
Computing Optimal Transport Maps and Wasserstein Barycenters Using Conditional Normalizing Flows

Gabriele Visentin¹ Patrick Cheridito¹

Abstract

We present a novel method for efficiently computing optimal transport maps and Wasserstein barycenters in high-dimensional spaces. Our approach uses conditional normalizing flows to approximate the input distributions as invertible pushforward transformations from a common latent space. This makes it possible to directly solve the primal problem using gradient-based minimization of the transport cost, unlike previous methods that rely on dual formulations and complex adversarial optimization. We show how this approach can be extended to compute Wasserstein barycenters by solving a conditional variance minimization problem. A key advantage of our conditional architecture is that it enables the computation of barycenters for hundreds of input distributions, which was computationally infeasible with previous methods. Our numerical experiments illustrate that our approach yields accurate results across various high-dimensional tasks and compares favorably with previous state-of-the-art methods.

1. Introduction

Optimal transport (OT) and Wasserstein barycenters are active research areas with applications in economics, operations research, statistics, physics, PDE theory, and machine learning (Peyré & Cuturi, 2019; Villani, 2021). OT has been successfully applied to generative modeling (Arjovsky et al., 2017; Petzka et al., 2017; Wu et al., 2018; Liu et al., 2019; Cao et al., 2019; Leygonie et al., 2019) and domain adaptation (Luo et al., 2018; Shen et al., 2018; Xie et al., 2019), while Wasserstein barycenters, by providing a natural notion of average for probability distributions, found

applications in problems of shape interpolation (Solomon et al., 2015), image interpolation (Lacombe et al., 2023; Simon & Aberdam, 2020), color translation (Rabin et al., 2014), style translation (Mroueh, 2019), Bayesian subset posterior estimation (Srivastava et al., 2015), clustering in Wasserstein space (Del Barrio et al., 2019; Ho et al., 2017), and fairness (Chzhen et al., 2020; Gouic et al., 2020).

Traditional numerical methods work with discrete or discretized probability measures (Peyré & Cuturi, 2019), via linear programming (Anderes et al., 2016) and entropy regularization (Cuturi & Doucet, 2014; Solomon et al., 2015). They yield accurate results for discrete measures on low-dimensional spaces but scale poorly in the number of support points, so that their performance degrades for continuous distributions, especially in high-dimensional settings (Fan et al., 2020; Korotin et al., 2021b).

These limitations have generated substantial interest in numerical methods for OT and Wasserstein barycenters problems for continuous distributions. While other methods are reviewed in Section 4, we note here that almost all of them solve the Kantorovich dual of the OT problem and employ neural networks to parametrize either the dual potentials (Li et al., 2020) or the transport maps (Kolesov et al., 2024), which often results in complex bi-level (Kolesov et al., 2024; Korotin et al., 2019; Xie et al., 2019; Lu et al., 2020) or tri-level (Fan et al., 2020) adversarial learning.

Contributions. We propose a different approach. Our main contributions are the following:

1. We propose a method that bypasses complex adversarial training by directly solving the primal formulation of the OT problem. This approach directly yields both OT maps and values. In addition, it enables an intuitive reformulation of the Wasserstein barycenter problem as an expected conditional variance minimization problem.
2. In our setup, OT maps are bijective. Therefore, we approximate them with conditional normalizing flows, which are flexible bijective models and therefore a natural choice in this context.
3. For barycenter problems, our method directly yields

¹Department of Mathematics, ETH Zurich, Switzerland. Correspondence to: Gabriele Visentin <gabriele.visentin@math.ethz.ch>.

OT maps and their inverses between the input distributions and the barycenter without requiring additional computations. Moreover, it provides a generative model of the barycenter which allows for direct sampling without querying the input distributions.

4. Our model’s conditional architecture scales well in the number of input distributions and can be used to compute Wasserstein barycenters of hundreds of input distributions, which is computationally infeasible with existing methods.

2. Background and Preliminaries

In this section we recall the OT problem (Section 2.1) and the Wasserstein barycenter problem (Section 2.2).

Notation. We denote by $\|\cdot\|$ the Euclidean norm on $\mathbb{R}^d$ and by $L^p(\mu)$ the space $L^p(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d), \mu)$ equipped with the norm $\|f\|_{L^p(\mu)} := (\int_{\mathbb{R}^d} \|f(z)\|^p \mu(dz))^{1/p}$. $\mathcal{P}(\mathbb{R}^d)$ denotes the space of all Borel probability measures on $\mathbb{R}^d$, $\mathcal{P}_p(\mathbb{R}^d)$ is the space of all $\mu \in \mathcal{P}(\mathbb{R}^d)$ such that $\int_{\mathbb{R}^d} \|x\|^p \mu(dx) < \infty$, while $\mathcal{P}_{ac}(\mathbb{R}^d)$ is the space of all $\mu \in \mathcal{P}(\mathbb{R}^d)$ that are absolutely continuous with respect to the Lebesgue measure. Finally, we define $\mathcal{P}_{p,ac}(\mathbb{R}^d) := \mathcal{P}_p(\mathbb{R}^d) \cap \mathcal{P}_{ac}(\mathbb{R}^d)$.

2.1. Optimal Transport

The Monge-formulation (Monge, 1781) of the OT problem between two distributions $\mu, \nu \in \mathcal{P}(\mathbb{R}^d)$ for p -cost is:

$$\mathbb{W}_p^p(\mu, \nu) = \inf_{T_{\#}\mu = \nu} \int_{\mathbb{R}^d} \|x - T(x)\|^p \mu(dx), \quad (1)$$

where $T_{\#}\mu$ denotes the pushforward measure of μ under T . This problem is difficult to handle and does not always have a solution. Therefore, it was relaxed by Kantorovich (1942) as follows:

$$\mathbb{W}_p^p(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu, \nu)} \int_{\mathbb{R}^d} \|x - y\|^p \gamma(dx, dy), \quad (2)$$

where $\Gamma(\mu, \nu)$ is the set of all couplings between μ and ν , i.e. the set of all probability measures in $\mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d)$ with marginals μ and ν .

However, under some mild regularity assumptions, problems (1) and (2) are equivalent (see Lemma 3.1 below), and $\mathbb{W}_p(\mu, \nu)$ defines a metric on the space $\mathcal{P}_p(\mathbb{R}^d)$, known as the Wasserstein- p metric.

2.2. Wasserstein Barycenters

Wasserstein barycenters are Fréchet means on the Wasserstein spaces $(\mathcal{P}_p(\mathbb{R}^d), \mathbb{W}_p)$. In the following we focus on

the case $p = 2$, which is of most practical interest. Let $(\mu_s)_{s \in \mathcal{S}} \subseteq \mathcal{P}_{2,ac}(\mathbb{R}^d)$ be a family of distributions on $\mathbb{R}^d$ indexed by a finite set $\mathcal{S}$ and $(w_s, s \in \mathcal{S})$ non-negative weights such that $\sum_{s \in \mathcal{S}} w_s = 1$. Then the w -weighted Wasserstein-2 barycenter is given by:

$$\bar{\mu} = \arg \inf_{\nu \in \mathcal{P}_2(\mathbb{R}^d)} \sum_{s \in \mathcal{S}} w_s \mathbb{W}_2^2(\nu, \mu_s). \quad (3)$$

Under these assumptions, the Wasserstein-2 barycenter $\bar{\mu}$ exists, is unique and is absolutely continuous with respect to the Lebesgue measure (Agueh & Carlier, 2011; Brizzi et al., 2025).

3. Proposed Method

In this section, we first introduce some additional notation (Section 3.1). Then we derive our main theoretical results (Section 3.2) and present their algorithmic implementation (Section 3.3). The proofs of all lemmas and theorems can be found in Appendix B.

3.1. Notation

Given two Borel probability measures $\mu, \nu \in \mathcal{P}(\mathbb{R}^d)$ and a Borel function $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$, we say that a Borel function $\tilde{f} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a (μ, ν) -inverse of f if $\tilde{f} \circ f(x) = x$ for μ -almost all $x \in \mathbb{R}^d$ and $f \circ \tilde{f}(x) = x$ for ν -almost all $x \in \mathbb{R}^d$.

We denote by $B(\mu, \nu)$ the set of all Borel functions $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that $f_{\#}\mu = \nu$ and f admits a (μ, ν) -inverse. In Lemmas A.3 and A.4 below we show some basic facts about this set. In particular, we show that if $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$ and $f \in B(\mu, \nu)$, then $\|f\|$ belongs to $L^p(\mu)$, the (μ, ν) -inverse $\tilde{f}$ is ν -almost surely unique and it belongs to $B(\nu, \mu)$. A special case is when μ and ν have non-vanishing densities with respect to the Lebesgue measure on $\mathbb{R}^d$, then $B(\mu, \nu)$ is just the set of all almost everywhere bijective pushforward maps that push μ to ν . Furthermore, if $f \in B(\mu, \nu)$, then its (μ, ν) -inverse is almost everywhere unique and coincides almost everywhere with the inverse map f^{-1} , whenever the latter exists.

3.2. Theoretical Results

Bijective pushforward maps arise naturally in OT theory, as shown in the following version of Brenier’s theorem, which follows from Gangbo & McCann (1996).

Lemma 3.1. *Let $p > 1$ and $\mu, \nu \in \mathcal{P}_{p,ac}(\mathbb{R}^d)$. Then, Kantorovich’s problem (2) has a unique solution $\gamma \in \Gamma(\mu, \nu)$. Moreover, γ is of the form $(id, T)_{\#}\mu$, where $T \in B(\mu, \nu)$ is a μ -almost surely unique solution to Monge’s problem (1).*

The next theorem allows us to reformulate the OT problem

with p -cost as a constrained $L^p(\lambda)$ -minimization problem for a given latent distribution $\lambda \in \mathcal{P}_{p,ac}(\mathbb{R}^d)$.

Theorem 3.2. *Let $\lambda, \mu, \nu \in \mathcal{P}_{p,ac}(\mathbb{R}^d)$ for some $p > 1$. Then:*

$$\mathbb{W}_p^p(\mu, \nu) = \min_{\substack{f \in B(\lambda, \mu) \\ g \in B(\lambda, \nu)}} \|f - g\|_{L^p(\lambda)}^p.$$

Furthermore, if f and g are solutions of the optimization problem above and $\tilde{f}$ is a (λ, μ) -inverse of f , then $g \circ \tilde{f}$ is an optimal transport map between μ and ν .

Theorem 3.2 can be understood as a generalization of the closed-form solution of OT in one dimension using quantile functions. Recall, that for $\mu, \nu \in \mathcal{P}_p(\mathbb{R})$, one has:

$$\mathbb{W}_p^p(\mu, \nu) = \|q_\mu - q_\nu\|_{L^p([0,1])}^p,$$

where $L^p([0,1])$ is the L^p space over $[0,1]$ equipped with the uniform distribution and q_μ is the quantile function of μ (see, for instance, Remark 2.30 in [Peyré & Cuturi \(2019\)](#)). Theorem 3.2 shows that for $\mu, \nu \in \mathcal{P}_{p,ac}(\mathbb{R}^d)$ we can replace the uniform distribution with any absolutely continuous latent distribution λ and the comonotonicity constraint on q_μ and q_ν (which is implicit in the definition of quantile function) with closeness in $L^p(\lambda)$.

Our next result shows how to compute Wasserstein-2 barycenters as weighted averages of certain bijective push-forward maps that minimize an expected conditional variance criterion. Consider a non-empty finite set $\mathcal{S} = \{1, \dots, n\}$ and a collection of probability distributions $\mu_s \in \mathcal{P}_{2,ac}(\mathbb{R}^d)$, $s \in \mathcal{S}$, together with weights $w_s \geq 0$, $s \in \mathcal{S}$, such that $\sum_{s \in \mathcal{S}} w_s = 1$. Then the w -weighted Wasserstein-2 barycenter of $(\mu_s)_{s \in \mathcal{S}}$ is the unique $\bar{\mu} \in \mathcal{P}_{2,ac}(\mathbb{R}^d)$ satisfying Equation (3). In the following theorem, we realize $\bar{\mu}$ as the distribution of a d -dimensional random vector of the form $h(Z)$ for a Borel function $h : \mathbb{R}^d \rightarrow \mathbb{R}^d$ and a d -dimensional random vector Z defined on an underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$ that also carries an independent $\mathcal{S}$ -valued random variable S such that $\mathbb{P}(S = s) = w_s$ for all $s \in \mathcal{S}$. The expectation and variance corresponding to $\mathbb{P}$ are denoted by $\mathbb{E}$ and Var , respectively.

Theorem 3.3. *Let $\mathcal{S}$ be a non-empty finite set and consider a collection of probability distributions $\mu_s \in \mathcal{P}_{2,ac}(\mathbb{R}^d)$, $s \in \mathcal{S}$, and weights $w_s \geq 0$, $s \in \mathcal{S}$, satisfying $\sum_{s \in \mathcal{S}} w_s = 1$. Let Z be a d -dimensional random vector with an arbitrary distribution $\lambda \in \mathcal{P}_{2,ac}(\mathbb{R}^d)$ and S an independent $\mathcal{S}$ -valued random variable taking the values s with probabilities w_s , $s \in \mathcal{S}$. Then, the following hold:*

1. *The minimization problem*

$$\min_{\substack{f: \mathbb{R}^d \times \mathcal{S} \rightarrow \mathbb{R}^d \\ \text{s.t. } f(\cdot, s) \in B(\lambda, \mu_s) \\ \text{for all } s \in \mathcal{S}}} \sum_{i=1}^d \mathbb{E} [\text{Var}(f_i(Z, S) \mid Z)] \quad (4)$$

has a solution, and

2. *For every solution $f : \mathbb{R}^d \times \mathcal{S} \rightarrow \mathbb{R}^d$ of problem (4), the w -weighted Wasserstein-2 barycenter of $(\mu_s)_{s \in \mathcal{S}}$ is equal to the distribution of $h(Z)$, where $h : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is given by the weighted sum*

$$h(z) = \sum_{s \in \mathcal{S}} w_s f(z, s).$$

3.3. Implementation

Our approach uses conditional normalizing flows. After introducing them in Section 3.3.1, we present our algorithms for OT maps (Section 3.3.2) and Wasserstein barycenters (Section 3.3.3). In Appendix C we provide all necessary background information on normalizing flows.

3.3.1. CONDITIONAL NORMALIZING FLOWS.

Given a finite set of input distributions $(\mu_s)_{s \in \mathcal{S}} \in \mathcal{P}_{p,ac}(\mathbb{R}^d)$ and a reference distribution $\lambda \in \mathcal{P}_{p,ac}(\mathbb{R}^d)$, we use *conditional normalizing flows* to parametrize a map $f : \mathbb{R}^d \times \mathcal{S} \rightarrow \mathbb{R}^d$ such that $f(\cdot, s) \in B(\lambda, \mu_s)$ for all $s \in \mathcal{S}$, and we use this parametrization to solve the optimization problems in Theorem 3.2 and Theorem 3.3 with gradient descent in parameter space. [Teshima et al. \(2020\)](#) have shown that coupling-based normalizing flows are universal approximators for bijections with the caveat that their construction relies on ill-conditioned networks ([Koehler et al., 2021](#); [Draxler et al., 2024](#)). On the other hand, [Draxler et al. \(2024\)](#) have introduced a well-conditioned flow which can approximate arbitrary distributions but not necessarily every bijection.

We choose the standard d -dimensional Gaussian distribution as latent distribution λ and we enforce the pushforward constraint $f(\cdot, s)_\# \lambda = \mu_s$ via conditional likelihood maximization. The conditioning variable $s \in \mathcal{S}$ may be $\mathbb{R}$ -valued, taking finitely many values (as in Section 5.2.4, where $\mathcal{S}$ is a finite grid in $[0, 1]$) or one-hot encoded (as in all other numerical experiments), depending on how regular we expect the dependence of μ_s on s to be.

We design and implement our own conditional versions of Real NVP ([Dinh et al., 2016](#)) and Glow ([Kingma & Dhariwal, 2018](#)) with multi-scale architecture. In the case of Real NVP, we use a conditional affine coupling layer with the following coupling function:

$$f_{\theta(z^B, s)}(z^A) = z^A \odot \exp(a(z^B, s)) + b(z^B, s),$$

where the parameters $\theta(\cdot, \cdot) = (a(\cdot, \cdot), b(\cdot, \cdot))$ are feedforward neural networks, with an additional residual connection for the conditioning input s . A similar conditional architecture, but without the residual connection, was already proposed by [Atanov et al. \(2019\)](#), but in our experiments,

we found that adding this residual connection improves the performance. In the case of Glow, we obtain a conditional architecture simply by concatenating the conditioning variable s pixel-wise with the latent input image.

3.3.2. COMPUTING OT MAPS

Our method for computing OT maps using Theorem 3.2 is presented in Algorithm 1. The algorithm is given for the case of quadratic cost (i.e. $p = 2$), but it can also be used for OT costs of the form $c(x, y) = h(x - y)$ for a strictly convex function $h: \mathbb{R}^d \rightarrow [0, \infty)$ satisfying the conditions (H1)–(H3) of Gangbo & McCann (1996).

As shown in Algorithm 1, we learn the source distribution $\mu = \mu_1$ and the target distribution $\nu = \mu_2$ using a joint conditional normalizing flow with conditioning variable $s \in \{1, 2\}$. In principle, it would be possible to use two separate normalizing flows for μ_1 and μ_2 . But in our experiments, a conditional model resulted in better performance, probably due to the fact that this architecture introduces a smooth dependence on the conditioning variable $s \in \{0, 1\}$, which facilitates learning pushforward maps that are close in $L^2(\lambda)$.

Algorithm 1 OT Map via Conditional Normalizing Flows.

Input: input distributions μ_1 and μ_2 accessible by sampling; conditional normalizing flow $f_\theta(\cdot, \cdot) : \mathbb{R}^d \times \{1, 2\} \rightarrow \mathbb{R}^d$ with initialized parameters θ and latent distribution λ ; number of iterations T ; learning rate η ; decreasing weights $(\zeta_t)_{t=1}^T$.

Output: OT map from μ_1 to μ_2 given by $x \mapsto f_\theta(f_\theta^{-1}(x, 1), 2)$.

for $t = 1$ **to** T **do**

sample $S \sim \text{Uniform}(\{1, 2\})$, $X \sim \mu_S$, $Z \sim \lambda$;
 compute model likelihood on (X, S) :
 $p_\theta(X, S) = p_Z(f_\theta^{-1}(X, S)) \cdot |\det(Df_\theta^{-1}(X, S))|$;
 compute L^2 -cost:
 $L_\theta^2(Z) = \|f_\theta(Z, 1) - f_\theta(Z, 2)\|^2$;
 update model parameters θ by gradient descent:
 $\theta \leftarrow \theta - \eta \nabla_\theta (-\log(p_\theta(X, S)) + \zeta_t L_\theta^2(Z))$.

end for

Decreasing weights. Both Algorithm 1 and Algorithm 2 require as input a sequence of decreasing weights $(\zeta_t)_{t=1}^T$ for the L^2 -cost. This is motivated by the fact that we are trying to solve a multi-objective optimization problem with two competing objectives: the pushforward maps need to be as close as possible in $L^2(\lambda)$ while still being far enough to satisfy their pushforward constraints. Naively minimizing both the L^2 -cost and the negative log-likelihood in each gradient step leads to poor performance. Instead, we gradually decrease the importance of the L^2 -cost, thus guaranteeing that at convergence the pushforward constraints are satisfied. The optimal choice of weights depends on the relative

values of the optimal likelihood loss and L^2 -cost. In all our numerical experiments we chose an exponentially decreasing schedule; see Appendix D for a full description of the hyperparameters used in each numerical experiment.

At the end of training, the OT map from μ_1 to μ_2 can be efficiently retrieved by performing an inverse pass from μ_1 to λ followed by a forward pass from λ to μ_2 , i.e. it is given by $x \mapsto f_\theta(f_\theta^{-1}(x, 1), 2)$.

3.3.3. COMPUTING WASSERSTEIN-2 BARYCENTERS.

The algorithm for computing Wasserstein-2 barycenters builds upon Theorem 3.3 and is presented in Algorithm 2.

Algorithm 2 Wasserstein-2 Barycenter via Conditional Normalizing Flows.

Input: input distributions $(\mu_s)_{s \in \mathcal{S}}$ accessible by sampling; weights $(w_s)_{s \in \mathcal{S}}$ and weighting measure $\sigma(s) := w_s$; conditional normalizing flow $f_\theta(\cdot, \cdot) : \mathbb{R}^d \times \mathcal{S} \rightarrow \mathbb{R}^d$ with initialized parameters θ and latent distribution λ ; number of iterations T ; learning rate η ; decreasing weights $(\zeta_t)_{t=1}^T$.

Output: learned Wasserstein-2 barycenter $h_\# \lambda \approx \bar{\mu}$, where $h(z) = \sum_{s \in \mathcal{S}} w_s f_\theta(z, s)$.

for $t = 1$ **to** T **do**

sample $S \sim \sigma$, $X \sim \mu_S$, $Z \sim \lambda$;
 compute model likelihood on (X, S) :
 $p_\theta(X, S) = p_Z(f_\theta^{-1}(X, S)) \cdot |\det(Df_\theta^{-1}(X, S))|$;
 compute L^2 -cost:
 $L_\theta^2(Z, S) = \|f_\theta(Z, S) - \sum_{s \in \mathcal{S}} w_s f_\theta(Z, s)\|^2$;
 update model parameters θ by gradient descent:
 $\theta \leftarrow \theta - \eta \nabla_\theta (-\log(p_\theta(X, S)) + \zeta_t L_\theta^2(Z, S))$.

end for

Also in this case we employ a conditional normalizing flow architecture and train using a sequence of decreasing weights. At the end of training, we obtain a generative model of the barycenter given by $h_\# \lambda$ for the function $h(z) = \sum_{s \in \mathcal{S}} w_s f_\theta(z, s)$. Additionally, the OT map from any input distribution μ_s to the barycenter is given by $x \mapsto h(f_\theta^{-1}(x, s))$, while the OT map from a sampled barycenter datapoint $x = h(z)$ to an input distribution μ_s is given by $x \mapsto f_\theta(h^{-1}(x), s) = f_\theta(z, s)$.

4. Related Work

Traditional numerical methods tackle OT between discrete or discretized distributions with linear programming (Anderson et al., 2016; Peyré & Cuturi, 2019). The Sinkhorn-Knopp algorithm (Sinkhorn & Knopp, 1967; Cuturi, 2013) yields good results for entropy-regularized discrete OT problems in low dimensions. But discrete methods loose accuracy for continuous distributions and become infeasible in high-dimensional settings. This has lead to continuous

methods, most of which tackle a dual formulation of the OT problem with neural networks or kernel expansions; see e.g. Korotin et al. (2021a).

Methods for Wasserstein barycenters can be divided into non-generative and generative. While non-generative models are limited to transporting existing samples from the input distributions, generative models output a learned barycenter distribution which can be used for sampling.

Non-generative models. An early example of such a model has been given by Li et al. (2020), who exploit the dual formulation of the entropic regularized Wasserstein barycenter problem and parametrize the potentials using neural networks. Their method produces only approximate barycenters, due to the bias introduced by the entropic regularization, and outputs only the learned potentials, instead of the OT maps. Korotin et al. (2021b) parametrize potentials with input-convex neural networks (ICNNs) (Amos et al., 2017) and tackle the dual OT problem by imposing a congruence and a cycle-consistency condition. The congruence condition requires fixing a dominating measure that may be difficult to choose *a priori*, while there is evidence that the cycle-consistency condition (Amos et al., 2017) and the use of ICNNs (Korotin et al., 2021a) may be suboptimal. Kolesov et al. (2024) present a model that can be applied to generic cost functionals (both regularized and exact), but relies on bi-level adversarial learning and does not provide a generative model of the barycenter.

Generative models. More closely related to our approach are generative models of Wasserstein barycenters. Also for this class of models the dominant approach is to start from the dual formulation and to solve a tri-level or bi-level optimization problem. Fan et al. (2020) solve the barycenter problem by computing Wasserstein distances through a min-max-min tri-level optimization problem using ICNNs, which may be unstable and may result in under-training of the generative model of the barycenter (Korotin et al., 2021b). Korotin et al. (2022) use the fixed point algorithm of Álvarez-Esteban et al. (2016) and the reversed maximin neural OT solver by Dam et al. (2019) to obtain a VAE-based generative model of the barycenter through an iterative procedure. Unfortunately, their fixed point algorithm is not guaranteed to converge. Additionally, the model does not output the OT maps between the input distributions and the barycenter, which therefore, need to be computed in an additional step. Finally, we mention two early generative methods that share some similarities with our approach. Lu et al. (2020) also tackle directly the primal formulation of the OT problem, but enforce the pushforward constraint by adversarial learning (due to their choice of a GAN as generative model) and the bijectivity constraint through a cycle-consistency condition. The model of Xie et al. (2019) also uses a bijective generative model, specifically a neural

ODE model, for the optimal coupling, but they enforce the pushforward constraint using the Kantorovich-Rubinstein duality for the Wasserstein-1 distance, which results in a minimax optimization problem. Both these models do not address the problem of computing Wasserstein barycenters.

Advantages of generative models. Depending on the application of interest, generative models may offer substantial benefits over non-generative ones. For instance, in shape interpolation, the density of the generative model is precisely the interpolating shape and does not require any surface reconstruction from point clouds. In style transfer the generative model can generate previously unseen samples (e.g. new MNIST digits in a given style), as opposed to just transporting already existing ones. In fair regression the generative model effectively performs distributional regression for the fair premium, which allows, for instance, the construction of confidence intervals, the estimation of any statistic of the barycenter distribution, as well as further transformations of the distribution; e.g. the barycenter can be shifted to have mean zero, which is useful, for instance, in actuarial applications.

Since our model is generative, in Section 5.2 we compare with other state-of-the-art generative models, specifically SCW₂B (Fan et al., 2020) and WIN (Korotin et al., 2019).

5. Numerical Experiments

We showcase the performance of our method in a series of numerical experiments. Section 5.1 presents results for the computation of OT maps in high-dimensional settings using Algorithm 1, while Section 5.2 deals with the computation of Wasserstein-2 barycenters on the following data sets: the Swiss roll data set (Section 5.2.1), two high-dimensional location-scatter data sets (Section 5.2.2), the MNIST data set (Section 5.2.3), a high-dimensional Gaussian data set with a large number of input distributions (Section 5.2.4), and a real-life dataset for multivariate fair regression (Section 5.2.5). Hyperparameter choices and all other implementation details are given in Appendix D. Our source code is available online at <https://github.com/gvisen/NormalizingFlowsBarycenter>.

Location-scatter families. Location-scatter families have become standard benchmarks for Wasserstein barycenter models (Korotin et al., 2019; 2021b; Kolesov et al., 2024). They are families of distributions obtained by transforming a base distribution $\mu_0 \in \mathcal{P}(\mathbb{R}^d)$ through invertible affine transformations $f_{M,u} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ of the form $f_{M,u} = Mx + u$, for a positive definite matrix $M \in \mathbb{R}^{d \times d}$ and a vector $u \in \mathbb{R}^d$. The Wasserstein-2 barycenter of finitely many distributions belonging to a given location-scatter family can be computed numerically using a fixed-point algorithm (Álvarez-Esteban et al., 2016). In the following experiments,

whenever we mention a location-scatter data set, we will implicitly reproduce the experimental set-up of [Korotin et al. \(2021b\)](#). In particular, we fix $\mathcal{S} = \{1, 2, 3, 4\}$, a vector of weights $w = (w_s)_{s \in \mathcal{S}} = (0.4, 0.3, 0.2, 0.1)$, and choose as base distribution μ_0 either the Swiss roll distribution ($d = 2$), the standard Gaussian distribution $\mathcal{N}(0, I)$ on $\mathbb{R}^d$ or the uniform distribution $\text{Uniform}([- \sqrt{3}, \sqrt{3}]^d)$. We then generate four distributions $(\mu_s)_{s \in \mathcal{S}}$ from the base distribution by setting $\mu_s := f_{M_s, 0} \# \mu_0$, where $f_{M_s, 0}$ is an affine transformation with positive definite matrix $M_s := R_s^T \Lambda R_s$ and zero shift, R_s is a random rotation matrix sampled uniformly from the Haar measure on $\text{SO}(d)$ and $\Lambda \in \mathbb{R}^{d \times d}$ is a diagonal matrix with diagonal $(\frac{1}{2}b^0, \frac{1}{2}b^1, \dots, \frac{1}{2}b^{d-1})$, for $b = 4^{1/(d-1)}$.

Metrics. As proposed in [Korotin et al. \(2021b\)](#), we use the unexplained variance percentage to measure the relative difference between a target distribution μ with its estimate $\hat{\mu}$:

$$\text{UVP}(\mu, \hat{\mu}) := 100 \cdot \frac{\mathbb{W}_2^2(\mu, \hat{\mu})}{\text{Var}(\mu)} (\%).$$

In a high-dimensional setting it's not possible to compute $\text{UVP}(\mu, \hat{\mu})$ exactly, due to the intractability of $\mathbb{W}_2^2(\mu, \hat{\mu})$. Nevertheless, as shown in Lemma A.2 of [Korotin et al. \(2019\)](#), if $\mu = T \# \mu_0$ and $\hat{\mu} = \hat{T} \# \mu_0$, then the following upper bound holds:

$$\text{UVP}(\mu, \hat{\mu}) \leq 100 \frac{\|T - \hat{T}\|_{L^2(\mu_0)}^2}{\text{Var}(T \# \mu_0)} =: \mathcal{L}^2\text{-UVP}(T, \hat{T}; \mu_0).$$

Additionally, as shown by [Dowson & Landau \(1982\)](#), the following lower bound holds:

$$\text{BW}_2^2\text{-UVP}(\mu, \hat{\mu}) := 100 \frac{\text{BW}_2^2(\mu, \hat{\mu})}{\text{Var}(\mu)} \leq \text{UVP}(\mu, \hat{\mu}),$$

where $\text{BW}_2^2(\mu, \hat{\mu})$ denotes the Bures–Wasserstein metric on Gaussian distributions, computed using the means and covariance matrices of μ and $\hat{\mu}$.

When solving an OT problem between μ_1 and μ_2 with known OT map T , we evaluate the performance of our estimated optimal transport map $\hat{T}$ by computing $\mathcal{L}^2\text{-UVP}(T, \hat{T}; \mu_1)$ and $\text{BW}_2^2\text{-UVP}(\mu_2, \hat{\mu}_2)$. For barycenter problems, we compute the metrics $\mathcal{L}^2\text{-UVP}(T_s, \hat{T}_s; \mu_s)$ for each $s \in \mathcal{S}$ (where T_s and $\hat{T}_s$ are the true and estimated OT maps from μ_s to the barycenter $\bar{\mu}$) and then report the mean $\mathcal{L}^2\text{-UVP}$ metric, averaged over all $s \in \mathcal{S}$ using the weights $(w_s)_{s \in \mathcal{S}}$. Additionally, we report the $\text{BW}_2^2\text{-UVP}(\bar{\mu}, \hat{\mu})$ metric, where $\bar{\mu}$ is the true barycenter and $\hat{\mu}$ is the model estimate.

5.1. Computing OT Maps

We first validate Algorithm 1 by computing OT maps between high-dimensional distributions, for which close-form

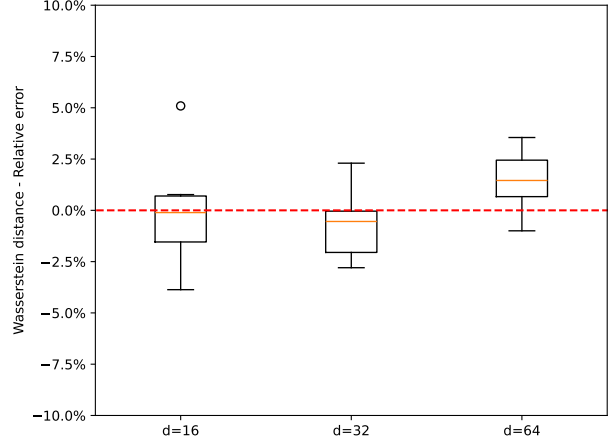


Figure 1. $\mathbb{W}_2^2$ relative error on high-dimensional Gaussian data.

solutions are known (see Remark 2.11 in [Peyré & Cuturi \(2019\)](#)). We report the $\mathcal{L}^2\text{-UVP}$ and $\text{BW}_2^2\text{-UVP}$ metrics in Table 1 and Table 2, for Gaussian and uniform distributions respectively, together with their standard deviation from ten trials. In terms of Wasserstein-2 distance, in Figure 1 we report a box plot of the (signed) relative error of the estimated distance computed on high-dimensional Gaussian data for ten random initializations of our model. All results show a very good performance in high-dimensions.

5.2. Computing Wasserstein-2 Barycenters

5.2.1. SWISS ROLL DATASET

We begin by evaluating Algorithm 2 qualitatively on a two-dimensional location-scatter data set with a Swiss roll base distribution, $\mathcal{S} = \{1, 2, 3, 4\}$ and weights $w = (0.4, 0.3, 0.2, 0.1)$. Figure 2 shows samples from the true input distributions $(\mu_s)_{s \in \mathcal{S}}$ (first row) and samples generated from our trained model (second row). At convergence, all input distributions have been learned well, which implies that the pushforward constraints are fully satisfied. In Figure 3 we compare samples from the true barycenter and the learned barycenter, i.e. $h \# \lambda$. Unlike other generative models (see Figure 1 in [Korotin et al. \(2021b\)](#) for a comparison with SCW_2B), our learned barycenter distribution is sharp and reproduces the highly non-linear structure of the Swiss roll distribution. We also verify in Figure 4 that the learned OT maps correctly transport samples from the input distributions to the barycenter via the transformations $x \mapsto h(f_\theta^{-1}(x, s))$, for all $s \in \mathcal{S}$.

5.2.2. HIGH-DIMENSIONAL EXPERIMENTS

We now compare our method's performance to other state-of-the-art models on two location-scatter benchmark

Table 1. $\mathcal{L}^2$ -UVP and BW_2^2 -UVP metrics for our method when computing OT maps between high-dimensional Gaussian distributions.

METRIC	$d = 2$	$d = 4$	$d = 8$	$d = 16$	$d = 32$	$d = 64$	$d = 128$
$\mathcal{L}^2$ -UVP	0.004 ± 0.001	0.086 ± 0.001	0.204 ± 0.002	0.623 ± 0.004	2.389 ± 0.018	1.658 ± 0.005	3.102 ± 0.009
BW_2^2 -UVP	0.002 ± 0.001	0.015 ± 0.005	0.008 ± 0.001	0.014 ± 0.001	0.034 ± 0.002	0.054 ± 0.002	0.088 ± 0.002

 Table 2. $\mathcal{L}^2$ -UVP and BW_2^2 -UVP metrics for our method when computing OT maps between high-dimensional uniform distributions.

METRIC	$d = 2$	$d = 4$	$d = 8$	$d = 16$	$d = 32$	$d = 64$	$d = 128$
$\mathcal{L}^2$ -UVP	0.54 ± 0.006	1.506 ± 0.019	2.046 ± 0.031	1.404 ± 0.006	4.255 ± 0.016	3.833 ± 0.014	7.137 ± 0.012
BW_2^2 -UVP	0.007 ± 0.001	0.029 ± 0.003	0.047 ± 0.004	0.036 ± 0.003	0.704 ± 0.009	0.738 ± 0.01	1.223 ± 0.005

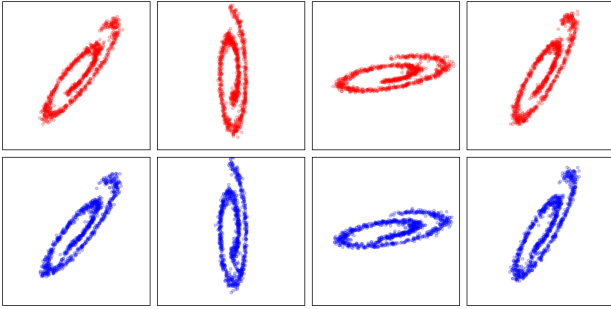


Figure 2. Samples from the true input distributions (first row) and from the input distributions as learned by our model (second row).

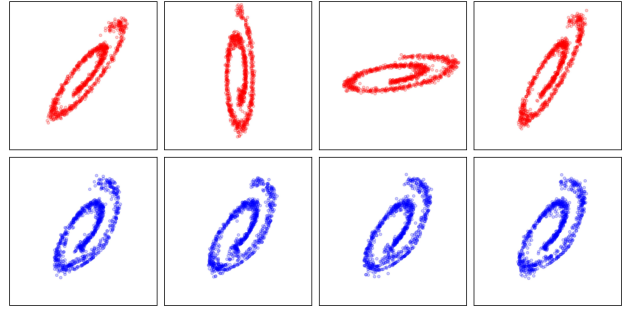
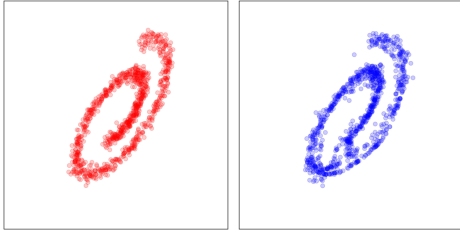

 Figure 4. Samples from the input distributions μ_s (first row) and their image after being transported to the barycenter (second row).


Figure 3. True barycenter (left) and learned barycenter (right).

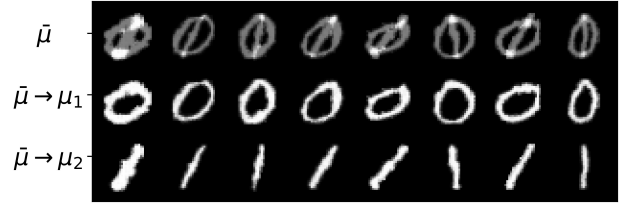


Figure 5. Sample from learned barycenter transported to the two input distributions.

datasets across a range of input dimensions (from $d = 2$ to $d = 128$): a Gaussian data set (see Table 3) and a uniform one (see Table 4). Our model has better performance than the competitors, SCW₂B (Fan et al., 2020) and WIN (Korotin et al., 2022), with better or equal training time (for more details, see Appendix D).

5.2.3. MNIST DIGITS

Next, we test our model on image data using the MNIST data set, using our conditional Glow architecture. We first compute the barycenter of the empirical distributions of the digits zero and one. We recover the well-known result that

each barycenter sample is the average of images from the input measures (Korotin et al., 2019). In Figure 5 we show samples from the learned barycenter and their images when transported to the two input distributions. While Figure 6 and Figure 7 show the inverse operation on samples from the true input distributions.

We also train our model using as input distributions the empirical distributions of all ten digits. Thanks to our model’s conditional architecture, the training time scales well in the number of input distributions, therefore solving this Wasserstein barycenter task did not take more training time than the previous one. In Figure 8 we show samples from the

Table 3. $\mathcal{L}^2$ -UVP and BW_2^2 -UVP metrics on location-scatter Gaussian data (from Table 5 in Korotin et al. (2019) for SCW_2B and WIN).

METRIC	METHOD	$d = 2$	$d = 4$	$d = 8$	$d = 16$	$d = 32$	$d = 64$	$d = 128$
$\mathcal{L}^2$ -UVP	OURS	0.025 ± 0.0	0.079 ± 0.002	0.076 ± 0.001	0.1 ± 0.001	0.382 ± 0.001	0.544 ± 0.001	1.55 ± 0.002
BW_2^2 -UVP	SCW_2B	0.070	0.090	0.160	0.280	0.430	0.590	1.280
	WIN	0.010	0.020	0.010	0.080	0.110	0.230	0.380
	OURS	0.005 ± 0.001	0.004 ± 0.001	0.004 ± 0.001	0.007 ± 0.001	0.024 ± 0.001	0.046 ± 0.001	0.094 ± 0.001

 Table 4. $\mathcal{L}^2$ -UVP and BW_2^2 -UVP metrics on location-scatter uniform data (from Table 5 in Korotin et al. (2019) for SCW_2B and WIN).

METRIC	METHOD	$d = 2$	$d = 4$	$d = 8$	$d = 16$	$d = 32$	$d = 64$	$d = 128$
$\mathcal{L}^2$ -UVP	OURS	0.998 ± 0.016	0.399 ± 0.005	0.498 ± 0.003	0.697 ± 0.002	1.301 ± 0.002	2.428 ± 0.002	5.506 ± 0.004
BW_2^2 -UVP	SCW_2B	0.120	0.100	0.190	0.290	0.460	0.600	1.380
	WIN	0.040	0.060	0.060	0.080	0.110	0.270	0.460
	OURS	0.01 ± 0.004	0.053 ± 0.007	0.033 ± 0.005	0.058 ± 0.003	0.081 ± 0.002	0.159 ± 0.002	0.362 ± 0.003

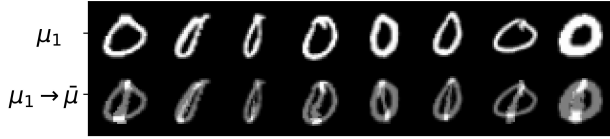


Figure 6. Sample from the first input distribution transported to the barycenter.



Figure 7. Sample from the second input distribution transported to the barycenter.

learned barycenter and their images when transported to all ten input distributions.

We illustrate a style translation application of our model in Figure 9, where we transport samples from a source distribution (e.g. the distribution of zeros), through the latent space, to a target distribution (e.g. the distribution of ones). Under this transformation, stylistic features of the zero digit – such as its slant and thickness – are preserved, showing that the latent space in our model encodes structural features of the input data and can be used to transfer them.

5.2.4. LARGE NUMBER OF INPUT DISTRIBUTIONS

In this section, we leverage the conditional architecture of our model to compute Wasserstein-2 barycenters for a large number of input distributions. Specifically, we evaluate

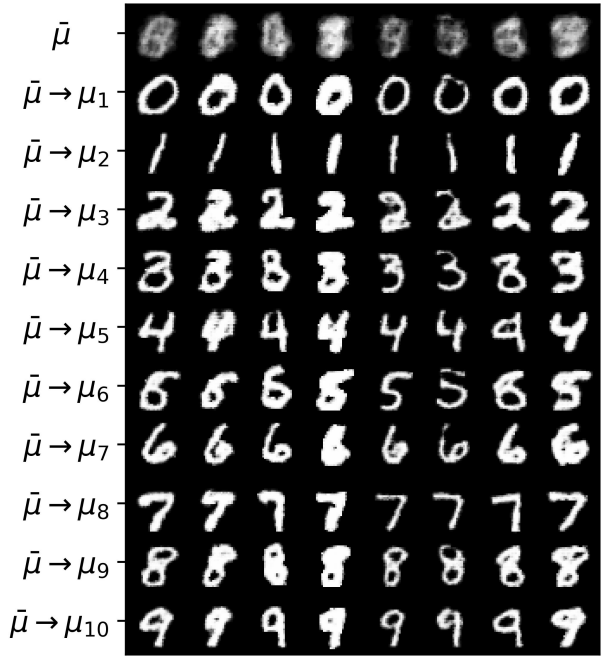


Figure 8. Sample from learned barycenter transported to all input distributions.

our model on high-dimensional Gaussian data sets ($d = 64$) with various numbers of input distributions ($n = 8, 64, 128$). Different input distributions are indexed by $\mathcal{S}$, the uniform grid on $[0, \pi]$ with n points, and are zero-mean Gaussian distributions with covariance $\Sigma_s = R(s)^T \Sigma_0 R(s)$, where Σ_0 is a diagonal matrix with diagonal $(2, 1/2, \dots, 1/2)$ and $R(s)$ is a rotation matrix describing a rotation in the first two coordinates by an angle $s \in \mathcal{S} \subset [0, \pi]$. In this way we can generate a large number of input distributions that vary smoothly in the univariate parameter s . We then com-

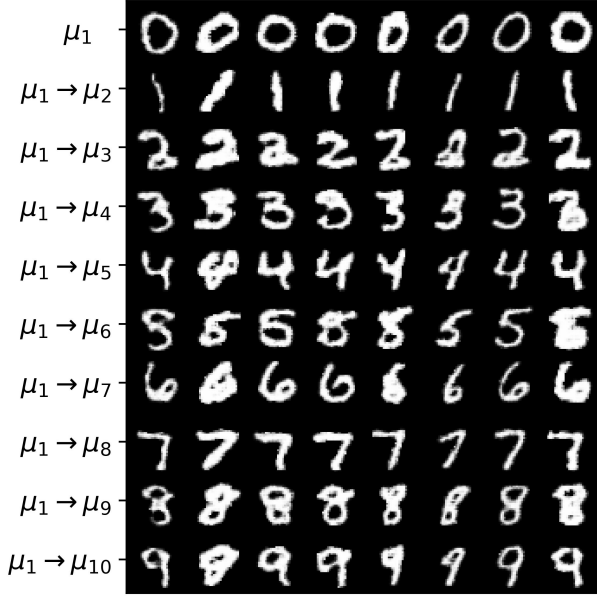


Figure 9. Sample from first input distribution transported to all other input distributions.

pute the Wasserstein-2 barycenters of these distributions using Algorithm 2. The results are presented in Table 5 and show no degradation in performance as the number of input distributions increases.

Table 5. Performance as number of input distributions increases.

METRIC	$n = 4$	$n = 16$	$n = 64$	$n = 128$
$\mathcal{L}^2$ -UVP	0.057	0.119	0.081	0.073
BW $_2^2$ -UVP	0.053	0.031	0.016	0.051

5.2.5. MULTIVARIATE FAIR REGRESSION

Fair regression, in the sense of demographic parity, looks for a regression function $f(X, S)$ that minimizes the cost $\mathbb{E}[\|Y - f(X, S)\|^2]$, such that $f(X, S)$ is independent of S . In this context S denotes sensitive features, such as race or gender, with respect to which discrimination is not allowed. The solution to this constrained optimization is precisely the Wasserstein-2 barycenter of the conditional distributions of Y given S (Chzhen et al., 2020; Gouic et al., 2020). To demonstrate the applicability of our model to multivariate fair regression, we work on the dataset “Communities and Crime” (Redmond, 2002), which is a benchmark dataset in the fairness literature, and we regress the target variables “percentage of officers assigned to drug units” and “total number of violent crimes per 100k population” (target variable Y) on 127 socio-economic features (non-sensitive

features X) and the percentage of the population that is African-American (sensitive feature S with range $[0, 1]$).

As shown in Table 6, a standard regression leads to strong correlation between the first predicted variable and the sensitive feature for several correlation measures, while our fair regression achieves almost perfect uncorrelatedness. In all cases, we fail to reject the null hypothesis of no association.

Table 6. Correlation between regressor and sensitive feature.

CORRELATION COEFFICIENT	STANDARD REGRESSION	FAIR REGRESSION
PEARSON	0.43	0.0003 (P-VALUE: 0.99)
SPEARMAN	0.47	0.0058 (P-VALUE: 0.80)
KENDALL	0.32	0.0039 (P-VALUE: 0.80)

We point out that this experiment requires computing the barycenter of 100 input distributions (the cardinality of the range of S , rounded up to the nearest percentage point), which would be computationally challenging for any other numerical Wasserstein barycenter method.

6. Conclusion

We have introduced a new method for computing OT maps in high-dimensional settings using conditional normalizing flows. Unlike previous methods, we avoid complex adversarial optimization and directly solve the primal OT problem using gradient-based minimization of the transport cost. We have shown how the approach can be extended to compute Wasserstein barycenters by solving a conditional variance minimization problem, and we have compared to state-of-the-art competitor models on several benchmarks. Our approach yields accurate results and can be used to compute Wasserstein barycenters for hundreds of input distributions, which was computationally infeasible with previous methods.

7. Limitations

A limitation of our approach is that Algorithm 2 applies only to Wasserstein-2 barycenters. Its extension to Wasserstein- p barycenters for $p \neq 2$ is currently still an open research question. Furthermore, training might be computationally intensive on image datasets much larger than the ones we tested on, i.e. for $d \gg 1000$.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be

specifically highlighted here.

References

- Agueh, M. and Carlier, G. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2): 904–924, 2011.
- Álvarez-Esteban, P. C., Del Barrio, E., Cuesta-Albertos, J., and Matrán, C. A fixed-point approach to barycenters in Wasserstein space. *Journal of Mathematical Analysis and Applications*, 441(2):744–762, 2016.
- Amos, B., Xu, L., and Kolter, J. Z. Input convex neural networks. In *International Conference on Machine Learning*, pp. 146–155. PMLR, 2017.
- Anderes, E., Borgwardt, S., and Miller, J. Discrete Wasserstein barycenters: Optimal transport for discrete data. *Mathematical Methods of Operations Research*, 84:389–409, 2016.
- Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223. PMLR, 2017.
- Atanov, A., Volokhova, A., Ashukha, A., Sosnovik, I., and Vetrov, D. Semi-conditional normalizing flows for semi-supervised learning. *arXiv preprint arXiv:1905.00505*, 920, 2019.
- Bogachev, V. I. and Ruas, M. A. S. *Measure Theory*, volume 1. Springer, 2007.
- Brizzi, C., Friesecke, G., and Ried, T. p -Wasserstein barycenters. *Nonlinear Analysis*, 251:113687, 2025.
- Brown, B. C., Caterini, A. L., Ross, B. L., Cresswell, J. C., and Loaiza-Ganem, G. Verifying the union of manifolds hypothesis for image data. *arXiv preprint arXiv:2207.02862*, 2022.
- Cao, J., Mo, L., Zhang, Y., Jia, K., Shen, C., and Tan, M. Multi-marginal Wasserstein GAN. *Advances in Neural Information Processing Systems*, 32, 2019.
- Chzhen, E., Denis, C., Hebiri, M., Oneto, L., and Pontil, M. Fair regression with Wasserstein barycenters. *Advances in Neural Information Processing Systems*, 33: 7321–7331, 2020.
- Cuturi, M. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. *Advances in Neural Information Processing Systems*, 26, 2013.
- Cuturi, M. and Doucet, A. Fast computation of Wasserstein barycenters. In *International Conference on Machine Learning*, pp. 685–693. PMLR, 2014.
- Dam, N., Hoang, Q., Le, T., Nguyen, T. D., Bui, H., and Phung, D. Three-player Wasserstein GAN via amortised duality. In *International Joint Conference on Artificial Intelligence 2019*, pp. 2202–2208. Association for the Advancement of Artificial Intelligence (AAAI), 2019.
- Del Barrio, E., Cuesta-Albertos, J. A., Matrán, C., and Mayo-Íscar, A. Robust clustering tools based on optimal transportation. *Statistics and Computing*, 29:139–160, 2019.
- Dinh, L., Krueger, D., and Bengio, Y. NICE: Non-linear Independent Components Estimation. *arXiv preprint arXiv:1410.8516*, 2014.
- Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using Real NVP. *arXiv preprint arXiv:1605.08803*, 2016.
- Dowson, D. and Landau, B. The Fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455, 1982.
- Draxler, F., Wahl, S., Schnörr, C., and Köthe, U. On the universality of volume-preserving and coupling-based normalizing flows. *arXiv preprint arXiv:2402.06578*, 2024.
- Durrett, R. *Probability: Theory and Examples*, volume 49. Cambridge University Press, 2019.
- Fan, J., Taghvaei, A., and Chen, Y. Scalable computations of Wasserstein barycenter via input convex neural networks. *arXiv preprint arXiv:2007.04462*, 2020.
- Folland, G. B. *Real Analysis: Modern Techniques and their Applications*, volume 40. John Wiley & Sons, 1999.
- Gangbo, W. and McCann, R. J. The geometry of optimal transportation. *Acta Mathematica*, 177(2):113–161, 1996.
- Gong, S., Boddeti, V. N., and Jain, A. K. On the intrinsic dimensionality of image representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3987–3996, 2019.
- Gouic, T. L., Loubes, J.-M., and Rigollet, P. Projection to fairness in statistical learning. *arXiv preprint arXiv:2005.11720*, 2020.
- Ho, N., Nguyen, X., Yurochkin, M., Bui, H. H., Huynh, V., and Phung, D. Multilevel clustering via Wasserstein means. In *International Conference on Machine Learning*, pp. 1501–1509. PMLR, 2017.
- Kantorovich, L. V. On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pp. 199–201, 1942.

- Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *Advances in Neural Information Processing Systems*, 31, 2018.
- Kobyzev, I., Prince, S. J., and Brubaker, M. A. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, 2020.
- Koehler, F., Mehta, V., and Risteski, A. Representational aspects of depth and conditioning in normalizing flows. In *International Conference on Machine Learning*, pp. 5628–5636. PMLR, 2021.
- Kolesov, A., Mokrov, P., Udovichenko, I., Gazdieva, M., Pammer, G., Burnaev, E., and Korotin, A. Estimating barycenters of distributions with neural optimal transport. In *International Conference on Machine Learning*, pp. 25016–25041. PMLR, 2024.
- Korotin, A., Egiazarian, V., Asadulaev, A., Safin, A., and Burnaev, E. Wasserstein-2 generative networks. *arXiv preprint arXiv:1909.13082*, 2019.
- Korotin, A., Li, L., Genevay, A., Solomon, J. M., Filippov, A., and Burnaev, E. Do neural optimal transport solvers work? A continuous Wasserstein-2 benchmark. *Advances in Neural Information Processing Systems*, 34:14593–14605, 2021a.
- Korotin, A., Li, L., Solomon, J., and Burnaev, E. Continuous Wasserstein-2 barycenter estimation without minimax optimization. *arXiv preprint arXiv:2102.01752*, 2021b.
- Korotin, A., Egiazarian, V., Li, L., and Burnaev, E. Wasserstein iterative networks for barycenter estimation. *Advances in Neural Information Processing Systems*, 35: 15672–15686, 2022.
- Lacombe, J., Digne, J., Courty, N., and Bonneel, N. Learning to generate Wasserstein barycenters. *Journal of Mathematical Imaging and Vision*, 65(2):354–370, 2023.
- Leygonie, J., She, J., Almahairi, A., Rajeswar, S., and Courville, A. Adversarial computation of optimal transport maps. *arXiv preprint arXiv:1906.09691*, 2019.
- Li, L., Genevay, A., Yurochkin, M., and Solomon, J. M. Continuous regularized Wasserstein barycenters. *Advances in Neural Information Processing Systems*, 33: 17755–17765, 2020.
- Liu, H., Gu, X., and Samaras, D. Wasserstein GAN with quadratic transport cost. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4832–4841, 2019.
- Loaiza-Ganem, G., Ross, B. L., Hosseinzadeh, R., Caterini, A. L., and Cresswell, J. C. Deep generative models through the lens of the manifold hypothesis: A survey and new connections. *Transactions on Machine Learning Research*, 09, 2024.
- Lu, G., Zhou, Z., Shen, J., Chen, C., Zhang, W., and Yu, Y. Large-scale optimal transport via adversarial training with cycle-consistency. *arXiv preprint arXiv:2003.06635*, 2020.
- Luo, Y., Zhang, S.-Y., Zheng, W.-L., and Lu, B.-L. WGAN domain adaptation for eeg-based emotion recognition. In *International Conference on Neural Information Processing*, pp. 275–286, 2018.
- Monge, G. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pp. 666–704, 1781.
- Mroueh, Y. Wasserstein style transfer. *arXiv preprint arXiv:1905.12828*, 2019.
- Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.
- Petzka, H., Fischer, A., and Lukovnicov, D. On the regularization of Wasserstein GANs. *arXiv preprint arXiv:1709.08894*, 2017.
- Peyré, G. and Cuturi, M. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.
- Pope, P., Zhu, C., Abdelkader, A., Goldblum, M., and Goldstein, T. The intrinsic dimension of images and its impact on learning. *arXiv preprint arXiv:2104.08894*, 2021.
- Rabin, J., Ferradans, S., and Papadakis, N. Adaptive color transfer with relaxed optimal transport. In *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 4852–4856. IEEE, 2014.
- Redmond, M. Communities and Crime. UCI Machine Learning Repository, 2002. DOI: <https://doi.org/10.24432/C53W3X>.
- Shen, J., Qu, Y., Zhang, W., and Yu, Y. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 4058–4065, 2018.
- Simon, D. and Aberdam, A. Barycenters of natural images constrained Wasserstein barycenters for image morphing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7910–7919, 2020.

- Sinkhorn, R. and Knopp, P. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.
- Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., and Guibas, L. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (ToG)*, 34(4):1–11, 2015.
- Srivastava, S., Cevher, V., Dinh, Q., and Dunson, D. WASP: Scalable Bayes via barycenters of subset posteriors. In *Artificial Intelligence and Statistics*, pp. 912–920. PMLR, 2015.
- Stimper, V., Liu, D., Campbell, A., Berenz, V., Ryll, L., Schölkopf, B., and Hernández-Lobato, J. M. normflows: A PyTorch Package for Normalizing Flows. *Journal of Open Source Software*, 8(86):5361, 2023. doi: 10.21105/joss.05361. URL <https://doi.org/10.21105/joss.05361>.
- Teshima, T., Ishikawa, I., Tojo, K., Oono, K., Ikeda, M., and Sugiyama, M. Coupling-based invertible neural networks are universal diffeomorphism approximators. *Advances in Neural Information Processing Systems*, 33:3362–3373, 2020.
- Villani, C. *Topics in Optimal Transportation*, volume 58. American Mathematical Soc., 2021.
- Wu, J., Huang, Z., Thoma, J., Acharya, D., and Van Gool, L. Wasserstein divergence for GANs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 653–668, 2018.
- Xie, Y., Chen, M., Jiang, H., Zhao, T., and Zha, H. On scalable and efficient computation of large scale optimal transport. In *International Conference on Machine Learning*, pp. 6882–6892. PMLR, 2019.

A. Auxiliary Results

The following are some standard results from measure and integration theory that are used in this paper.

Lemma A.1 (Change of variables). *Let $f : X \rightarrow Y$ be a measurable function between two measurable spaces $(X, \mathcal{A})$ and $(Y, \mathcal{B})$. If μ is probability measure on $(X, \mathcal{A})$, then a measurable function $g : Y \rightarrow \mathbb{R}$ satisfies $g \in L^1(f_{\#}\mu)$ if and only if $g \circ f \in L^1(\mu)$. Moreover, in this case one has*

$$\int_Y g(y)(f_{\#}\mu)(dy) = \int_X g(f(x))\mu(dx).$$

Proof. See Theorem 3.6.1 in [Bogachev & Ruas \(2007\)](#). □

Lemma A.2. *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ be a Borel function and μ a Borel probability measure on $\mathbb{R}^d$. Then $f = 0$ μ -almost surely if and only if $\int_{\mathbb{R}^d} \|f(x)\| \mu(dx) = 0$.*

Proof. $\int_{\mathbb{R}^d} \|f(x)\| \mu(dx) = 0$ if and only if $\|f(x)\| = 0$ μ -almost surely (see, for instance, Proposition 2.16 in [Folland \(1999\)](#)), which is the case if and only if $f(x)$ is the zero vector μ -almost surely. □

Lemma A.3. *Let $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$ for some $p \geq 1$. If $f \in B(\mu, \nu)$ and $\tilde{f}$ is a (μ, ν) -inverse of f , then*

- (i) $\|f\| \in L^p(\mu)$,
- (ii) $\tilde{f}$ is a ν -a.s. unique (μ, ν) -inverse of f ,
- (iii) $\tilde{f} \in B(\nu, \mu)$, $\|\tilde{f}\| \in L^p(\nu)$ and f is a μ -a.s. unique (ν, μ) -inverse of $\tilde{f}$.

Proof. (i) Let $f \in B(\mu, \nu)$. Then by Lemma A.1,

$$\int_{\mathbb{R}^d} \|f(x)\|^p \mu(dx) = \int_{\mathbb{R}^d} \|y\|^p \nu(dy) < \infty.$$

(ii) Assume $\bar{f}$ is another (μ, ν) -inverse of f , then

$$\begin{aligned} 0 &= \int_{\mathbb{R}^d} \|(\tilde{f} \circ f)(x) - x\| \mu(dx) && (\tilde{f} \circ f = \text{id}, \mu\text{-a.s.}) \\ &= \int_{\mathbb{R}^d} \|(\tilde{f} \circ f)(x) - (\bar{f} \circ f)(x)\| \mu(dx) && (\bar{f} \circ f = \text{id}, \mu\text{-a.s.}) \\ &= \int_{\mathbb{R}^d} \|\tilde{f}(y) - \bar{f}(y)\| \nu(dy), && (\text{change of variable } y = f(x), \text{ see Lemma A.1}) \end{aligned}$$

which, by Lemma A.2, implies that $\tilde{f} = \bar{f}$ ν -a.s.

(iii) We first show that $\tilde{f}_{\#}\nu = \mu$. Let $A \in \mathcal{B}(\mathbb{R}^d)$, then

$$\begin{aligned} \nu(\tilde{f}^{-1}(A)) &= \mu(f^{-1}(\tilde{f}^{-1}(A))) && (f_{\#}\mu = \nu) \\ &= \mu((\tilde{f} \circ f)^{-1}(A)) \\ &= \mu(A). && (\tilde{f} \circ f = \text{id}, \mu\text{-a.s.}) \end{aligned}$$

Since $\tilde{f}$ is the (μ, ν) -inverse of f , clearly f satisfies the definition of a (ν, μ) -inverse of $\tilde{f}$, therefore $\tilde{f}$ belongs to $B(\nu, \mu)$. Finally, $\|f\| \in L^p(\nu)$ simply follows from $\tilde{f} \in B(\nu, \mu)$ and (i). □

Lemma A.4. *Let $\lambda, \mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$. If $f \in B(\lambda, \mu)$ and $g \in B(\mu, \nu)$, then $g \circ f \in B(\lambda, \nu)$. Furthermore, if $\tilde{f}$ is a (λ, μ) -inverse of f and $\tilde{g}$ is a (μ, ν) -inverse of g , then $\tilde{f} \circ \tilde{g}$ is a (λ, ν) -inverse of $g \circ f$.*

Proof. Clearly $(g \circ f)_\# \lambda = g_\#(f_\# \lambda) = g_\# \mu = \nu$. Furthermore,

$$\begin{aligned}
 0 &= \int_{\mathbb{R}^d} \|(\tilde{f} \circ \tilde{g})(y) - (\tilde{f} \circ \tilde{g})(y)\| \nu(dy) \\
 &= \int_{\mathbb{R}^d} \|(\tilde{f} \circ \tilde{g} \circ g \circ \tilde{g})(y) - (\tilde{f} \circ \tilde{g})(y)\| \nu(dy) && (g \circ \tilde{g} = \text{id } \nu\text{-a.s.}) \\
 &= \int_{\mathbb{R}^d} \|(\tilde{f} \circ \tilde{g} \circ g)(x) - \tilde{f}(x)\| \mu(dx) && (\text{change of variables } x = \tilde{g}(y)) \\
 &= \int_{\mathbb{R}^d} \|(\tilde{f} \circ \tilde{g} \circ g \circ f \circ \tilde{f})(x) - \tilde{f}(x)\| \mu(dx) && (f \circ \tilde{f} = \text{id } \mu\text{-a.s.}) \\
 &= \int_{\mathbb{R}^d} \|(\tilde{f} \circ \tilde{g} \circ g \circ f)(z) - z\| \lambda(dz), && (\text{change of variables } z = \tilde{f}(x))
 \end{aligned}$$

which shows that $\tilde{f} \circ \tilde{g} \circ g \circ f = \text{id}$ λ -almost surely by Lemma A.2. An analogous computation shows that $g \circ f \circ \tilde{f} \circ \tilde{g} = \text{id}$ ν -almost surely. $\square$

B. Proofs

Proof of Lemma 3.1. Existence of a unique solution $\gamma \in \Gamma(\mu, \nu)$ to Kantorovich's problem (2) and the fact that γ is of the form $(\text{id}, T)_\# \mu$ for a measurable map $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ pushing μ to ν , follows from Theorem 3.7 of Gangbo & McCann (1996). That T is μ -almost surely unique and belongs to $B(\mu, \nu)$ is a consequence of Theorem 4.5 of Gangbo & McCann (1996). $\square$

The proofs of Theorem 3.2 and Theorem 3.3 rely on the following lemma.

Lemma B.1. Consider probability measures $\lambda, \mu, \nu \in \mathcal{P}_{p,ac}(\mathbb{R}^d)$ for some $p > 1$. Then $B(\lambda, \mu)$ is non-empty, and for every $f \in B(\lambda, \mu)$, there exists a $g \in B(\lambda, \nu)$ such that

$$\mathbb{W}_p^p(\mu, \nu) = \int_{\mathbb{R}^d} \|f(z) - g(z)\|^p \lambda(dz) = \int_{\mathbb{R}^d} \|x - g \circ \tilde{f}(x)\|^p \mu(dx), \quad (5)$$

where $\tilde{f}$ is any (λ, μ) -inverse of f . In particular, $g \circ \tilde{f}$ is an optimal transport map between μ and ν .

Proof. It follows from Theorems 3.7 and 4.5 of Gangbo & McCann (1996) that $B(\lambda, \mu)$ is non-empty and there exists a $T \in B(\mu, \nu)$ such that

$$\int_{\mathbb{R}^d} \|x - T(x)\|^p \mu(dx) = \inf_{\pi \in \Pi(\mu, \nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^p \pi(dx, dy) = \mathbb{W}_p^p(\mu, \nu).$$

So, for a given $f \in B(\lambda, \mu)$, the mapping $g = T \circ f$ is in $B(\lambda, \nu)$ by Lemma A.4 and, by the change-of-variable-formula,

$$\int_{\mathbb{R}^d} \|f(z) - g(z)\|^p \lambda(dz) = \int_{\mathbb{R}^d} \|f(z) - T \circ f(z)\|^p \lambda(dz) = \int_{\mathbb{R}^d} \|x - T(x)\|^p \mu(dx) = \mathbb{W}_p^p(\mu, \nu).$$

Since $T = g \circ \tilde{f}$ μ -almost surely, this shows (5), and $g \circ \tilde{f}$ is an optimal transport map between μ and ν , which completes the proof of the lemma. $\square$

Proof of Theorem 3.2. We know from Lemma 3.1 that there exists an optimal transport map $T \in B(\mu, \nu)$. So, for any $f \in B(\lambda, \mu)$ and $g \in B(\lambda, \nu)$, we have

$$\|f - g\|_{L^p(\lambda)}^p = \int_{\mathbb{R}^d} \|x - g \circ \tilde{f}(x)\|^p \mu(dx) \geq \int_{\mathbb{R}^d} \|x - T(x)\|^p \mu(dx) = \mathbb{W}_p^p(\mu, \nu), \quad (6)$$

where $\tilde{f}$ is any (λ, μ) -inverse of f and we used the fact that $g \circ \tilde{f}$ pushes μ to ν by Lemma A.4.

On the other hand, it follows from Lemma B.1 that there exist $f \in B(\lambda, \mu)$ and $g \in B(\lambda, \nu)$ such that the inequality in (6) becomes an equality, which completes the proof. $\square$

Proof of Theorem 3.3. It follows from Brizzi et al. (2025) that the w -weighted Wasserstein-2 barycenter $\bar{\mu}$ of $(\mu_s)_{s \in \mathcal{S}}$ belongs to $\mathcal{P}_{2,ac}(\mathbb{R}^d)$. Moreover, we obtain from the definitions of the barycenter $\bar{\mu}$ and the Wasserstein-2 distance that

$$\sum_{s \in \mathcal{S}} w_s \mathbb{W}_2^2(\bar{\mu}, \mu_s) \leq \sum_{s \in \mathcal{S}} w_s \mathbb{W}_2^2(h_{\#}\lambda, \mu_s) \leq \sum_{s \in \mathcal{S}} w_s \int_{\mathbb{R}^d} \|h(z) - f(z, s)\|^2 \lambda(dz)$$

for all Borel maps $h : \mathbb{R}^d \rightarrow \mathbb{R}^d$ and $f : \mathbb{R}^d \times \mathcal{S} \rightarrow \mathbb{R}^d$ such that $f(\cdot, s)$ transports λ to μ_s for every $s \in \mathcal{S}$. On the other hand, we know from Proposition B.1 that there exist $h \in B(\lambda, \bar{\mu})$ and $f(\cdot, s) \in B(\lambda, \mu_s)$, $s \in \mathcal{S}$, such that

$$\mathbb{W}_2^2(\bar{\mu}, \mu_s) = \int_{\mathbb{R}^d} \|h(z) - f(z, s)\|^2 \lambda(dz) \quad \text{for all } s \in \mathcal{S},$$

and therefore,

$$\sum_{s \in \mathcal{S}} w_s \mathbb{W}_2^2(\bar{\mu}, \mu_s) = \sum_{s \in \mathcal{S}} w_s \mathbb{W}_2^2(h_{\#}\lambda, \mu_s) = \sum_{s \in \mathcal{S}} w_s \int_{\mathbb{R}^d} \|h(z) - f(z, s)\|^2 \lambda(dz).$$

In particular, h minimizes

$$\sum_{s \in \mathcal{S}} w_s \int_{\mathbb{R}^d} \|h(z) - f(z, s)\|^2 \lambda(dz) = \mathbb{E} [\|h(Z) - f(Z, S)\|^2]$$

over all Borel maps from $\mathbb{R}^d$ to $\mathbb{R}^d$, which implies that it is given by the component-wise conditional expectation

$$h(Z) = \mathbb{E}[f(Z, S) \mid Z] = \sum_{s \in \mathcal{S}} w_s f(Z, s);$$

see, e.g. Durrett (2019). Moreover, f minimizes

$$\sum_{i=1}^d \mathbb{E} [(\mathbb{E}[f_i(Z, S) \mid Z] - f_i(Z, S))^2] = \sum_{i=1}^d \mathbb{E} [\text{Var}(f_i(Z, S) \mid Z)]$$

over all mappings $f : \mathbb{R}^d \times \mathcal{S} \rightarrow \mathbb{R}^d$ satisfying $f(\cdot, s) \in B(\lambda, \mu_s)$ for all $s \in \mathcal{S}$. This proves (i).

Now, assume $\tilde{f} : \mathbb{R}^d \times \mathcal{S} \rightarrow \mathbb{R}^d$ is another function such that $\tilde{f}(\cdot, s) \in B(\lambda, \mu_s)$ for all $s \in \mathcal{S}$ and

$$\sum_{i=1}^d \mathbb{E} [\text{Var}(\tilde{f}_i(Z, S) \mid Z)] = \sum_{i=1}^d \mathbb{E} [\text{Var}(f_i(Z, S) \mid Z)].$$

Then, for $\tilde{h}(Z) = \mathbb{E}[\tilde{f}(Z, S) \mid Z] = \sum_{s \in \mathcal{S}} w_s \tilde{f}(Z, s)$, one has

$$\begin{aligned} \sum_{s \in \mathcal{S}} w_s \mathbb{W}_2^2(\tilde{h}_{\#}\lambda, \mu_s) &\leq \sum_{s \in \mathcal{S}} w_s \int_{\mathbb{R}^d} \|\tilde{h}(z) - \tilde{f}(z, s)\|^2 \lambda(dz) = \sum_{i=1}^d \mathbb{E} [\text{Var}(\tilde{f}_i(Z, S) \mid Z)] \\ &= \sum_{i=1}^d \mathbb{E} [\text{Var}(f_i(Z, S) \mid Z)] = \sum_{s \in \mathcal{S}} w_s \int_{\mathbb{R}^d} \|h(z) - f(z, s)\|^2 \lambda(dz) = \sum_{s \in \mathcal{S}} w_s \mathbb{W}_2^2(\bar{\mu}, \mu_s). \end{aligned}$$

But since $\bar{\mu}$ is the unique w -weighted Wasserstein-2 barycenter, we conclude that $\tilde{h}_{\#}\lambda = \bar{\mu}$, which shows (ii). $\square$

C. Background on Normalizing Flows

Our method relies on conditional normalizing flows to compute OT maps and Wasserstein barycenters. For the reader’s convenience, we collect here all the necessary background on normalizing flows. For more information, we point the reader to [Kobyzev et al. \(2020\)](#) and [Papamakarios et al. \(2021\)](#), which are two recent, well-written survey papers.

Normalizing flows are generative models used to learn high-dimensional data distributions by transforming a simple latent distribution (e.g. a standard normal distribution) through a sequence of diffeomorphisms. A key advantage of normalizing flows is that, unlike other generative models such as VAEs and GANs, density evaluations of the model distribution are efficient, thanks to the change of variables formula, which allows efficient learning by likelihood maximization. More precisely, if Z is an $\mathbb{R}^d$ -valued random variable with density p_Z and $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a composition of diffeomorphisms $f = f_1 \circ \dots \circ f_N$, then the random variable $X = f(Z)$ has density:

$$p(x) = p_Z(f^{-1}(x)) |\det(D_x f^{-1}(x))|, \quad (7)$$

where $f^{-1} = f_N^{-1} \circ \dots \circ f_1^{-1}$ and

$$|\det(D_x f^{-1}(x))| = \prod_{i=1}^N |\det(D_x f_i^{-1}(x))|.$$

The challenge in designing good normalizing flows consists in finding flows that have a tractable Jacobian and are easy to invert, but are expressive enough to be able to approximate interesting data distributions.

Real NVP. Real NVP was the first normalizing flow model with competitive performance on benchmark image data sets ([Dinh et al., 2016](#)). It leverages coupling layers, first introduced by [Dinh et al. \(2014\)](#), to provide an economical way of representing highly expressive transformations. In a coupling layer the input is first split into two components $z = (z^A, z^B)$, then z^A is transformed using a simple parametric bijective transformation, whose parameters depend (possibly in a highly non-linear way) on z^B , as follows:

$$\begin{cases} x^A &= f_{\theta(z^B)}(z^A) \\ x^B &= z^B \end{cases}$$

where $f_{\theta}(\cdot)$ is a parametric bijection with parameters θ , called a *coupling function*. In particular, Real NVP uses affine coupling layers with log-scale and shift parameters, $\theta(z^B) = (a(z^B), b(z^B))$, resulting in a coupling function of the form:

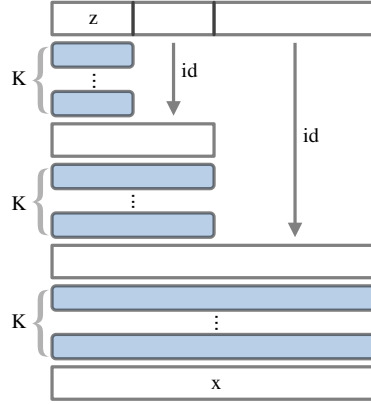
$$f_{\theta(z^B)}(z^A) = z^A \odot \exp(a(z^B)) + b(z^B),$$

where $a(\cdot)$ and $b(\cdot)$ are feedforward or convolutional neural networks.

Stacking many affine coupling layers and permutation layers (i.e. bijective transformations $z \mapsto \pi(z)$, for a random permutation π fixed at initialization) results in highly complex, non-linear transformations.

Glow. [Kingma & Dhariwal \(2018\)](#) proposed a normalizing flow model with improved performance on image data. They replaced the random permutation layers of Real NVP with invertible 1x1 convolutions, thereby allowing the model to learn the permutations, instead of randomly fixing them at initialization. They also added activation normalization layers, which adapt the batch normalization technique to the small mini-batch regime typical of training on high-resolution image data.

Multi-scale architecture. A common problem of normalizing flows is that the dimension of the latent space must match the dimension of the target distribution to be learned, due to the bijectivity constraint. In high-dimensional settings, this results in very deep and large models, that may be difficult to train. Multi-scale architectures solve this problem by exploiting the fact that empirical data distributions, especially for image data, tend to concentrate around low dimensional manifolds ([Gong et al., 2019](#); [Pope et al., 2021](#); [Brown et al., 2022](#); [Loaiza-Ganem et al., 2024](#)). This is done by dividing the model’s flows into L separate *scales* or *levels*, as shown in Figure 10, consisting of K layers each, with latent dimensions being gradually added at each subsequent scale. In this way, only a low-dimensional latent vector z is pushed through the full depth of the model, while the remaining dimensions undergo simpler, easier-to-train transformations, capturing finer details.


 Figure 10. Multi-scale architecture with $L = 3$ scales of K flows each.

D. Experimental details

Table 7 presents all hyperparameter choices for our model in all numerical experiments. The architecture type (either conditional Real NVP or conditional Glow) reports the number of hidden neurons in each hidden layer of the conditioner network. L and K denote the number of scales and flows per scale in the multi-scale architecture.

Table 7. Hyperparameter choices for our model for all numerical experiments.

DATASET	OPT	LR	ITERS	BS	TYPE		L	K	WEIGHTS
SWISS ROLL	ADAM	10^{-4}	$5 \cdot 10^3$	10^4	REAL [64, 64]	NVP	1	32	LOGSPACE [0, -4]
HIGH-DIM GAUSSIAN ($d = 2 - 128$)	ADAM	10^{-3}	10^4	10^4	REAL [64, 64]	NVP	$\log_2(d)$	$\begin{cases} 32 & \text{FOR } d = 2 \\ 16 & \text{FOR } d = 4, 8, 16 \\ 8 & \text{FOR } d \geq 32 \end{cases}$	LOGSPACE [0, -2]
HIGH-DIM UNIFORM ($d = 2 - 128$)	ADAM	10^{-4}	10^4	10^4	REAL [64, 64]	NVP	$\log_2(d)$	$\begin{cases} 32 & \text{FOR } d = 2 \\ 16 & \text{FOR } d = 4, 8, 16 \\ 8 & \text{FOR } d \geq 32 \end{cases}$	LOGSPACE [0, -2]
MNIST	ADAM	10^{-4}	$5 \cdot 10^4$	32	GLOW, CHANNELS	256	4	16	LOGSPACE [1, -2]
LARGE NUMBER OF IN- PUT DISTRIBUTIONS	ADAM	10^{-3}	10^4	10^3	REAL [64, 64]	NVP	$\log_2(d)$	32	LOGSPACE [0, -2]

All numerical experiments were run on an NVIDIA GeForce RTX 4090 GPU with 24 GB of memory, except for the MNIST data set experiment, which was run on an NVIDIA RTX 6000 Ada with 48 GB of memory. Our implementation is written in Python, is both GPU and CPU-compatible, and builds on PyTorch, the normflows package by [Stimper et al. \(2023\)](#) and the code repository by [Korotin et al. \(2021b\)](#).

The training times for the location-scatter experiments in Section 5.2.2 are shown in Table 8 and Table 9. We emphasize, though, that in practice all methods converge in approximately 10-20 minutes, with longer times needed only for top-notch performance. On the MNIST data set our method converges in approximately 30 minutes.

Table 8. Training times on high-dimensional location-scatter Gaussian data.

METHOD	$d = 2$	$d = 4$	$d = 8$	$d = 16$	$d = 32$	$d = 64$	$d = 128$
SCW ₂ B	114M 9s	110M51s	108M21s	108M 4s	108M52s	109M36s	114M56s
WIN	60M 9s	57M56s	58M32s	75M53s	57M26s	92M54s	93M28s
OURS	23M32s	23M58s	35M28s	47M 7s	30M 1s	35M59s	42M 3s

Table 9. Training times on high-dimensional location-scatter Uniform data.

METHOD	$d = 2$	$d = 4$	$d = 8$	$d = 16$	$d = 32$	$d = 64$	$d = 128$
SCW ₂ B	114M 9s	110M51s	108M21s	108M 4s	108M52s	109M36s	114M56s
WIN	90M17s	58M55s	59M45s	60M45s	60M 8s	93M16s	103M13s
OURS	110M 1s	114M36s	110M46s	111M14s	110M11s	108M55s	111M49s