pm 0.00}$	$58.15_{\pm 0.64}$	$81.11_{\pm 0.00}$	$64.44_{\pm 0.00}$	$56.67_{\pm 0.00}$	$41.11_{\pm 0.00}$	$65.00_{\pm 0.17}$	5
BaryFM (no Universal)	✓	✓	✓	✗	$68.52_{\pm 1.60}$	$94.81_{\pm 0.64}$	$66.30_{\pm 0.64}$	$78.15_{\pm 0.64}$	$99.39_{\pm 0.06}$	$86.67_{\pm 0.00}$	$48.52_{\pm 0.64}$	$48.89_{\pm 0.00}$	$68.89_{\pm 0.00}$	$47.41_{\pm 0.64}$	$74.44_{\pm 0.00}$	$64.44_{\pm 1.11}$	$65.50_{\pm 0.51}$	$58.52_{\pm 0.64}$	$66.30_{\pm 0.64}$	$57.78_{\pm 0.00}$	$81.11_{\pm 0.00}$	$65.19_{\pm 0.64}$	$56.67_{\pm 0.00}$	$40.74_{\pm 0.64}$	$64.89_{\pm 0.17}$	10
BaryFM	✓	✓	✓	✓	$69.44_{\pm 0.00}$	$94.44_{\pm 0.00}$	$65.93_{\pm 0.64}$	$78.52_{\pm 0.64}$	$99.17_{\pm 0.00}$	$86.67_{\pm 0.00}$	$48.89_{\pm 0.00}$	$48.89_{\pm 0.00}$	$68.89_{\pm 0.00}$	$47.41_{\pm 0.64}$	$75.19_{\pm 1.28}$	$64.44_{\pm 1.11}$	$64.53_{\pm 0.51}$	$58.89_{\pm 0.00}$	$66.67_{\pm 0.00}$	$58.15_{\pm 0.64}$	$81.11_{\pm 0.00}$	$65.19_{\pm 0.64}$	$57.04_{\pm 0.64}$	$41.11_{\pm 0.00}$	$65.00_{\pm 0.11}$	5

Table 24: Detailed results on the Physio benchmark.

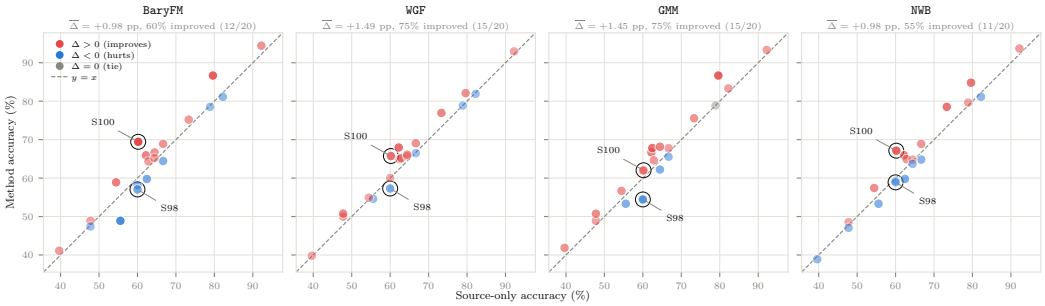


Figure 21: Source-only accuracy vs. method accuracy for `BaryFM` (5th), `WGF` (1st), `GMM` (1st) and `NWB` (4th). The bisecting line defines the no-improvement region. Points under this line (e.g., S98) denote subjects for which performance is worse than the source-only. Points above it denote subjects for which performance improved over that baseline.

G.3.9 SHU-MI

SHU-MI is a benchmark proposed by Ma et al. (2022) concerned with motor imagery (like Physio and BCI-CIV-2a), with 2 classes, where subjects were asked to imagine left and right hand grasping. The EEGs in this benchmark were collected with $V = 32$ electrodes, and 25 subjects over 5 separate sessions, with 100 trials per session. There are, in total, 11988 samples. Subjects are divided into $1, \cdots, 15$ used for training, $16, \cdots, 20$ for checkpoint selection, and $21, \cdots, 25$ for test. We refer readers to Wang et al. (2025) for further details on pre-processing. We report our results in Table 25.

Method	CFM	Dual OT	$\mathcal{X} \times \mathcal{Y}$	Universal	S21	S22	S23	S24	S25	Pool.	Rank
Source-Only	-	-	-	-	$76.97_{\pm 0.21}$	$67.50_{\pm 0.42}$	$67.97_{\pm 1.46}$	$52.86_{\pm 1.43}$	$46.46_{\pm 1.19}$	$62.68_{\pm 0.41}$	14
Discrete ✗	-	-	✗	✗	$57.95_{\pm 6.86}$	$61.53_{\pm 3.34}$	$61.52_{\pm 7.94}$	$48.36_{\pm 3.81}$	$49.06_{\pm 2.61}$	$55.89_{\pm 3.26}$	15
Discrete ✓	-	-	✓	✗	$53.32_{\pm 4.50}$	$59.10_{\pm 4.90}$	$58.57_{\pm 10.66}$	$49.05_{\pm 3.50}$	$48.27_{\pm 1.63}$	$53.80_{\pm 4.03}$	16
WGF ✗	-	-	✗	✗	$76.69_{\pm 1.04}$	$68.12_{\pm 2.29}$	$67.35_{\pm 0.63}$	$53.47_{\pm 0.95}$	$47.19_{\pm 1.15}$	$62.87_{\pm 0.90}$	13
WGF ✓	-	-	✓	✗	$78.08_{\pm 0.24}$	$68.75_{\pm 0.62}$	$69.14_{\pm 0.00}$	$53.70_{\pm 0.95}$	$47.19_{\pm 0.57}$	$63.70_{\pm 0.11}$	1
CW2B	-	-	✗	✗	$76.90_{\pm 0.73}$	$67.43_{\pm 0.24}$	$69.07_{\pm 1.32}$	$53.32_{\pm 0.79}$	$46.90_{\pm 0.98}$	$63.05_{\pm 0.20}$	7
WIN	-	-	✗	✗	$76.90_{\pm 0.24}$	$67.01_{\pm 0.12}$	$68.52_{\pm 0.21}$	$53.85_{\pm 1.15}$	$46.97_{\pm 1.32}$	$62.96_{\pm 0.32}$	10
NOT	-	-	✗	✗	$76.35_{\pm 0.21}$	$67.71_{\pm 1.65}$	$69.07_{\pm 0.86}$	$53.55_{\pm 1.21}$	$46.68_{\pm 0.12}$	$62.99_{\pm 0.52}$	9
U-NOT	-	-	✗	✗	$76.63_{\pm 0.48}$	$67.57_{\pm 1.22}$	$69.27_{\pm 0.83}$	$52.71_{\pm 1.06}$	$46.75_{\pm 0.37}$	$62.92_{\pm 0.21}$	12
TDSB	-	-	✗	✗	$76.14_{\pm 0.21}$	$67.92_{\pm 0.55}$	$68.66_{\pm 1.34}$	$53.62_{\pm 0.87}$	$47.84_{\pm 0.37}$	$63.14_{\pm 0.21}$	4
TSBM	-	-	✗	✗	$76.69_{\pm 0.84}$	$67.36_{\pm 1.42}$	$69.14_{\pm 1.25}$	$53.09_{\pm 1.27}$	$47.40_{\pm 1.21}$	$63.06_{\pm 0.70}$	6
NormFlow	-	-	✗	✗	$76.35_{\pm 0.83}$	$67.64_{\pm 0.98}$	$68.79_{\pm 0.43}$	$53.32_{\pm 0.83}$	$47.04_{\pm 1.44}$	$62.95_{\pm 0.26}$	11
GMM	-	-	✓	✗	$78.01_{\pm 1.30}$	$68.54_{\pm 0.00}$	$69.41_{\pm 0.52}$	$53.55_{\pm 0.40}$	$46.25_{\pm 1.54}$	$63.49_{\pm 0.46}$	3
NWB	-	-	✗	✗	$76.63_{\pm 1.22}$	$67.99_{\pm 0.32}$	$69.14_{\pm 0.82}$	$53.39_{\pm 1.08}$	$46.97_{\pm 0.94}$	$63.14_{\pm 0.60}$	4
NWBU	-	-	✗	✓	$77.04_{\pm 0.84}$	$67.29_{\pm 0.72}$	$68.79_{\pm 0.12}$	$53.39_{\pm 0.58}$	$47.11_{\pm 0.54}$	$63.05_{\pm 0.12}$	7
NeuralFP	✗	✓	✓	✓	$78.15_{\pm 0.84}$	$68.54_{\pm 0.42}$	$68.86_{\pm 0.43}$	$54.31_{\pm 1.17}$	$46.68_{\pm 0.45}$	$63.63_{\pm 0.49}$	2
$\text{BaryFM}_{\text{Emp}}$	✓	✗	✓	✓	$77.59_{\pm 0.36}$	$68.33_{\pm 0.72}$	$68.45_{\pm 1.34}$	$54.08_{\pm 0.35}$	$48.05_{\pm 1.32}$	$63.61_{\pm 0.66}$	2
$\text{BaryFM}_{\text{unsup}}$	✓	✓	✗	✓	$78.56_{\pm 1.14}$	$66.67_{\pm 0.21}$	$69.07_{\pm 1.32}$	$54.54_{\pm 1.40}$	$47.69_{\pm 0.54}$	$63.61_{\pm 0.30}$	2
BaryFM (no Universal)	✓	✓	✓	✗	$78.84_{\pm 0.75}$	$66.81_{\pm 0.12}$	$69.55_{\pm 0.62}$	$53.47_{\pm 0.48}$	$47.04_{\pm 0.33}$	$63.47_{\pm 0.28}$	3
BaryFM	✓	✓	✓	✓	$78.01_{\pm 0.21}$	$68.96_{\pm 0.72}$	$69.00_{\pm 0.12}$	$53.39_{\pm 0.26}$	$47.47_{\pm 1.09}$	$63.70_{\pm 0.15}$	2

Table 25: Per-domain SHU-MI results, along pooled performance, and rank.

In this benchmark, most empirical solvers fail (e.g., the discrete solver achieves $55.89_{\pm 3.26}$ vs. the source-only baseline at $62.68_{\pm 0.41}$). One exception is `WGF`, who achieves the best performance of $63.70_{\pm 0.11}$. Conversely, all neural solvers improve over the source-only baseline. Our `BaryFM` achieves 2nd place, being statistically indistinguishable from `WGF` ($63.70_{\pm 0.15}$ ours, $63.70_{\pm 0.11}$ theirs).

G.3.10 TENNESSEE EASTMAN PROCESS

The Tennessee Eastman Process (TEP) is a large scale, widely used process simulator in the chemical engineering community, first introduced by Downs & Vogel (1993). Here, we use its domain adaptation instantiation introduced by Montesuma et al. (2026b), based on the simulations of Reinartz et al. (2021). Essentially, it consists on multi-variate time series with $V = 34$ variates of different physical quantities (e.g., pressure, temperature, concentration). For each simulation, there is one associated label. These labels describe the state of the system (one of 28 possible faults, plus a healthy state). The goal is *fault diagnosis*, meaning, from the time-series, to determine the state of the system. This benchmark has 17289 samples, more or less uniformly distributed among 6 modes of operation. We refer readers to Montesuma et al. (2026b) for further information on this benchmark.

Method	CFM	Dual OT	$\mathcal{X} \times \mathcal{Y}$	Universal	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Avg.	Rank
Source-Only	-	-	-	-	$82.40_{\pm 0.07}$	$67.03_{\pm 1.03}$	$88.52_{\pm 0.07}$	$78.30_{\pm 0.15}$	$73.51_{\pm 0.13}$	$85.34_{\pm 0.07}$	$79.18_{\pm 0.18}$	16
Discrete ✗	-	-	✗	✗	$89.66_{\pm 0.75}$	$73.53_{\pm 1.35}$	$84.96_{\pm 1.95}$	$88.34_{\pm 0.75}$	$84.91_{\pm 0.89}$	$85.81_{\pm 2.34}$	$84.54_{\pm 1.47}$	6
Discrete ✓	-	-	✓	✗	$89.97_{\pm 1.25}$	$73.11_{\pm 1.54}$	$88.48_{\pm 1.76}$	$88.52_{\pm 0.83}$	$84.46_{\pm 0.84}$	$86.85_{\pm 1.07}$	$85.23_{\pm 1.26}$	5
WGF ✗	-	-	✗	✗	$91.52_{\pm 1.32}$	$73.78_{\pm 1.25}$	$87.44_{\pm 1.17}$	$89.39_{\pm 0.62}$	$84.88_{\pm 1.76}$	$86.09_{\pm 2.23}$	$85.52_{\pm 1.48}$	3
WGF ✓	-	-	✓	✗	$92.34_{\pm 1.05}$	$76.10_{\pm 1.57}$	$89.79_{\pm 0.90}$	$89.32_{\pm 1.04}$	$85.92_{\pm 1.36}$	$87.75_{\pm 1.52}$	$86.87_{\pm 1.27}$	2
CW2B	-	-	✗	✗	$88.76_{\pm 1.51}$	$68.37_{\pm 0.85}$	$85.34_{\pm 1.48}$	$88.17_{\pm 1.00}$	$81.44_{\pm 2.71}$	$85.92_{\pm 0.95}$	$83.00_{\pm 1.55}$	13
WIN	-	-	✗	✗	$85.10_{\pm 1.50}$	$68.05_{\pm 1.59}$	$89.27_{\pm 1.44}$	$86.84_{\pm 2.81}$	$79.29_{\pm 2.10}$	$86.92_{\pm 1.91}$	$82.58_{\pm 1.95}$	14
NOT	-	-	✗	✗	$90.28_{\pm 2.52}$	$70.44_{\pm 2.95}$	$85.27_{\pm 1.79}$	$88.59_{\pm 1.65}$	$83.07_{\pm 1.83}$	$82.43_{\pm 1.45}$	$83.35_{\pm 2.10}$	12
U-NOT	-	-	✗	✗	$89.07_{\pm 1.49}$	$71.88_{\pm 1.95}$	$85.34_{\pm 1.80}$	$87.85_{\pm 0.83}$	$83.90_{\pm 2.43}$	$82.64_{\pm 2.50}$	$83.45_{\pm 1.92}$	10
TDSB	-	-	✗	✗	$88.28_{\pm 2.12}$	$70.19_{\pm 0.94}$	$87.48_{\pm 1.71}$	$87.75_{\pm 1.18}$	$83.70_{\pm 1.43}$	$85.16_{\pm 1.31}$	$83.76_{\pm 1.50}$	9
TSBM	-	-	✗	✗	$90.41_{\pm 1.45}$	$69.81_{\pm 0.86}$	$88.48_{\pm 1.18}$	$87.33_{\pm 0.75}$	$84.01_{\pm 1.33}$	$86.47_{\pm 1.64}$	$84.42_{\pm 1.24}$	7
NormFlow	-	-	✗	✗	$88.62_{\pm 0.81}$	$69.77_{\pm 1.80}$	$87.89_{\pm 1.42}$	$87.96_{\pm 1.13}$	$82.86_{\pm 1.53}$	$86.43_{\pm 0.83}$	$83.92_{\pm 1.30}$	8
GMM	-	-	✓	✗	$89.33_{\pm 0.48}$	$72.89_{\pm 1.49}$	$88.81_{\pm 0.28}$	$88.34_{\pm 0.72}$	$85.53_{\pm 0.79}$	$87.14_{\pm 0.65}$	$85.34_{\pm 0.63}$	4
NWB	-	-	✗	✗	$87.62_{\pm 0.95}$	$67.45_{\pm 1.34}$	$88.48_{\pm 1.18}$	$88.17_{\pm 0.86}$	$80.89_{\pm 2.90}$	$87.57_{\pm 1.77}$	$83.36_{\pm 1.65}$	11
NWBU	-	-	✗	✓	$88.79_{\pm 0.49}$	$58.21_{\pm 2.02}$	$84.37_{\pm 1.66}$	$85.79_{\pm 1.99}$	$84.42_{\pm 1.35}$	$81.84_{\pm 0.85}$	$80.57_{\pm 1.50}$	15
NeuralFP	✗	✓	✓	✓	$92.38_{\pm 1.06}$	$76.56_{\pm 1.79}$	$90.10_{\pm 0.98}$	$89.56_{\pm 1.02}$	$85.92_{\pm 1.31}$	$87.88_{\pm 1.56}$	$87.07_{\pm 1.32}$	1
BaryFM$_{\text{Emp}}$	✓	✗	✓	✓	$92.41_{\pm 1.03}$	$76.91_{\pm 1.44}$	$90.13_{\pm 0.92}$	$89.56_{\pm 0.74}$	$86.40_{\pm 1.32}$	$88.12_{\pm 1.28}$	$87.26_{\pm 1.15}$	1
BaryFM$_{\text{unsup}}$	✓	✓	✗	✓	$92.45_{\pm 0.91}$	$77.08_{\pm 1.67}$	$90.31_{\pm 0.70}$	$89.91_{\pm 0.66}$	$87.10_{\pm 1.52}$	$87.09_{\pm 1.31}$	$87.32_{\pm 1.20}$	1
BaryFM (no Universal)	✓	✓	✓	✗	$92.34_{\pm 1.03}$	$76.77_{\pm 1.48}$	$89.82_{\pm 0.91}$	$89.63_{\pm 0.87}$	$86.19_{\pm 1.53}$	$88.06_{\pm 1.33}$	$87.14_{\pm 1.22}$	1
BaryFM	✓	✓	✓	✓	$92.69_{\pm 0.92}$	$76.77_{\pm 2.14}$	$89.96_{\pm 1.04}$	$89.77_{\pm 0.82}$	$86.82_{\pm 1.50}$	$87.81_{\pm 1.48}$	$87.30_{\pm 1.39}$	1

Table 26: Per-mode TEP results, along average over modes, and rank.

We adopt the reporting style of Montesuma et al. (2026b), who pre-defined 5 folds on each domain. Therefore, we fix seed 42, and run each method using the embeddings of all source domain data, with an additional of 80% of target domain points (folds $f_1, \cdots, f_4$) with fold $f_5 \in \{1, \cdots, 5\}$ kept for testing. The testing data is never seen neither through the training of the backbone, nor through the adaptation stage. We report the mean over folds, $\pm$ the standard deviation, in Table 26.

From Table 26, our `BaryFM` performs best in all variants, against `WGF` (2nd best). For instance, our best performing variant, `BaryFM`$_{\text{unsup}}$ achieves $87.32_{\pm 1.15}$ against `WGF`'s $86.87_{\pm 1.37}$. Furthermore, all methods achieve a statistically significant improvement over the source-only baseline.

H Domain Generalization

H.1 Data Augmentation for Domain Generalization

Our goal is training a model on measures from the Wasserstein simplex, $\mathcal{W}(p_{1:K})$. The underlying idea is that, despite the vertices $p_1, \cdots, p_K$, the barycenters $\text{Bar}(\lambda, p_{1:K})$ will help enrich the source domain data, improving generalization. We therefore compose an augmented sample pool $\{\{\tilde{x}_{\ell,i}, \tilde{y}_{\ell,i}\}_{i=1}^m\}_{\ell=1}^N$, where $(\tilde{x}_{\ell,i}, \tilde{y}_{\ell,i}) \sim q_{\lambda_\ell}$ and $\lambda_\ell \sim \text{Dir}(\alpha)$. We then learn a model minimizing the risk,

$$\theta^\star = \arg\min_{\theta\in\Theta} \frac{1-\rho}{n}\sum_{i=1}^{n} \ell(y_{k,i}, f_\theta(x_{k,i})) + \frac{\rho}{mN}\sum_{\ell=1}^{N}\sum_{i=1}^{m} \ell(\tilde{y}_{\ell,i}, f_\theta(\tilde{x}_{\ell,i})),$$

where f_θ is a neural network, such as a ResNet 50 (He et al., 2016), θ are its weights, and $\rho \in [0, 1]$ is a importance factor. Note that this application requires 3 properties of barycenter solvers: (i) amortization over $z \sim \text{Bar}(\lambda, p_{1:K})$, (ii) *universality*, i.e., amortization over $\lambda \in \Delta_K$ and (iii) joint modeling over $z = (x, y)$, all of which, *to the best of our knowledge*, only our `BaryFM` suffices. We call this strategy $\text{BaryFM}_{\text{Random}}$. Alternatively, we build the augmented sampling pool with $N = K$ and $\lambda_\ell = e_\ell, \ell = 1, \cdots, K$ being the vertices of the simplex. This roughly corresponds to sampling from the marginal measures.

As the main point of comparison, we adopt MixUp (Zhang et al., 2017) and Domain MixUp (Wang et al., 2020) which correspond to interpolating $p_{1:K}$ linearly. First, MixUp corresponds to sampling $\{(x_1, y_1), (x_2, y_2)\} \sim \sum_k (n_k/n) p_k \otimes \sum_k (n_k/n) p_k$, where n_k is the number of samples from measure k, and $\lambda \sim \text{Beta}(\alpha, \alpha)$ then generating $\tilde{x} = (1-\lambda)x_1 + \lambda x_2$, $\tilde{y} = (1-\lambda)y_1 + \lambda y_2$. In contrast, Domain MixUp samples $(x_1, y_1), \cdots, (x_K, y_K) \sim p_1 \otimes \cdots \otimes p_K$ then mixes using,

$$\tilde{x} = \sum_{k=1}^{K} \lambda_k x_k, \text{ and } \tilde{y} = \sum_{k=1}^{K} \lambda_k y_k. \tag{36}$$

In practice, for better capturing the semantics of complex objects, we perform interpolation in the latent space of an autoencoder using either version of BaryFM or MixUp. We use a VAE (Kingma & Welling, 2013; Rombach et al., 2022) and PCA for EEG data.

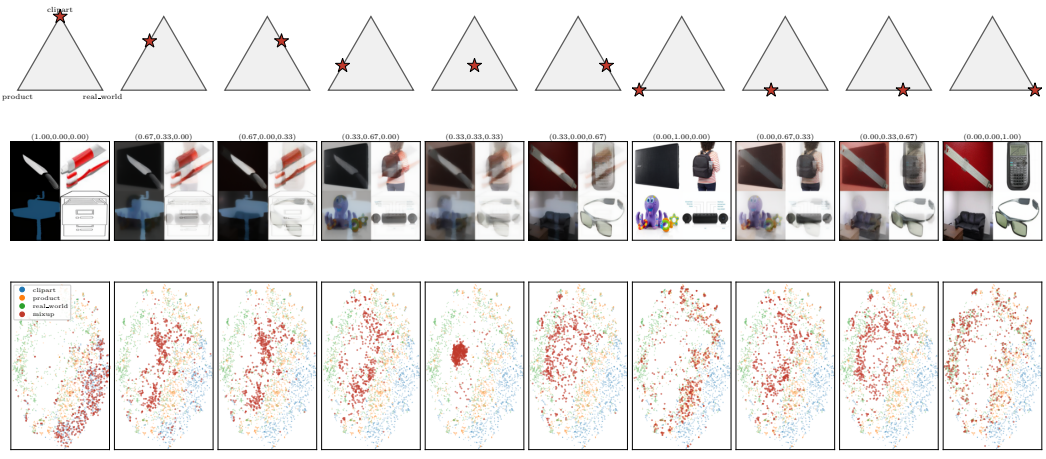


Figure 22: Visualization of the linear mixing of multiple domain data via Domain MixUp of Wang et al. (2020). In the first row, we show the simplex, and the picked mixing weights λ. In the second row, we show the mixed samples obtained by equation 36, and decoded back into image space. In the third row, we show the t-SNE visualization (Maaten & Hinton, 2008) of the VAE embeddings of the mixed samples.

We show a comparison in the blending of domains in Figures 22 (Domain MixUp) and 23 (`BaryFM`, ours). As previously mentioned, these methods blend samples in the latent space of an autoencoder (a VAE for images, and PCA for EEG). As is natural with MixUp (Zhang et al., 2017), linearly

blending samples induces a ghosting effect. While doing that in the latent space of a VAE alleviates that effect to an extent, these artifacts are still seen in the images (c.f., non-vertices of the simplex). From the t-SNE, we see that the generated samples span a different region of the latent space, indicating that, through Domain MixUp, one covers new regions, enriching the source data.

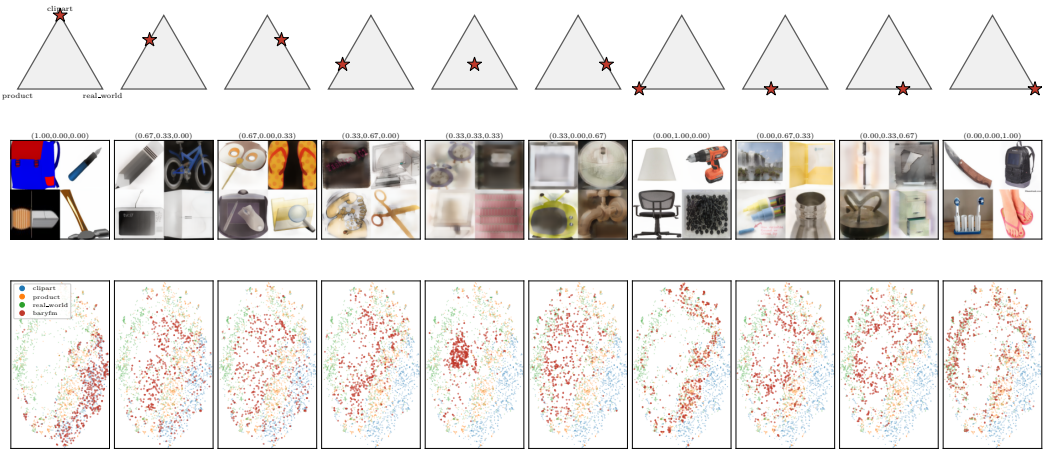


Figure 23: Visualization of nonlinear mixing of multiple domain data via `BaryFM` (ours). In the first row, we show the simplex, and picked mixing weights λ. In the second row, we show the decoded images of samples from $\text{Bar}(\lambda, p_{1:K})$. In the third row, we show a t-SNE visualization (Maaten & Hinton, 2008) of the VAE embeddings of the samples from the barycenter.

In comparison with Domain MixUp, our proposed `BaryFM` interpolated data exhibit fuzziness, rather than ghosting artifacts. Looking at the t-SNE embeddings in the third row of Figure 23, we can also see that the interpolated samples span a wider region of the latent space. For instance, for uniform weights, $\lambda = (0.33, 0.33, 0.33)$, our `BaryFM` concentrates less than linear mixing. These results help explaining the different performance of `BaryFM` and Domain MixUp, as well as the gain in performance our method has over Domain MixUp (Wang et al., 2020) and MixUp (Zhang et al., 2017) on the Office Home dataset.

H.2 Experimental Details

As shown in Figure 8, we perform data augmentation to complement source domain data in the latent space of an autoencoder. For images, we use a VAE (Kingma & Welling, 2013; Rombach et al., 2022), and for EEGs, we use a linear autoencoder (that is, PCA). The overall pipeline consists of pre-encoding the source domain data, then training a `BaryFM` model over the latent codes. The main advantage here consists of exploiting the semantic information encoded in such a latent space.



(a) Office 31 (b) Office Home (c) Domain Net

Figure 24: The effect of autoencoding images in the Office31, Office Home, and Domain Net, using the VAE of Rombach et al. (2022). On each sub-figure, we show in the left the original images in the dataset x, and the reconstructed images from the latent codes, $\hat{x} = \text{Dec}(\text{Enc}(x))$.

Variational Autoencoder and `BaryFM` architecture. We use the VAE of Rombach et al. (2022) (`sd-vae-ft-mse`), which downsamples images by a factor of 8. In parallel, following standard practice (Peng et al., 2019), each image in the Office 31, Office Home, and DomainNet is resized to

$3 \times 256 \times 256$. This means that the VAE produces latent codes of shape $4 \times 32 \times 32$, that is, the dimensionality of inputs is $\dim_x = 4096$. We show the effect of this round trip in Figure 24.

As a result of these choices, the latent codes in the image experiments in domain generalization have an underlying spatial structure. We therefore adapt the MLP architecture we have used in domain adaptation, fairness and Bayesian posterior aggregation to take this structure into account. An overview of our architecture is shown in Figure 25. Briefly, x is fed into the U-Net, which outputs the $x-$component v_x of v_θ. We use the bottleneck of the U-Net, after an average pooling operation, to produce the label component of the velocity field, $v_y \in \mathbb{R}^C$, and the dual OT head $f_{\epsilon,\psi}(z, k, \lambda)$. The conditioning variables y, k, λ and t are embedded using 2-layer MLPs ($y \in \Delta_C$ and $\lambda \in \Delta_K$), a lookup table $k \in \{1, \cdots, K\}$, and sinusoidal positional encoding ($t \in [0, 1]$). These produce embedding vectors E_t, E_k, E_λ, E_y which are summed to produce a conditional embedding $E_{\text{cond}} \in \mathbb{R}^{\dim_{\text{cond}}}$, with $\dim_{\text{cond}} = 256$. The embedding E_{cond} modulates the residual blocks in the UNet through a feature-wise affine transformation (Perez et al., 2018). Furthermore, following Ho et al. (2020), the residual blocks at resolutions 16×16, 8×8, 4×4, and the bottleneck, are followed by multi-head and self-attention (Vaswani et al., 2017) over spatial positions of the feature maps. We summarize this architecture in Table 27.

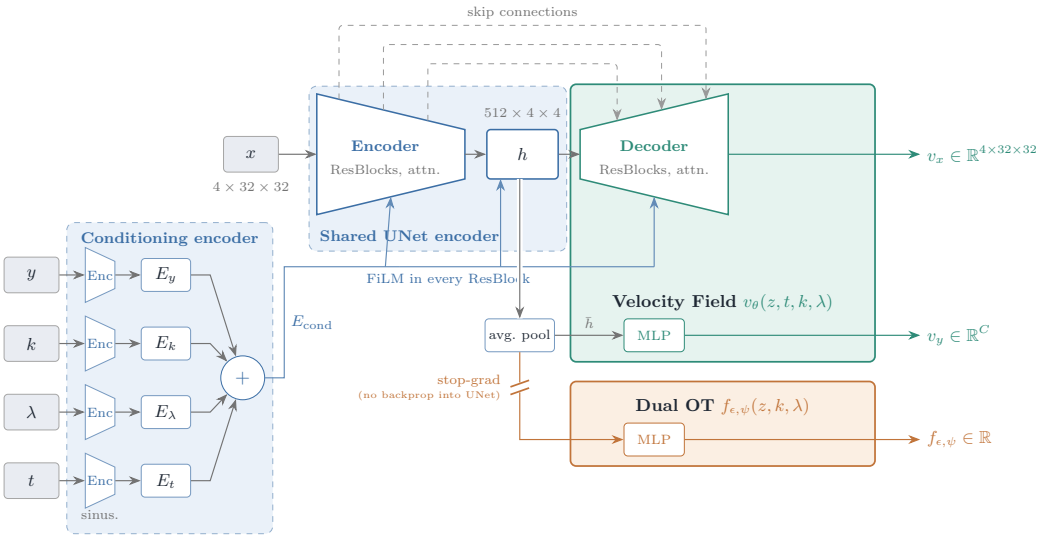


Figure 25: Architecture of our `BaryFM` used in the domain generalization experiments.

Dataset	Latent shape	K	C	Channels	Attention res.	# Params
Office 31	$4 \times 32 \times 32$	2	31	$(128, 256, 512, 512)$	$\{16, 8, 4\}$	117.23M
Office-Home	$4 \times 32 \times 32$	3	65	$(128, 256, 512, 512)$	$\{16, 8, 4\}$	117.25M
DomainNet	$4 \times 32 \times 32$	5	345	$(128, 256, 512, 512)$	$\{16, 8, 4\}$	117.39M

Table 27: Architecture of the UNet used for generating the augmented pools in domain generalization. K is the number of source domains in each leave-one-domain-out fold, and C the number of classes. All networks use 2 residual blocks per level, 4 attention heads, and a conditioning dimension $\dim_c = 256$. The last column gives the total number of trainable parameters, including the label and dual OT heads.

Principal Component Analysis. For BCI-CIV-2a and SEED-VIG, we fit a PCA model to the standardized EEGs in the source subjects of each benchmark. In these benchmarks, input data is $x \in \mathbb{R}^{V \times T}$, where V is the number of electrodes and T is the number of timesteps. We flatten these EEGs into $x \in \mathbb{R}^{VT}$ before learning PCA. This makes a raw data space of $22 \times 800 = 17600$ dimensions for BCI-CIV-2a, and $17 \times 1600 = 27200$ dimensions for SEED-VIG. In this context, we measure several metrics to select a reasonable dimensionality for the latent space. In the following we denote $\hat{x} = \text{Dec}(\text{Enc}(x))$,

1. variate-wise BW2-UVP% between x and $\hat{x}$. This metric reduces each EEG to a vector $\mu \in \mathbb{R}^V$ of variate-wise means, and compute the BW2-UVP% metric over it. It measures how much the latent space distorts the distributional information of the variates.
2. Mean Squared Error (MSE) of log spectral features, denoted S for the Welch spectral density of x (resp. $\hat{S}$ for $\hat{x}$).
3. MSE between x and $\hat{x}$.
4. Downstream performance. For BCI-IV-2a, we report the per-subject, intra-subject classification accuracy. For SEED-VIG, we report the per-subject, intra-subject Pearson correlation in %. This metric probes how well the latent space preserves the task structure (e.g., class separation for classification).

We show a comparison of these metrics as a function of number of principal components in Figure 26. Overall, we can see that the metrics improve with more principal components, and, especially, around dim $= 2048$ the downstream task performance matches the raw signals. We, therefore, use dim $= 2048$ for BCI-CIV-2a, and dim $= 4096$ for SEED-VIG.

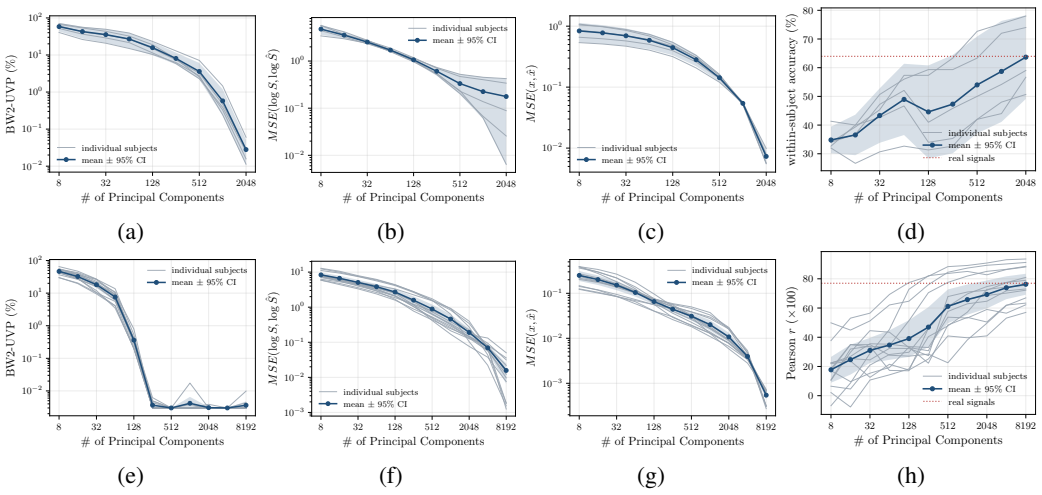


Figure 26: Channel-wise BW2-UVP%, log-spectral distance, MSE and downstream performance as a function of number of principal components for BCI-IV-2a (a – d) and SEED-VIG (e – h).

Hyper-parameter	Office 31	Office-Home	DomainNet	BCI-CIV-2a	SEED-VIG
Latent space	VAE, $4 \times 32 \times 32$	VAE, $4 \times 32 \times 32$	VAE, $4 \times 32 \times 32$	PCA, 2048	PCA, 4096
Architecture	UNet	UNet	UNet	MLP	MLP
Optimizer	Adam	Adam	Adam	AdamW	AdamW
Learning rate	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
Weight decay	–	–	–	0.05	0.05
Learning rate schedule	Cosine, $\text{LR}_{\text{min}} = 10^{-6}$	Cosine, $\text{LR}_{\text{min}} = 10^{-6}$	Cosine, $\text{LR}_{\text{min}} = 10^{-6}$	Cosine, $\text{LR}_{\text{min}} = 10^{-6}$	Cosine, $\text{LR}_{\text{min}} = 10^{-6}$
Training iterations	30 000	30 000	30 000	50 000	50 000
Batch size	128	128	128	512	256
Gradient clipping	None	None	None	1.0	1.0
EMA decay	0.9999	0.9999	0.9999	0.9999	0.9999
Entropic regularization	10^{-2}	10^{-2}	10^{-2}	10^{-3}	10^{-3}
Dual steps per iteration	1	1	1	1	1
Fixed-point iterations	1	1	1	1	1
ODE solver	Euler	Euler	Euler	Midpoint	Midpoint
n_{gen}	100,000	100,000	100,000	5,000	26,550
ODE steps T	100	100	100	100	100
Seeds (DG loop)	`0--4`	`0--4`	`0--2`	`0--9`	`0--9`

Table 28: Training and inference hyper-parameters used in our domain generalization experiments.

Hyper-Parameters. We summarize the overall training and inference hyper-parameters in Table 28 which are similar to the hyper-parameters reported for domain adaptation (Tables 11 and 12).

H.3 Detailed Results

Similarly do domain adaptation, we detail our obtained results per domain for each benchmark. Note that results are different from domain adaptation (and sometimes, inferior) due to different factors. First, the WBT pipeline runs over embeddings, and can therefore be seen as linear probing. In contrast, our domain generalization results fine-tune the complete network from the synthetic generated data. Second, in domain generalization one does not have access to the target domain data, which makes the task itself harder than domain adaptation.

Method	Amazon	dSLR	Webcam	Avg.	Rank
ERM (ResNet 50)	$60.39_{\pm 0.74}$	$98.40_{\pm 0.55}$	$94.34_{\pm 1.33}$	$84.38_{\pm 0.53}$	5
MixUp	$60.74_{\pm 1.08}$	$99.00_{\pm 0.00}$	$97.23_{\pm 1.14}$	$85.66_{\pm 0.69}$	2
Domain MixUp	$58.72_{\pm 1.49}$	$99.20_{\pm 0.84}$	$96.98_{\pm 1.03}$	$84.97_{\pm 0.49}$	3
$\text{BaryFM}_{\text{Vertex}}$	$60.04_{\pm 1.08}$	$98.80_{\pm 0.45}$	$95.85_{\pm 1.14}$	$84.89_{\pm 0.58}$	4
$\text{BaryFM}_{\text{Random}}$	$62.20_{\pm 1.31}$	$99.40_{\pm 0.89}$	$96.23_{\pm 1.18}$	$85.94_{\pm 0.66}$	1

Table 29: Per domain Office 31 domain generalization results. Results denote classification accuracy (in %) over 5 seeds.

On Office 31 (Saenko et al., 2010), we report our results in Table 29. The performance of methods on dSLR and Webcam is comparable to domain adaptation (c.f. Table 16). This is intuitive, since for each of these targets, there is a closely resembling domain in the source data (e.g., Webcam in source, for dSLR as target). The biggest difference comes from Amazon (c.f., DA achieves around 70% accuracy, while DG achieves 62.20), as this is the most different and challenging domain. In comparison, $\texttt{BaryFM}_{\text{Random}}$ achieves the best rank, with `MixUp` being a close second best. We note the gap between `Domain MixUp` and our proposed $\texttt{BaryFM}_{\text{Random}}$ of 0.97 ($85.94_{\pm 0.66}$ ours vs. $84.97_{\pm 0.69}$ theirs), higher than seed noise.

Method	Art	Clipart	Product	Real World	Avg.	Rank
ERM (ResNet 50)	$65.68_{\pm 1.19}$	$53.91_{\pm 1.10}$	$79.37_{\pm 1.16}$	$80.00_{\pm 0.69}$	$69.74_{\pm 0.57}$	4
MixUp	$65.56_{\pm 0.94}$	$57.54_{\pm 1.43}$	$78.99_{\pm 0.69}$	$79.64_{\pm 0.55}$	$70.43_{\pm 0.34}$	3
Domain MixUp	$63.12_{\pm 2.06}$	$55.25_{\pm 1.83}$	$78.30_{\pm 1.16}$	$78.78_{\pm 0.75}$	$68.86_{\pm 0.82}$	5
$\text{BaryFM}_{\text{Vertex}}$	$66.43_{\pm 1.56}$	$55.74_{\pm 1.00}$	$80.15_{\pm 0.79}$	$79.87_{\pm 0.82}$	$70.55_{\pm 0.47}$	2
$\text{BaryFM}_{\text{Random}}$	$67.18_{\pm 1.34}$	$57.49_{\pm 1.11}$	$80.17_{\pm 0.92}$	$79.82_{\pm 0.38}$	$71.17_{\pm 0.76}$	1

Table 30: Per domain Office Home domain generalization results. denote classification accuracy (in %) over 5 seeds.

On Office Home (Venkateswara et al., 2017), we report our results in Table 30. Here, except for Domain MixUp, all methods are able to improve the ERM baseline over the seed standard deviation, with $\texttt{BaryFM}_{\text{Random}}$ being the best method. We provide an example of the different interpolations acquired by our method in the Office Home dataset in Figures 22 and 23. Note that, while Domain MixUp contains some degree of a ghosting effect, and the samples concentrate in the embedding space as the mixing weights go away from the vertices of the simplex. Our `BaryFM`, on the other hand, becomes more blurred, and its samples are more distributed in the latent space.

Method	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Avg.	Rank
ERM (ResNet 101)	$69.97_{\pm 0.33}$	$28.34_{\pm 0.94}$	$58.34_{\pm 1.20}$	$13.61_{\pm 0.07}$	$70.13_{\pm 0.34}$	$57.05_{\pm 0.71}$	$49.57_{\pm 0.24}$	4
MixUp	$69.71_{\pm 0.21}$	$28.24_{\pm 0.45}$	$58.99_{\pm 0.66}$	$13.50_{\pm 0.59}$	$69.59_{\pm 0.60}$	$57.19_{\pm 0.34}$	$49.53_{\pm 0.11}$	5
Domain MixUp	$71.00_{\pm 0.27}$	$28.91_{\pm 0.54}$	$59.53_{\pm 1.01}$	$14.97_{\pm 0.30}$	$70.93_{\pm 0.60}$	$58.43_{\pm 0.40}$	$50.63_{\pm 0.20}$	1
$\text{BaryFM}_{\text{Vertex}}$	$69.72_{\pm 0.39}$	$28.25_{\pm 0.81}$	$58.79_{\pm 0.82}$	$13.43_{\pm 0.09}$	$70.57_{\pm 0.68}$	$57.40_{\pm 1.18}$	$49.69_{\pm 0.34}$	3
$\text{BaryFM}_{\text{Random}}$	$70.57_{\pm 0.18}$	$28.68_{\pm 0.47}$	$59.83_{\pm 0.84}$	$13.83_{\pm 0.56}$	$70.25_{\pm 0.72}$	$57.99_{\pm 0.47}$	$50.19_{\pm 0.21}$	2

Table 31: Per domain DomainNet domain generalization results. Results denote classification accuracy (in %) over 3 seeds.

Next, we analyze DomainNet (Peng et al., 2019), which is the more challenging and large scale image benchmark. Overall, improvements in this benchmark are smaller (biggest difference being Domain MixUp vs. ERM, with $\delta = 1.06$). This is normal, due the hardness of the task. This dataset contains 345 classes, under severe domain shift (e.g., Quickdraw). Overall, Domain MixUp of Wang et al. (2020) proves very effective, being the best method over all domains. Our $\texttt{BaryFM}_{\text{Random}}$ comes in second. Nonetheless, we remark that both methods improve over the ERM, which proves the effectiveness of the augmentation strategy by mixing domain characteristics.

Method	S8	S9	Pool.	Rank
ERM (CBraMod)	$50.06_{\pm 4.84}$	$51.20_{\pm 2.78}$	$50.63_{\pm 3.45}$	5
MixUp	$51.04_{\pm 3.30}$	$53.70_{\pm 3.36}$	$52.37_{\pm 3.11}$	4
Domain MixUp	$50.89_{\pm 3.02}$	$54.51_{\pm 2.90}$	$52.70_{\pm 2.78}$	3
$\text{BaryFM}_{\text{Vertex}}$	$53.75_{\pm 4.06}$	$54.62_{\pm 3.76}$	$54.18_{\pm 3.48}$	1
$\text{BaryFM}_{\text{Random}}$	$52.71_{\pm 3.70}$	$54.76_{\pm 3.91}$	$53.73_{\pm 3.43}$	2

Table 32: Per domain BCI-IV-2a domain generalization results. denote classification accuracy (in %) over 5 seeds.

Now, we analyze the generalization results on BCI-CIV-2a (Brunner et al., 2008), which are run in the same setting as Wang et al. (2025). We use subjects 1 – 5 as sources, 6 – 7 for checkpoint selection, and 8 – 9 for testing. In general, EEG datasets have significantly wider confidence interval margins than image benchmarks, due the intrinsic difficulty of EEG decoding. Overall, our interpolations improve over the ERM and MixUp variations. For instance, our $\texttt{BaryFM}_{\text{Vertex}}$ is the best performing method with an improvement of 3.54% over ERM. The second best performing method is $\texttt{BaryFM}_{\text{Random}}$, with an improvement of 3.09 over ERM. The larger improvement over ERM goes in the same direction as our domain adaptation results (c.f. Table 19).

Method	S14	S15	S16	S17	Pool.	Rank
ERM (CBraMod)	$55.57_{\pm 4.05}$	$33.58_{\pm 5.75}$	$80.43_{\pm 3.16}$	$62.29_{\pm 7.41}$	$58.30_{\pm 2.44}$	5
MixUp	$53.46_{\pm 2.91}$	$39.40_{\pm 4.83}$	$80.79_{\pm 2.19}$	$59.13_{\pm 6.94}$	$61.11_{\pm 2.57}$	2
Domain MixUp	$57.46_{\pm 2.61}$	$40.36_{\pm 5.49}$	$82.77_{\pm 2.28}$	$68.12_{\pm 6.10}$	$61.04_{\pm 3.15}$	3
$\text{BaryFM}_{\text{Vertex}}$	$53.62_{\pm 2.43}$	$47.04_{\pm 4.94}$	$79.04_{\pm 2.09}$	$59.13_{\pm 5.07}$	$59.25_{\pm 5.63}$	4
$\text{BaryFM}_{\text{Random}}$	$54.93_{\pm 2.84}$	$43.18_{\pm 5.95}$	$80.68_{\pm 2.33}$	$63.45_{\pm 6.31}$	$62.08_{\pm 2.38}$	1

Table 33: Per domain SEED-VIG domain generalization results. Results denote Pearson correlation (in %) over 5 seeds.

Finally, we test augmentations on SEED-VIG (Zheng & Lu, 2017), where we use subjects 1 – 13 as source subjects, 14 – 17 as test subjects, and the rest for checkpoint selection. This split is different from the results used in domain adaptation (c.f. Table 21). Overall, our $\texttt{BaryFM}_{\text{Random}}$ has an improvement of 3.78% in Pearson correlation, above MixUp variants (2.81 and 2.74 for MixUp and Domain MixUp, respectively).

Conclusion. Overall, our $\texttt{BaryFM}_{\text{Random}}$, which consists of drawing samples from randomly sampled barycenters in $\mathcal{W}(p_{1:K})$ is the best augmentation method in 3 (Office 31, Office Home and SEED-VIG) our of 5 datasets, being more consistent than $\texttt{BaryFM}_{\text{Vertex}}$, which augments by sampling the vertices of the simplex. This demonstrates that there is a true advantage in mixing domain characteristics.