GENERATIVE MODELING WITH OPTIMAL TRANSPORT MAPS

Anonymous authors
Paper under double-blind review

ABSTRACT

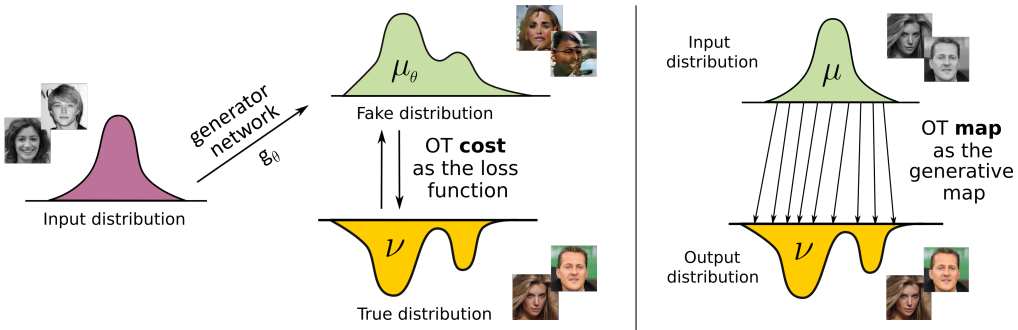
With the discovery of Wasserstein GANs, Optimal Transport (OT) has become a powerful tool for large-scale generative modeling tasks. In these tasks, OT cost is typically used as the loss for training GANs. In contrast to this approach, we show that the OT map itself can be used as a generative model, providing comparable performance. Previous analogous approaches consider OT maps as generative models only in the latent spaces due to their poor performance in the original high-dimensional ambient space. In contrast, we apply OT maps directly in the ambient space, e.g., a space of high-dimensional images. First, we derive a min-max optimization algorithm to efficiently compute OT maps for the quadratic cost (Wasserstein-2 distance). Next, we extend the approach to the case when the input and output distributions are located in the spaces of different dimensions and derive error bounds for the computed OT map. We evaluate the algorithm on image generation and unpaired image restoration tasks. In particular, we consider denoising, colorization, and inpainting, where the optimality of the restoration map is a desired attribute, since the output (restored) image is expected to be close to the input (degraded) one.

1 INTRODUCTION

Since the discovery of Generative Adversarial Networks (GANs, Goodfellow et al. (2014)), there has been a surge in generative modeling (Radford et al., 2016; Arjovsky et al., 2017; Brock et al., 2019; Karras et al., 2019). In the past few years, Optimal Transport (OT, Villani (2008)) theory has been pivotal in addressing important issues of generative models. In particular, the usage of Wasserstein distance has improved diversity (Arjovsky et al., 2017; Gulrajani et al., 2017), convergence (Sanjabi et al., 2018), and stability (Miyato et al., 2018; Kim et al., 2021) of GANs.

Generative models based on OT can be split into two classes depending on what OT is used for. First, the **optimal transport cost serves as the loss** for generative models, see Figure 1a. This is the most prevalent class of methods which includes WGAN (Arjovsky et al., 2017) and its modifications: WGAN-GP (Gulrajani et al., 2017), WGAN-LP (Petzka et al., 2018), and WGAN-QC (Liu et al., 2019). Second, the **optimal transport map is used as a generative model** itself, see Figure 1b. Such approaches include LSOT (Seguy et al., 2018), AE-OT (An et al., 2020a), ICNN-OT (Makkuva et al., 2020), W2GN (Korotin et al., 2021a). Models of the first class have been well-studied, but limited attention has been paid to the second class. Existing approaches of the second class primarily consider **OT maps in latent spaces** of pre-trained autoencoders (AE), see Figure 3. The performance of such generative models depends on the underlying AEs, in which decoding transformations are often not accurate; as a result this deficiency limits practical applications in high-dimensional ambient spaces. For this reason, using OT in the latent space does not necessarily guarantee superior performance in generative modeling.

The focus of our paper is the second class of OT-based models using OT map as the generative map. Finding an optimal mapping is motivated by its ability to preserve specific attributes of the input samples, a desired property in unpaired learning. For example, in unpaired image-to-image translation, the learner has to fit a map between two data distributions which preserves the image content. CycleGAN-based models (Zhu et al., 2017) are widely used for this purpose. However, they typically have complex optimization objectives consisting of several losses (Amodio & Krishnaswamy, 2019; Lu et al., 2019) in order to make the fitted map preserve the required attributes.



(a) OT cost as the loss for the generative model.

(b) OT map as the generative model.

Figure 1: Two existing approaches to use optimal transport in generative models.

The main contributions of this paper are as follows:

1. We propose an end-to-end algorithm (§4.3) to fit OT maps for the quadratic cost (Wasserstein-2 distance) between distributions located on the spaces of equal dimensions (§4.1) and extend the method to unequal dimensions as well (§4.2). We prove error bounds for the method (§4.4).
2. We demonstrate large-scale applications of OT maps in popular computer vision tasks. We consider image generation (§5.1) and unpaired image restoration (§5.2) tasks.

Our strict OT-based framework allows the theoretical analysis of the recovered transport map. The OT map obtained by our method can be directly used in large-scale computer vision problems which is in high contrast to previous related methods relying on autoencoders and OT maps in the latent space. Importantly, the performance and computational complexity of our method is comparable to OT-based generative models using OT cost as the loss.

Notations. In what follows, $\mathcal{X}$ and $\mathcal{Y}$ are two complete metric spaces, $\mu(x)$ and $\nu(y)$ are probability distributions on $\mathcal{X}$ and $\mathcal{Y}$, respectively. For a measurable map $T : \mathcal{X} \to \mathcal{Y}$, $T_{\#}\mu$ denotes the pushforward distribution of μ, i.e., the distribution for which any measurable set $E \subset \mathcal{Y}$ satisfies $T_{\#}\mu(E) = \mu(T^{-1}(E))$. For a vector x, $\|x\|$ denotes its Euclidean norm. We use $\langle x, y\rangle$ to denote the inner product of vectors x and y. We use $\Pi(\mu, \nu)$ to denote the set of joint probability distributions on $\mathcal{X} \times \mathcal{Y}$ whose marginals are μ and ν, respectively (couplings). For a function $f : \mathbb{R}^D \to \mathbb{R} \cup \{\pm\infty\}$ its Legendre–Fenchel transform (the convex conjugate) is $\overline{f}(y) = \sup_{x\in\mathbb{R}^D}\{\langle x, y\rangle - f(x)\}$. It is convex, even if f is not.

2 Background on Optimal Transport

Consider a cost of transportation, $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ defined over the product space of $\mathcal{X}$ and $\mathcal{Y}$.

Monge's Formulation. The *optimal transport cost* between μ and ν for ground cost $c(\cdot, \cdot)$ is

$$\text{Cost}(\mu, \nu) \stackrel{\text{def}}{=} \inf_{T_{\#}\mu=\nu} \int_{\mathcal{X}} c\left(x, T(x)\right) d\mu(x), \tag{1}$$

where the infimum is taken over all measurable maps $T : \mathcal{X} \to \mathcal{Y}$ pushing μ to ν, see Figure 2. The map T^* on which the infimum in (1) is attained is called the *optimal transport map*. Monge's formulation does not allow splitting. For example, when μ is a Dirac distribution and ν is a non-Dirac distribution, the feasible set of equation (1) is empty.

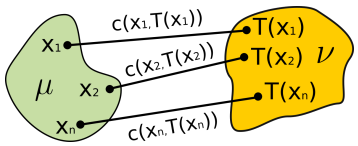


Figure 2: Monge's OT.

Kantorovich's Relaxation. Instead of asking to which particular point $y \in \mathcal{Y}$ should all the probability mass of x be moved, Kantorovich (1948) asks how the mass of x should be distributed among all $y \in \mathcal{Y}$. Formally, a transport coupling replaces a transport map; the OT cost is given by:

$$\text{Cost}(\mu, \nu) \stackrel{\text{def}}{=} \inf_{\pi\in\Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c\left(x, y\right) d\pi(x, y), \tag{2}$$

where the infimum is taken over all couplings $\pi \in \Pi(\mu, \nu)$ of μ and ν. The coupling π^* attaining the infimum of (2) is called the *optimal transport plan*. Unlike the formulation of (1), the formulation of (2) is well-posed, and with mild assumptions on spaces $\mathcal{X}, \mathcal{Y}$ and ground cost $c(\cdot, \cdot)$, the minimizer π^* of (2) always exists (Villani, 2008, Theorem 4.1). In particular, if π^* is deterministic, i.e., $\pi^* = [\mathrm{id}_{\mathcal{X}}, T^*]_{\#}\mu$ for some $T^* : \mathcal{X} \to \mathcal{Y}$, then T^* minimizes (1).

Duality. The dual form of (2) is given by (Kantorovich, 1948):

$$\mathrm{Cost}(\mu, \nu) = \sup_{(u,v)} \left\{ \int_{\mathcal{X}} u(x) d\mu(x) + \int_{\mathcal{Y}} v(y) d\nu(y) \colon u(x) + v(y) \leq c(x, y) \right\}, \tag{3}$$

with $u \in L^1(\mu)$, $v \in L^1(\nu)$ called Kantorovich *potentials*. For $u : \mathcal{X} \to \mathbb{R}$ and $v : \mathcal{Y} \to \mathbb{R}$ define their c-transforms by $u^c(y) = \inf_{x \in \mathcal{X}} \{c(x, y) - u(x)\}$ and $v^c(x) = \inf_{y \in \mathcal{Y}} \{c(x, y) - v(y)\}$ respectively. Using c-transform, (3) is reformulated as (Villani, 2008, §5)

$$\mathrm{Cost}(\mu, \nu) = \sup_{v} \left\{ \int_{\mathcal{X}} v^c(x) d\mu(x) + \int_{\mathcal{Y}} v(y) d\nu(y) \right\} = \sup_{u} \left\{ \int_{\mathcal{X}} u(x) d\mu(x) + \int_{\mathcal{Y}} u^c(y) d\nu(y) \right\}. \tag{4}$$

Primal-dual relationship. For certain ground costs $c(\cdot, \cdot)$, the primal solution T^* of (1) can be recovered from the dual solution u^* of (3). For example, if $\mathcal{X} = \mathcal{Y} = \mathbb{R}^D$, $c(x, y) = h(x - y)$ with strictly convex $h : \mathbb{R}^D \to \mathbb{R}$ and μ is absolutely continuous supported on the compact set, then

$$T^*(x) = x - (\nabla h)^{-1}\big(u^*(x)\big), \tag{5}$$

see (Santambrogio, 2015, Theorem 1.17). For general costs, see (Villani, 2008, Theorem 10.28).

3 Optimal Transport in Generative Models

Optimal transport cost as the loss (Figure 1a). Starting with the works of Arjovsky & Bottou (2017); Arjovsky et al. (2017), the usage of OT cost as the loss has become a major way to apply OT for generative modeling. In this setting, given data distribution ν and fake distribution μ_θ, the goal is to minimize $\mathrm{Cost}(\mu_\theta, \nu)$ w.r.t. the parameters θ. Typically, μ_θ is a pushforward distribution of some given distribution, e.g., $\mathcal{N}(0, I)$, via generator network G_θ.

The *Wasserstein-1* distance ($\mathcal{W}_1$), i.e., the transport cost for ground cost $c(x, y) = \|x - y\|$, is the most practically prevalent example of such a loss. Models based on this loss are known as Wasserstein GANs (WGANs). They estimate $\mathcal{W}_1(\mu_\theta, \nu)$ based on the dual form as given by (4). For $\mathcal{W}_1$, the optimal potentials u^*, v^* of (4) satisfy $u^* = -v^*$ where u^* is a 1-Lipschitz function (Villani, 2008, Case 5.16). As a result, to compute $\mathcal{W}_1$, one needs to optimize the following simplified form:

$$\mathcal{W}_1(\mu_\theta, \nu) = \sup_{\|u\|_L \leq 1} \left\{ \int_{\mathcal{X}} u(x) d\mu_\theta(x) - \int_{\mathcal{Y}} u(y) d\nu(y) \right\}. \tag{6}$$

In WGANs, the potential u is called the *discriminator*. Optimization of (6) reduces constrained optimization of (4) with two potentials u, v to optimization of only one discriminator u. In practice, enforcing the Lipschitz constraint on u is challenging. Most methods to do this are regularization-based, e.g., they use gradient penalty (Gulrajani et al., 2017, WGAN-GP) and Lipschitz penalty (Petzka et al., 2018, WGAN-LP). Other methods enforce Lipschitz property via incorporating certain hard restrictions on the discriminator's architecture (Anil et al., 2019; Tanielian & Biau, 2021).

General transport costs (other than $\mathcal{W}_1$) can also be used as the loss for generative models. They are less popular since they do not have a dual form reducing to a single potential function similar to (6) for $\mathcal{W}_1$. Consequently, the challenging estimation of the c-transform u^c is needed. To avoid this, Sanjabi et al. (2018) consider the dual form of (3) with two potentials u, v instead form (4) with one u and softly enforce the condition $u(x) + v(y) \leq c(x, y)$ via entropy or quadratic regularization. Nhan Dam et al. (2019) use the dual form of (4) and amortized optimization to compute u^c via an additional neural network. Both methods work for general $c(\cdot, \cdot)$, though the authors test them for $c(x, y) = \|x - y\|$ only, i.e., $\mathcal{W}_1$ distance. Mallasto et al. (2019) propose a fast way to approximate the c-transform and test the approach (WGAN-(q, p)) with several costs, in particular, the *Wasserstein-2* distance ($\mathcal{W}_2$), i.e., the transport cost for the quadratic ground cost $c(x, y) = \frac{1}{2}\|x - y\|^2$. Specifically for $\mathcal{W}_2$, Liu et al. (2019) approximate the c-transform via a linear program (WGAN-QC).

A fruitful branch of OT-based losses for generative models comes from modified versions of OT cost, such as Sinkhorn (Genevay et al., 2018), sliced (Deshpande et al., 2018) and minibatch (Fatras et al., 2019) OT distances. They typically have lower sample complexity than usual OT and can be accurately estimated from random mini-batches without using dual forms such as (3). In practice, these approaches usually learn the ground OT cost $c(\cdot, \cdot)$.

The aforementioned methods use OT cost in the ambient space to train GANs. There also exist approaches using OT cost in the latent space. For example, Tolstikhin et al. (2017); Patrini et al. (2020) use OT cost between encoded data and a given distribution as an additional term to reconstruction loss for training an AE. As the result, AE's latent distribution becomes close to the given one.

OPTIMAL TRANSPORT MAP AS THE GENERATIVE MAP (Figure 1b). Methods to compute the OT map (plan) are less common in comparison to those computing the cost. Recovering the map from the primal form (1) or (2) usually yields complex optimization objectives containing several adversarial terms (Xie et al., 2019; Liu et al., 2021; Lu et al., 2020). Such procedures require careful hyperparameter choice. This needs to be addressed before using these methods in practice.

Primal-dual relationship (§2) makes it possible to recover the OT map via solving the dual form (3). Dual-form based methods primarily consider $\mathcal{W}_2$ cost due to its nice theoretical properties and relation to convex functions (Brenier, 1991). In the semi-discrete case (μ is continuous, ν is discrete), An et al. (2020a) and Lei et al. (2019) compute the dual potential and the OT map by using the Alexandrov theory and convex geometry. For the continuous case, Seguy et al. (2018) use the entropy (quadratic) regularization to recover the dual potentials and extract OT map from them via the barycenteric projection. Taghvaei & Jalali (2019), Makkuva et al. (2020), Korotin et al. (2021a) employ input-convex neural networks (ICNNs, see Amos et al. (2017)) to parametrize potentials in the dual problem and recover OT maps by using their gradients.

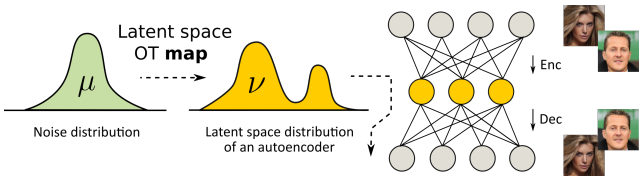


Figure 3: The existing most prevalent approach to use OT maps in generative models.

The aforementioned dual form methods compute OT maps in **LATENT SPACES** for problems such as domain adaptation and latent space mass transport, see Figure 3. OT maps in high-dimensional ambient spaces, e.g., natural images, are usually not considered. Recent evaluation of continuous OT methods for $\mathcal{W}_2$ (Korotin et al., 2021b) reveals their crucial limitations, which negatively affect their scalability, such as poor expressiveness of ICNN architectures or bias due to regularization.

4 END-TO-END SOLUTION TO LEARN OPTIMAL MAPS

4.1 EQUAL DIMENSIONS OF INPUT AND OUTPUT DISTRIBUTIONS

In this section, we use $\mathcal{X} = \mathcal{Y} = \mathbb{R}^D$ and consider the Wasserstein-2 distance ($\mathcal{W}_2$), i.e., the optimal transport for the quadratic ground cost $c(x, y) = \frac{1}{2}\|x - y\|^2$. We use the dual form (4) to derive a saddle point problem the solution of which yields the OT map T^*. We consider distributions μ, ν with finite second moments. We assume that for distributions μ, ν in view there *exists* a *unique* OT plan π^* minimizing (3) and it is deterministic, i.e., $\pi^* = [\text{id}_{\mathbb{R}^D}, T^*]_{\#}\mu$. Here T^* is an OT map which minimizes (1). Previous related works (Makkuva et al., 2020; Korotin et al., 2021a) assumed the absolute continuity of μ, which implied the existence and uniqueness of T^* (Brenier, 1991).

Let $\psi(y) \stackrel{\text{def}}{=} \frac{1}{2}\|y\|^2 - v(y)$, where v is the potential of (4). Note that

$$v^c(x) = \inf_{y \in \mathbb{R}^D} \left\{ \frac{1}{2}\|x - y\|^2 - v(y) \right\} = \frac{1}{2}\|x\|^2 - \sup_{y \in \mathbb{R}^D} \{\langle x, y \rangle - \psi(y)\} = \frac{1}{2}\|x\|^2 - \overline{\psi}(x). \quad (7)$$

Therefore, (4) is equivalent to

$$\mathcal{W}_2^2(\mu, \nu) = \int_{\mathcal{X}} \frac{\|x\|^2}{2} d\mu(x) + \int_{\mathcal{Y}} \frac{\|y\|^2}{2} d\nu(x) + \sup_{\psi} \left\{ - \int_{\mathcal{X}} \overline{\psi}(x) d\mu(x) - \int_{\mathcal{Y}} \psi(y) d\nu(y) \right\} = \quad (8)$$

$$\text{Constant}(\mu,\nu) - \inf_{\psi}\left\{\int_{\mathcal{X}} \overline{\psi}(x)d\mu(x) + \int_{\mathcal{Y}} \psi(y)d\nu(y)\right\} = \quad (9)$$

$$\text{Constant}(\mu,\nu) - \inf_{\psi}\left\{\int_{\mathcal{X}} \sup_{y\in\mathbb{R}^D}\left\{\langle x,y\rangle - \psi(y)\right\} d\mu(x) + \int_{\mathcal{Y}} \psi(y)d\nu(y)\right\} = \quad (10)$$

$$\text{Constant}(\mu,\nu) - \inf_{\psi}\left\{\sup_{T}\int_{\mathcal{X}} \left\{\langle x,T(x)\rangle - \psi\big(T(x)\big)\right\} d\mu(x) + \int_{\mathcal{Y}} \psi(y)d\nu(y)\right\} \quad (11)$$

where between lines (10) and (11) we replace the optimization over $y \in \mathbb{R}^D$ with the equivalent optimization over functions $T : \mathbb{R}^D \to \mathbb{R}^D$. The equivalence follows from the interchange between the integral and the supremum (Rockafellar, 1976, Theorem 3A). Thanks to the following lemma, we may solve saddle point problem (11) and obtain the OT map T^* from its solution (ψ^*, T^*).

Lemma 4.1. *Let T^* be the OT map from μ to ν. Then, for every optimal potential ψ^*,*

$$T^* \in \arg\sup_{T}\int_{\mathcal{X}} \left\{\langle x,T(x)\rangle - \psi^*\big(T(x)\big)\right\} d\mu(x). \quad (12)$$

We prove Lemma 4.1 in Appendix A.1. For general μ, ν the $\arg\sup_T$ set for optimal ψ^* might contain not only OT map T^*, but other functions as well. Working with real-world data in experiments (§5.2), we observe that despite this issue, optimization (11) still recovers T^*.

Relation to previous works. The use of the function T to approximate the c-transform was proposed by Nhan Dam et al. (2019) to estimate the Wasserstein loss in WGANs. For $\mathcal{W}_2$, the fact that T^* is an OT map was used by Makkuva et al. (2020); Korotin et al. (2021a) who primarily assumed continuous μ, ν and reduced (11) to convex ψ and $T = \nabla\phi$ for convex ϕ. Issues with non-uniqueness of solution of (12) were softened, but using ICNNs to parametrize ψ became necessary.

Korotin et al. (2021b) demonstrated that ICNNs negatively affect practical performance of OT and tested an unconstrained formulation similar to (11). As per the evaluation, it provided the best empirical performance (Korotin et al., 2021b, §4.5). The method ⌊MM:R⌉ they consider parametrizes $\frac{1}{2}\|\cdot\|^2 - \psi(\cdot)$ by a neural network, while we directly parametrize $\psi(\cdot)$ by a neural network (§4.3).

Recent work by Fan et al. (2021) exploits formulation similar to (11) for general costs $c(\cdot,\cdot)$. While their formulation leads to a max-min scheme with general costs (Fan et al., 2021, Theorem 3), our approach gives rise to a min-max method for quadratic cost. In particular, we extend the formulation to learn OT maps between distributions in spaces with unequal dimensions, see the next subsection.

4.2 Unequal Dimensions of Input and Output Distributions

Consider the case when $\mathcal{X} = \mathbb{R}^H$ and $\mathcal{Y} = \mathbb{R}^D$ have different dimensions, i.e., $H \neq D$. In order to map the probability distribution μ to ν, a straightforward solution is to embed $\mathcal{X}$ to $\mathcal{Y}$ via *some* $Q : \mathcal{X} \to \mathcal{Y}$ and then to fit the OT map between $Q_{\#}\mu$ and ν for the quadratic cost on $\mathcal{Y} = \mathbb{R}^D$. In this case, the optimization objective becomes

$$\inf_{\psi}\sup_{T}\left\{\int_{\mathcal{X}} \left\{\langle Q(x),T\big(Q(x)\big)\rangle - \psi\big(T\big(Q(x)\big)\big)\right\} d\mu(x) + \int_{\mathcal{Y}} \psi(y)d\nu(y)\right\} \quad (13)$$

with the optimal T^* recovering the OT map from $Q_{\#}\mu$ to ν. For equal dimensions $H = D$ and the identity embedding $Q(x) \equiv x$, expression (13) reduces to optimization (11) up to a constant.

Instead of optimizing (13) over functions $T : Q(\mathcal{X}) \to \mathcal{Y}$, we propose to consider optimization directly over generative mappings $G : \mathcal{X} \to \mathcal{Y}$:

$$\mathcal{L}(\psi, G) \stackrel{\text{def}}{=} \inf_{\psi}\sup_{G}\left\{\int_{\mathcal{X}} \left\{\langle Q(x),G(x)\rangle - \psi\big(G(x)\big)\right\} d\mu(x) + \int_{\mathcal{Y}} \psi(y)d\nu(y)\right\} \quad (14)$$

Our following lemma establishes connections between (14) and OT with unequal dimensions:

Lemma 4.2. *Assume that exists a unique OT plan between $Q_{\#}\mu$ and ν and it is deterministic, i.e., $[id_{\mathbb{R}^D}, T^*]_{\#}(Q_{\#}\mu)$. Then $G^*(x) = T^*\big(Q(x)\big)$ is the OT map between μ and ν for the Q-**embedded quadratic cost** $c(x,y) = \frac{1}{2}\|Q(x) - y\|^2$. Moreover, for every optimal potential ψ^* of problem (14),*

$$G^* \in \arg\sup_{G}\int_{\mathcal{X}} \left\{\langle Q(x),G(x)\rangle - \psi^*\big(G(x)\big)\right\} d\mu(x). \quad (15)$$

We prove Lemma 4.2 in Appendix A.2 and schematically present its idea in Figure 4. Analogously to Lemma 4.1, it provides a way to compute the OT map G^* for the Q-embedded quadratic cost between distributions μ and ν by solving the saddle point problem (14). Note the situation with non-uniqueness of $\arg\sup_G$ is similar to §4.1.

Relation to previous works. In practice, learning OT maps directly between spaces of unequal dimensions was considered in the work by (Fan et al., 2021, §5.2) but only on toy examples. We demonstrate that our method works well in large-scale generative modeling tasks (§5.1). Theoretical properties of OT maps for embedded costs are studied, e.g., in (Pass, 2010; McCann & Pass, 2020).

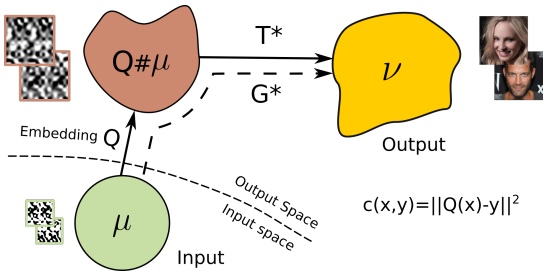


Figure 4: The scheme of our approach for learning OT maps between unequal dimensions. In the figure, the setup of §5.1 is shown: μ is a noise, Q is the bicubic upscaling, ν is a distribution of images.

4.3 Practical Aspects and Optimization Procedure

To optimize functional (14), we approximate $G : \mathbb{R}^H \to \mathbb{R}^D$ and $\psi : \mathbb{R}^D \to \mathbb{R}$ with neural networks G_θ, ψ_ω and optimize their parameters via stochastic gradient descent-ascent (SGDA) by using mini-batches from μ, ν. The practical optimization procedure is given in Algorithm 1 below. Following the usual practice in GANs, we add small gradient penalty (Gulrajani et al., 2017) on potential ψ_ω for better stability. The penalty in not included in Algorithm 1 to keep it simple.

Algorithm 1: Learning the optimal transport map between unequal dimensions.

Input : Input distribution μ on $\mathcal{X} = \mathbb{R}^H$; output distribution ν on $\mathcal{Y} = \mathbb{R}^D$;
generator network $G_\theta : \mathbb{R}^H \to \mathbb{R}^D$; potential network $\psi_\omega : \mathbb{R}^D \to \mathbb{R}$;
number of iterations per network: K_G, K_ψ; embedding $Q : \mathcal{X} \to \mathcal{Y}$;
Output: Trained generator G_θ representing OT map from μ to ν;

1 **repeat**
2 **for** $k_\psi = 1$ to K_ψ **do**
3   Draw batch $X \sim \mu$ and $Y \sim \nu$;
4   $\mathcal{L}_\psi \leftarrow \frac{1}{|Y|}\sum_{y\in Y}\psi_\omega(y) - \frac{1}{|X|}\sum_{x\in X}\psi_\omega\big(G_\theta(x)\big)$;
5   Update ω by using $\frac{\partial \mathcal{L}_\psi}{\partial \omega}$ to minimize $\mathcal{L}_\psi$;
6 **for** $k_G = 1$ to K_G **do**
7   Draw batch $X \sim \mu$;
8   $\mathcal{L}_G \leftarrow \frac{1}{|X|}\sum_{x\in X}\big[\psi\big(G(x)\big) - \langle Q(x), G_\theta(x)\rangle\big]$;
9   Update θ by using $\frac{\partial \mathcal{L}_G}{\partial \theta}$ to minimize $\mathcal{L}_G$;
10 **until** not converged;
11 **return** G_θ

Relation to previous works. WGAN by Arjovsky & Bottou (2017) uses $\mathcal{W}_1$ as the loss to update the generator while we solve a diferent task — we fit the generator G to be the OT map for Q-embedded quadratic cost. Despite this, our Algorithm 1 has similarities with WGAN's training. The update of ψ (line 4) coincides with discriminator's update in WGAN. The update of generator G (line 8) differs from WGAN's update by the term $-\langle Q(\cdot), G_\theta(\cdot)\rangle$. Besides, in WGAN the optimization is $\inf_G \sup_D$. We have $\inf_\psi \sup_G$, i.e., the generator in our case is the solution of the inner problem.

4.4 Error Analysis

Given a pair $(\hat{\psi}, \hat{G})$ approximately solving (14), a natural question to ask is how good is the recovered OT map $\hat{G}$. In this subsection, we provide a bound on the difference between G^* and $\hat{G}$ based on the duality gaps for solving outer and inner optimization problems.

In (8), and, as the result, in (10), (11), (13), (14), it is enough to consider optimization over *convex* functions ψ, see (Villani, 2008, Case 5.17). Our theorem below assumes the convexity of $\hat{\psi}$ although it might not hold in practice since in practice $\hat{\psi}$ is a neural network.

Theorem 4.3. *Assume that exists a unique deterministic OT plan for Q-embedded quadratic cost between μ and ν, i.e., $\pi^* = [id_{\mathbb{R}^H}, G^*]_{\#}\mu$ for $G^* : \mathbb{R}^H \to \mathbb{R}^D$. Assume that $\hat{\psi}$ is β-strongly convex ($\beta > 0$) and $\hat{G} : \mathbb{R}^H \to \mathbb{R}^D$. Define*

$$\epsilon_1 = \sup_G \mathcal{L}(\hat{\psi}, G) - \mathcal{L}(\hat{\psi}, \hat{G}) \qquad \textit{and} \qquad \epsilon_2 = \sup_G \mathcal{L}(\hat{\psi}, G) - \inf_{\psi} \sup_G \mathcal{L}(\psi, G)$$

Then the following bound holds true for the OT map G^* from μ to ν:

$$\frac{\text{FID}(\hat{G}_{\#}\mu, \nu)}{L^2} \leq 2 \cdot \mathcal{W}_2^2(\hat{G}_{\#}\mu, \nu) \leq \int_{\mathcal{X}} \|\hat{G}(x) - G^*(x)\|^2 d\mu(x) \leq \frac{2}{\beta}(\sqrt{\epsilon_1} + \sqrt{\epsilon_2})^2, \qquad (16)$$

where FID is the Fréchet inception distance (Heusel et al., 2017) and L is the Lipschitz constant of the feature extractor of the pre-trained InceptionV3 neural network (Szegedy et al., 2016).

We prove Theorem 4.3 in Appendix A.3. The duality gaps upper bound $L^2(\mu)$ norm between computed $\hat{G}$ and true G^* maps, and the $\mathcal{W}_2^2$ between true ν and generated (fake) distribution $\hat{G}_{\#}\mu$. Consequently, they upper bound FID between data ν and fake (generated) $\hat{G}_{\#}\mu$ distributions.

Relation to previous works. Makkuva et al. (2020); Korotin et al. (2021a) prove related bounds for $\mathcal{W}_2$ with μ, ν located on the spaces of the same dimension. Our result holds for different dimensions.

5 Experiments

We evaluate our algorithm in generative modeling of the data distribution from a noise (§5.1) and unpaired image restoration task (§5.2). Technical details are given in Appendix B. Additionally, in Appendix B.3 we test our method on toy 2D datasets and evaluate it on the Wasserstein-2 benchmark (Korotin et al., 2021b) in Appendix B.2. The code is in the supplementary material.

5.1 Modeling Data distribution from Noise Distribution

In this subsection, μ is a 128-dimensional normal noise and ν the high-dimensional data distribution.

Let the images from ν be of size $w \times h$ with c channels. As the embedding $Q : \mathcal{X} \to \mathcal{Y}$ we use a *naive* upscaling of a noise. For $x \in \mathbb{R}^{128}$ we represent it as 1-channel 16×8 image, stack c copies over channel dimension and bicubically upscale it to the size $w \times h$ of data images from ν.

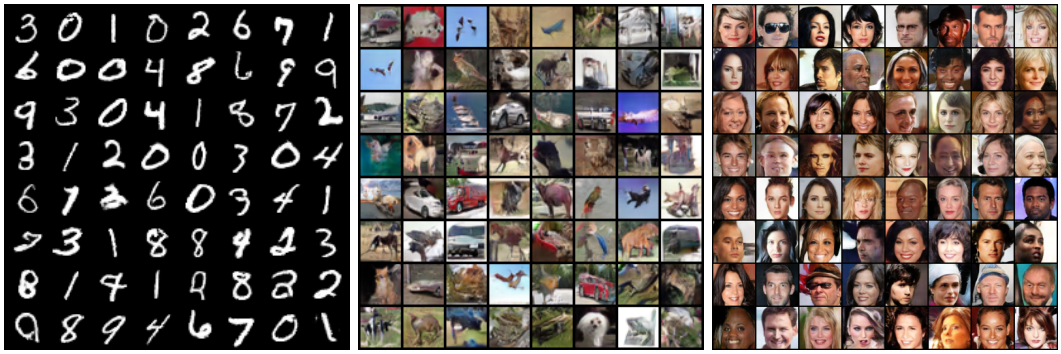
We test our method on MNIST 32×32 (LeCun et al., 1998), CIFAR10 32×32 (Krizhevsky et al., 2009), and CelebA 64×64 (Liu et al., 2015) image datasets. In Figure 5, we show random samples generated by our approach, namely **Optimal Transport Modeling** (OTM). To quantify the results, in Tables 1 and 2 we give the inception (Salimans et al., 2016) and FID (Heusel et al., 2017) scores of generated samples. Similar to (Song & Ermon, 2019, Appendix B.2), we compute them on 50K real and generated samples. Additionally, in Appendix B.3, we test our method on 128×128 CelebA faces. We provide qualitative results (images of generated faces) in Figure 11.

For comparison, we include the scores of existing generative models of three types: (1) OT map as the generative model; (2) OT cost as the loss; (3) not OT-based. Note that models of the first type compute OT in the *latent space* of an autoencoder in contrast to our approach. According to our evaluation, the performance of our method is better or comparable to existing alternatives.

5.2 Unpaired Image Restoration

In this subsection, we consider unpaired image restoration tasks on CelebA faces dataset. In this case, the input distribution μ consists of degraded images, while ν are clean images. In all the cases, embedding Q is a straightforward identity embedding $Q(x) \equiv x$.

In image restoration, optimality of the restoration map is desired since the output (restored) image is expected to be close to the input (degraded) one minimizing the transport cost. Note that GANs



(a) MNIST, 28×28, grayscale (b) CIFAR10, 32×32, RGB (c) CelebA, 64×64, RGB

Figure 5: Randomly generated MNIST, CIFAR10, and CelebA samples by our method (OTM).

Table 1: Results on CIFAR10 dataset.

Model	Related Work	Inception ↑	FID ↓
NVAE	Vahdat & Kautz (2020)	-	51.71
PixelIQN	Ostrovski et al. (2018)	5.29	49.46
EBM	Du & Mordatch (2019)	6.02	40.58
DCGAN	Radford et al. (2016)	6.64±0.14	37.70
NCSN	Song & Ermon (2019)	8.87±0.12	25.32
NCP-VAE	Aneja et al. (2021)	-	24.08
WGAN	Arjovsky et al. (2017)	-	55.2
WGAN-GP	Gulrajani et al. (2017)	6.49±0.09	39.40
3P-WGAN	Nhan Dam et al. (2019)	7.38 ± 0.08	28.8
AE-OT	An et al. (2020a)	-	28.5
AE-OT-GAN	An et al. (2020b)	-	17.1
OTM	Ours	7.41±0.11	20.69

Table 2: Results on CelebA dataset.

Model	Related Work	FID ↓
DCGAN	Radford et al. (2016)	52.0
DRAGAN	Kodali et al. (2017)	42.3
BEGAN	Berthelot et al. (2017)	38.9
NVAE	Vahdat & Kautz (2020)	13.4
NCP-VAE	Aneja et al. (2021)	5.2
WGAN	Arjovsky et al. (2017)	41.3
WGAN-GP	Gulrajani et al. (2017)	30.0
WGAN-QC	Liu et al. (2019)	12.9
AE-OT	An et al. (2020a)	28.6
W2GN+AE	Korotin et al. (2021a)	17.2
AE-OT-GAN	An et al. (2020b)	7.8
OTM	Ours	7.7

do not seek for an optimal mapping. However, in practice, due to implicit inductive biases such as convolutional architectures, GANs still tend to fit low transport cost maps (Bézenac et al., 2021).

The experimental setup is shown in Figure 6. We split the dataset in 3 parts A, B, C containing 90K, 90K, 22K samples respectively. To each image we apply the degradation transform (decolorization, noising or occlusion) and obtain the degraded dataset containing of 3 respective parts A, B, C. For *unpaired* training we use part A of degraded and part B of clean images. For testing, we use parts C.

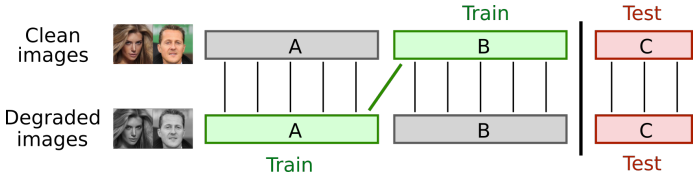


Figure 6: The training/testing scheme that we use for unpaired restoration tasks.

To quantify the results we compute FID of restored images w.r.t. clean images of part C. The scores for denoising, inpainting and colorization are given in Table 3, details of each experiment and qualitative results are given below.

Model	Denoising	Colorization	Inpainting
Input	166.59	32.12	47.65
WGAN-GP	25.49	7.75	16.51
OTM (ours)	10.95	5.66	9.96

Table 3: FID↓ on test part C in image restoration experiments.

As the baseline, we include WGAN-GP. For a fair comparison, we fit it using *exactly* the same hyperparameters as in our method. This is possible due to the similarities between our method and WGAN-GP's training procedure, see discussion in §4.3.

Denoising. To create noisy images, we add white normal noise with $\sigma = 0.3$ to each pixel. Figure 7 illustrates image denoising using our OTM approach on the test part of the dataset. We show additional qualitative results for varying σ in Figure 16 of §(B.3).

(a) Noisy samples. (b) Pushforward samples. (c) Original samples.

Figure 7: OTM for image denoising on test C part of CelebA, 64×64.

Colorization. To create grayscale images, we average the RGB values of each pixel. Figure 8 illustrates image colorization using OTM on the test part of the dataset.

(a) Grayscale samples. (b) Pushforward samples. (c) Original samples.

Figure 8: OTM for image colorization on test C part of CelebA, 64×64.

Inpainting. To create incomplete images, we replace the right half of each clean image with zeros. Figure 9 illustrates image inpainting using OTM on the test part of the dataset.

(a) Occluded samples. (b) Pushforward samples. (c) Original samples.

Figure 9: OTM for image inpainting on test C part of CelebA, 64×64.

6 Conclusion

Our method fits OT maps for the embedded quadratic transport cost between probability distributions. Unlike predecessors, it scales well to high dimensions producing applications of OT maps directly in ambient spaces, such as spaces of images. The performance is comparable to other existing generative models while the complexity of training is similar to that of popular WGANs.

Limitations. For distributions μ, ν we assume the existence of the OT map between them. In practice, this might not hold for all real-world μ, ν. Working with equal dimensions, we focus on the quadratic ground cost $\frac{1}{2}\|x - y\|^2$. Nevertheless, our approach extends to other costs $c(\cdot, \cdot)$, see Fan et al. (2021). When the dimensions are unequal, we restrict our analysis to *embedded* quadratic cost $\frac{1}{2}\|Q(x) - y\|^2$ where Q equalizes dimensions. Choosing the embedding Q might not be straightforward in some practical problems, but our evaluation (§5.1) shows that even naive choices of Q work well.

Potential impact and ethics. Real-world image restoration problems often do not have paired datasets limiting the application of supervised techniques. In these practical unpaired learning problems, we expect our optimal transport approach to improve the performance of the existing models. However, biases in data might lead to biases in the pushforward samples. This should be taken into account when using our method in practical problems.

Reproducibility. The `PyTorch` source code is provided in the supplementary material. The instructions to use the code are included in the `README.md` file.

REFERENCES

Matthew Amodio and Smita Krishnaswamy. Travelgan: Image-to-image translation by transformation vector learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8983–8992, 2019.

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In International Conference on Machine Learning, pp. 146–155. PMLR, 2017.

Dongsheng An, Yang Guo, Na Lei, Zhongxuan Luo, Shing-Tung Yau, and Xianfeng Gu. Ae-ot: A new generative model based on extended semi-discrete optimal transport. In International Conference on Learning Representations, 2020a. URL https://openreview.net/forum?id=HkldyTNYwH.

Dongsheng An, Yang Guo, Min Zhang, Xin Qi, Na Lei, and Xianfang Gu. Ae-ot-gan: Training gans from data specific latent distribution. In European Conference on Computer Vision, pp. 548–564. Springer, 2020b.

Jyoti Aneja, Alexander Schwing, Jan Kautz, and Arash Vahdat. Ncp-vae: Variational autoencoders with noise contrastive priors. In Advances in Neural Information Processing Systems Conference, 2021.

Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In International Conference on Machine Learning, pp. 291–301. PMLR, 2019.

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:1701.04862, 2017.

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

David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717, 2017.

Emmanuel de Bézenac, Ibrahim Ayed, and Patrick Gallinari. Cyclegan through the lens of (dynamical) optimal transport. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 132–147. Springer, 2021.

Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. Communications on pure and applied mathematics, 44(4):375–417, 1991.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

Ishan Deshpande, Ziyu Zhang, and Alexander G Schwing. Generative modeling using the sliced wasserstein distance. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3483–3491, 2018.

DC Dowson and BV Landau. The fréchet distance between multivariate normal distributions. Journal of multivariate analysis, 12(3):450–455, 1982.

Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. arXiv preprint arXiv:1903.08689, 2019.

Jiaojiao Fan, Shu Liu, Shaojun Ma, Yongxin Chen, and Haomin Zhou. Scalable computation of monge maps with general costs. arXiv preprint arXiv:2106.03812, 2021.

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.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In International Conference on Artificial Intelligence and Statistics, pp. 1608–1617. PMLR, 2018.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems Conference, pp. 2674–2680, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 5769–5779, 2017.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 30, 2017.

Leygonie Jacob, Jennifer She, Amjad Almahairi, Sai Rajeswar, and Aaron Courville. W2gan: Recovering an optimal transport map with a gan. 2018.

Leonid Vitalevich Kantorovich. On a problem of monge. Uspekhi Mat. Nauk, pp. 225–226, 1948.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4401–4410, 2019.

Cheolhyeong Kim, Seungtae Park, and Hyung Ju Hwang. Local stability of wasserstein gans with abstract gradient penalty. IEEE Transactions on Neural Networks and Learning Systems, 2021.

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

Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. arXiv preprint arXiv:1705.07215, 2017.

Alexander Korotin, Vage Egiazarian, Arip Asadulaev, Alexander Safin, and Evgeny Burnaev. Wasserstein-2 generative networks. In International Conference on Learning Representations, 2021a. URL https://openreview.net/forum?id=bEoxzW_EXsa.

Alexander Korotin, Lingxiao Li, Aude Genevay, Justin Solomon, Alexander Filippov, and Evgeny Burnaev. Do neural optimal transport solvers work? a continuous wasserstein-2 benchmark. arXiv preprint arXiv:2106.01954, 2021b.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

Na Lei, Kehua Su, Li Cui, Shing-Tung Yau, and Xianfeng David Gu. A geometric view of optimal transportation and generative model. Computer Aided Geometric Design, 68:1–21, 2019.

Huidong Liu, Xianfeng Gu, and Dimitris Samaras. Wasserstein gan with quadratic transport cost. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019.

Shu Liu, Shaojun Ma, Yongxin Chen, Hongyuan Zha, and Haomin Zhou. Learning high dimensional wasserstein geodesics. arXiv preprint arXiv:2102.02992, 2021.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In Proceedings of the IEEE international conference on computer vision, pp. 3730–3738, 2015.

Guansong Lu, Zhiming Zhou, Yuxuan Song, Kan Ren, and Yong Yu. Guiding the one-to-one mapping in cyclegan via optimal transport. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pp. 4432–4439, 2019.

Guansong Lu, Zhiming Zhou, Jian Shen, Cheng Chen, Weinan Zhang, and Yong Yu. Large-scale optimal transport via adversarial training with cycle-consistency. arXiv preprint arXiv:2003.06635, 2020.

Ashok Makkuva, Amirhossein Taghvaei, Sewoong Oh, and Jason Lee. Optimal transport mapping via input convex neural networks. In International Conference on Machine Learning, pp. 6672–6681. PMLR, 2020.

Anton Mallasto, Jes Frellsen, Wouter Boomsma, and Aasa Feragen. (q, p)-wasserstein gans: Comparing ground metrics for wasserstein gans. arXiv preprint arXiv:1902.03642, 2019.

Robert J McCann and Brendan Pass. Optimal transportation between unequal dimensions. Archive for Rational Mechanics and Analysis, 238(3):1475–1520, 2020.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In International Conference on Learning Representations, 2018.

Quan Hoang Nhan Dam, Trung Le, Tu Dinh Nguyen, Hung Bui, and Dinh Phung. Threeplayer wasserstein gan via amortised duality. In Proc. of the 28th Int. Joint Conf. on Artificial Intelligence (IJCAI), 2019.

Georg Ostrovski, Will Dabney, and Rémi Munos. Autoregressive quantile networks for generative modeling. In International Conference on Machine Learning, pp. 3936–3945. PMLR, 2018.

Brendan Pass. Regularity of optimal transportation between spaces with different dimensions. arXiv preprint arXiv:1008.1544, 2010.

Giorgio Patrini, Rianne van den Berg, Patrick Forre, Marcello Carioni, Samarth Bhargav, Max Welling, Tim Genewein, and Frank Nielsen. Sinkhorn autoencoders. In Uncertainty in Artificial Intelligence, pp. 733–743. PMLR, 2020.

Henning Petzka, Asja Fischer, and Denis Lukovnikov. On the regularization of wasserstein gans. In International Conference on Learning Representations, 2018.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016. URL http://arxiv.org/abs/1511.06434.

R Tyrrell Rockafellar. Integral functionals, normal integrands and measurable selections. In Nonlinear operators and the calculus of variations, pp. 157–207. Springer, 1976.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. Advances in neural information processing systems, 29: 2234–2242, 2016.

Maziar Sanjabi, Meisam Razaviyayn, Jimmy Ba, and Jason D Lee. On the convergence and robustness of training gans with regularized optimal transport. Advances in Neural Information Processing Systems, 2018:7091–7101, 2018.

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

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.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In Proceedings of the 33rd Annual Conference on Neural Information Processing Systems, 2019.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818–2826, 2016.

Amirhossein Taghvaei and Amin Jalali. 2-wasserstein approximation via restricted convex potentials with application to improved training for gans. arXiv preprint arXiv:1902.07197, 2019.

Ugo Tanielian and Gerard Biau. Approximating lipschitz continuous functions with groupsort neural networks. In International Conference on Artificial Intelligence and Statistics, pp. 442–450. PMLR, 2021.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. arXiv preprint arXiv:1711.01558, 2017.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. In Advances in Neural Information Processing Systems Conference, 2020.

Cédric Villani. Optimal transport: old and new, volume 338. Springer Science & Business Media, 2008.

Yujia Xie, Minshuo Chen, Haoming Jiang, Tuo Zhao, and Hongyuan Zha. On scalable and efficient computation of large scale optimal transport. In International Conference on Machine Learning, pp. 6882–6892. PMLR, 2019.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Computer Vision (ICCV), 2017 IEEE International Conference on, 2017.

A PROOFS

A.1 PROOF OF LEMMA 4.1

Proof. It is enough to prove that $\overline{\psi^*}(x) = \langle T^*(x), x\rangle - \psi^*\big(T(x)\big)$ holds μ-almost everywhere, i.e., $T^*(x) \in \arg\sup_{y\in\mathbb{R}^D}\{\langle x, y\rangle - \psi^*(y)\}$. Since $\nu = T^*_{\#}\mu$, we use (9) with $\psi \leftarrow \psi^*$ to derive

$$\begin{aligned}
&\mathcal{W}_2^2(\mu,\nu) - \int_{\mathcal{X}} \frac{1}{2}\|x\|^2 d\mu(x) - \int_{\mathcal{Y}} \frac{1}{2}\|y\|^2 d\nu(y) = \\
-\int_{\mathcal{X}} \overline{\psi^*}(x) d\mu(x) - \int_{\mathcal{Y}} \psi^*(y) d\nu(y) &= -\int_{\mathcal{X}} \overline{\psi^*}(x) d\mu(x) - \int_{\mathcal{Y}} \psi^*\big(T^*(x)\big) d\mu(x) = \\
&-\int_{\mathcal{X}} \big[\underbrace{\overline{\psi^*}(x) + \psi^*\big(T^*(x)\big)}_{\geq \langle T^*(x), x\rangle}\big] d\mu(x) \leq -\int_{\mathcal{X}} \langle T^*(x), x\rangle d\mu(x) = \\
\int_{\mathcal{X}} \frac{1}{2}\|x - T^*(x)\|^2 d\mu(x) &- \int_{\mathcal{X}} \frac{1}{2}\|x\|^2 d\mu(x) - \int_{\mathcal{X}} \frac{1}{2}\|T^*(x)\|^2 d\mu(x) = \\
&\mathcal{W}_2^2(\mu,\nu) - \int_{\mathcal{X}} \frac{1}{2}\|x\|^2 d\mu(x) - \int_{\mathcal{Y}} \frac{1}{2}\|y\|^2 d\nu(y).
\end{aligned} \tag{17}$$

As a result, inequality (17) becomes the equality, in particular, $\overline{\psi^*}(x) + \psi^*\big(T^*(x)\big) = \langle T^*(x), x\rangle$ holds μ-almost everywhere.

□

A.2 PROOF OF LEMMA 4.2

Proof. Let $Q\mathcal{W}_2^2$ denote the Q-embedded quadratic cost. We use the change of variables formula to derive

$$\begin{aligned}
Q\mathcal{W}_2^2(\mu,\nu) &= \inf_{\pi\in\Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \frac{1}{2}\|Q(x) - y\|^2 d\pi(x,y) = \\
&\inf_{\pi'\in\Pi(Q_{\#}\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \frac{1}{2}\|x - y\|^2 d\pi'(x,y) = \mathcal{W}_2^2(Q_{\#}\mu,\nu),
\end{aligned} \tag{18}$$

i.e., computing the OT plan for $Q\mathcal{W}_2^2(\mu,\nu)$ boils down to computing the OT plan for $\mathcal{W}_2^2(Q_{\#}\mu,\nu)$. It follows that $[\mathrm{id}_{\mathbb{R}^H}, T^*\big(Q(x)\big)]_{\#}\mu = [\mathrm{id}_{\mathbb{R}^H}, G^*]_{\#}\mu$ is an OT plan for $Q\mathcal{W}_2^2(\mu,\nu)$, and G^* is the OT map. Inclusion (15) now follows from Lemma 4.1. □

A.3 PROOF OF THEOREM 4.3

Proof. Pick any $G' \in \arg\sup_G \mathcal{L}(\hat{\psi}, G) = \arg\sup_G \int_{\mathcal{X}} \Big\{\langle Q(x), G(x)\rangle - \hat{\psi}\big(G(x)\big)\Big\} d\mu(x)$ or, equivalently, for all $x \in \mathbb{R}^H$, $G'(x) \in \arg\sup_y \Big\{\langle Q(x), y\rangle - \hat{\psi}(y)\Big\}$. Consequently, for all $y \in \mathbb{R}^D$

$$\langle Q(x), G'(x)\rangle - \hat{\psi}\big(G'(x)\big) \geq \langle Q(x), y\rangle - \hat{\psi}(y),$$

which after regrouping the terms yields

$$\hat{\psi}(y) \geq \hat{\psi}\big(G'(x)\big) + \langle Q(x), y - G'(x)\rangle.$$

This means that $Q(x)$ is contained in the subgradient $\partial\hat{\psi}$ at $G'(x)$ for a convex $\hat{\psi}$. Since $\hat{\psi}$ is β-strongly convex, for points $G(x), G'(x) \in \mathbb{R}^D$ and $Q(x) \in \partial\hat{\psi}\big(G'(x)\big)$ we derive

$$\hat{\psi}\big(G(x)\big) \geq \hat{\psi}\big(G'(x)\big) + \langle Q(x), G(x) - G'(x)\rangle + \frac{\beta}{2}\|G'(x) - G(x)\|^2.$$

Regrouping the terms, this gives

$$\big[\langle Q(x), G'(x)\rangle - \hat{\psi}\big(G'(x)\big)\big] - \big[\langle Q(x), G(x)\rangle - \hat{\psi}\big(G(x)\big)\big] \geq \frac{\beta}{2}\|G'(x) - G(x)\|^2.$$

Integrating w.r.t. $x \sim \mu$ yields

$$\epsilon_1 = \mathcal{L}(\hat{\psi}, G') - \mathcal{L}(\hat{\psi}, G') \geq \beta \int_{\mathcal{X}} \frac{1}{2}\|G'(x) - G(x)\|^2 d\mu(x) = \frac{\beta}{2} \cdot \|G - G'\|^2_{L^2(\mu)}. \tag{19}$$

Let G^* be the OT map from μ to ν. We use $G^*_{\#}\mu = \nu$ to derive

$$\begin{aligned}
\mathcal{L}(\hat{\psi}, G') = \int_{\mathcal{X}} \Big\{\langle Q(x), G'(x)\rangle - \hat{\psi}\big(G'(x)\big)\Big\} d\mu(x) + \int_{\mathcal{Y}} \hat{\psi}(y) d\nu(y) = \\
\int_{\mathcal{X}} \Big\{\langle Q(x), G'(x)\rangle - \hat{\psi}\big(G'(x)\big)\Big\} d\mu(x) + \int_{\mathcal{X}} \hat{\psi}\big(G^*(x)\big) d\mu(x) = \\
\int_{\mathcal{X}} \Bigg\{\underbrace{\langle Q(x), G'(x)\rangle - \hat{\psi}\big(G'(x)\big) + \hat{\psi}\big(G^*(x)\big)}_{\geq \langle Q(x), G^*(x)\rangle + \beta\frac{1}{2}\|G' - G^*\|^2}\Bigg\} d\mu(x) \geq \\
\int_{\mathcal{X}} \langle Q(x), G^*(x)\rangle d\mu(x) + \beta \int_{\mathcal{X}} \frac{1}{2}\|G' - G^*\|^2 d\mu(x).
\end{aligned} \tag{20}$$

Let ψ^* be an optimal potential in (14). Thanks to Lemma 4.2, we have

$$\begin{aligned}
\inf_{\psi} \sup_{G} \mathcal{L}(\psi, G) = \mathcal{L}(\psi^*, G^*) = \\
\int_{\mathcal{X}} \big\{\langle Q(x), G^*(x)\rangle - \psi^*\big(G^*(x)\big)\big\} d\mu(x) + \int_{\mathcal{Y}} \psi^*(y) d\nu(y) = \\
\int_{\mathcal{X}} \big\{\langle Q(x), G^*(x)\rangle - \psi^*\big(G^*(x)\big)\big\} d\mu(x) + \int_{\mathcal{X}} \psi^*\big(G^*(x)\big) d\mu(x) = \\
\int_{\mathcal{X}} \langle Q(x), G^*(x)\rangle d\mu(x)
\end{aligned} \tag{21}$$

By combining (20) with (21), we obtain

$$\epsilon_2 = \mathcal{L}(\hat{\psi}, G') - \mathcal{L}(\psi^*, G^*) \geq \beta \int_{\mathcal{X}} \frac{1}{2}\|G' - G^*\|^2 d\mu(x) = \frac{\beta}{2} \cdot \|G' - G^*\|^2_{L^2(\mu)} \tag{22}$$

The right-hand inequality of (16) follows from the triangle inequality combined with (19) and (22). The middle inequality of (16) follows from (Korotin et al., 2021a, Lemma A.2) and $G^*_{\#}\mu = \nu$.

Now we prove the left-hand inequality of (16). Let $\mathcal{I}$ be the feature extractor of the pre-trained InceptionV3 neural networks. FID score between generated (fake) $\hat{G}_{\#}\mu$ and data distribution ν is

$$\text{FID}(\hat{G}_{\#}\mu, \nu) = \text{FD}(\mathcal{I}_{\#}\hat{G}_{\#}\mu, \mathcal{I}_{\#}\nu) \leq 2 \cdot \mathcal{W}_2^2(\mathcal{I}_{\#}\hat{G}_{\#}\mu, \mathcal{I}_{\#}\nu), \tag{23}$$

where $\text{FD}(\cdot, \cdot)$ is the Fréchet distance which lower bounds $2 \cdot \mathcal{W}_2^2$, see (Dowson & Landau, 1982). Finally, from (Korotin et al., 2021a, Lemma A.1) it follows that

$$\mathcal{W}_2^2(\mathcal{I}_{\#}\hat{G}_{\#}\mu, \mathcal{I}_{\#}\nu) \leq L^2 \cdot \mathcal{W}_2^2(\hat{G}_{\#}\mu, \nu). \tag{24}$$

Here L is the Lipschitz constant of $\mathcal{I}$. We combine (23) and (24) to get the left-hand inequality in (16). □

B EXPERIMENTAL DETAILS

We use the PyTorch framework. All the experiments are conducted on 2×V100 GPUs. We compute inception and FID scores with the official implementation from OpenAI[1] and TTUR[2]. The compared results are taken from the respective papers or publicly available source codes.

[1]IS: `https://github.com/openai/improved-gan/tree/master/inception_score`
[2]FID: `https://github.com/bioinf-jku/TTUR`

B.1 General Training Details

MNIST (LeCun et al., 1998). On MNIST, we use $x \in \mathbb{R}^{128}$ and $y \in \mathbb{R}^{28\times28}$. The batch size is 64, learning rate $2 \cdot 10^{-4}$, optimizer Adam (Kingma & Ba, 2014) with betas $(0.5, 0.999)$, gradient penalty coefficient $\lambda = 10$, and the number of training epochs $T = 200$. We observe stable training while updating G once in multiple ψ updates, i.e., $k_G = 1$ and $k_\psi = 5$.

CIFAR10 (Krizhevsky et al., 2009). We use all 50000 samples while training. The latent vector $x \in \mathbb{R}^{128}$ and $y \in \mathbb{R}^{32\times32\times3}$, batch size 64, $\lambda = 10$, $k_G = 1$, $k_\psi = 5$, $T = 1000$, Adam optimizer with betas $(0, 0.99)$, and learning rate $2 \cdot 10^{-4}$.

CelebA (Liu et al., 2015). We use $x \in \mathbb{R}^{128}$ and $y \in \mathbb{R}^{64\times64\times3}$. The images are first cropped at the center with size 140 and then resized to 64×64. We consider all 202599 samples. We use Adam with betas $(0.5, 0.999)$, $T = 200$, and learning rate $1 \cdot 10^{-4}$. Other hyper-parameters are same as CIFAR10.

Image restoration. In the unpaired image restoration experiments, we use Adam optimizer with betas $(0, 0.9)$, learning rate $1 \cdot 10^{-3}$ and train for $T = 200$ epochs. Other hyper-parameters are same as the image generation experiment on CelebA.

CelebA128x128 (Liu et al., 2015). On this dataset, we resize the cropped images as in **CelebA** to 128×128, i.e. $y \in \mathbb{R}^{128\times128\times3}$. Here, $K_G = 10$, $K_\psi = 1$, $\lambda = 0.01$, learning rate $1 \cdot 10^{-4}$ and betas=$(0.5, 0.999)$. The batch size is 16 so as to fit in the GPU memory.

Toy datasets. The dimension is $D = H = 2$, total number of samples is 10000. We use the batch size 128, $\lambda = 1$, $K_\psi = 1$, $K_G = 16$, and $T = 100$. The optimizer is Adam with betas $(0.5, 0.99)$ and learning rate $1 \cdot 10^{-3}$. We use the following datasets: Gaussian to mixture of Gaussians[3], two moons (`sklearn.datasets.make_moons`), circles (`sklearn.datasets.make_circles`), gaussian to S-curve (`sklearn.datasets.make_s_curve`), and gaussian to swiss roll (`sklearn.datasets.make_swiss_roll`).

Wasserstein-2 benchmark (Appendix B.2). The dimension is $D = H = 64 \times 64 \times 3$. We use batch size 64, $\lambda = 0$, $K_\psi = 1$, $K_G = 5$, learning rate 10^{-4}, and Adam optimizer with default betas.

B.2 Evaluation on the Continuous Wasserstein-2 Benchmark

To empirically show that the method recovers the optimal transport maps well on equal dimensions, we evaluate it on the recent continuous Wasserstein-2 benchmark by Korotin et al. (2021b). The benchmark provides a number of artificial test pairs (μ, ν) of continuous probability distributions with analytically known OT map T^* between them.

For evaluation, we use the "Early" images benchmark pair ($D = 12288$), see (Korotin et al., 2021b, §4.1) for details. We adopt the $\mathcal{L}^2$-unexplained percentage metric (Korotin et al., 2021a, §5.1) to quantify the recovered OT map $\hat{T}$: $\mathcal{L}^2\text{-UVP}(\hat{T}) = 100 \cdot \|\hat{T} - T^*\|^2 / \text{Var}(\nu)\%$. For our method the $\mathcal{L}_2$-UVP metric is only $\approx 1\%$, see Table 4. This is comparable to the best ⌈MM:R⌉ method which the authors evaluate on their benchmark. The qualitative results are given in Figure 10.

Method	$\mathcal{L}^2$-UVP↓
⌊MM:R⌉	1.4%
OTM (ours)	1.32%

Table 4: $\mathcal{L}^2$-UVP metric of the recovered transport map on the "Early" images benchmark pair.

B.3 Additional Qualitative Results

CelebA128x128. Figure 11 shows the embedding $Q(x)$, the recovered transport map $\hat{G}(x)$, and independently drawn real samples $y \sim \nu$.

The extended qualitative results on **MNIST**, **CIFAR10**, and **CelebA** are shown in Figure 12, Figure 13, and Figure 14 respectively. The samples are generated randomly (uncurated) by fitted optimal transport maps between noise and ambient space, e.g., spaces of high-dimensional images.

[3] `https://github.com/AmirTag/OT-ICNN`

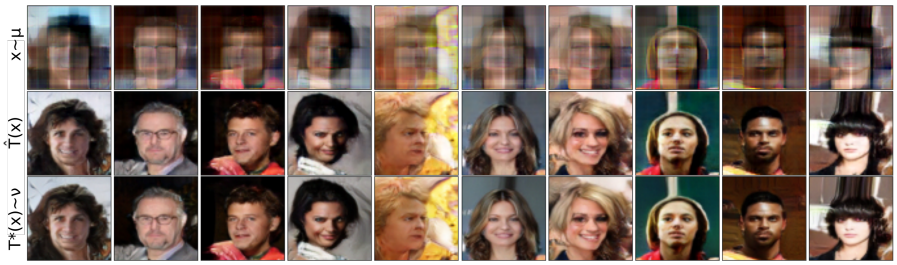


Figure 10: Qualitative results of OTM on the "Early" images benchmark pair (μ, ν) by Korotin et al. (2021b). The 1st line shows samples $x \sim \mu$, the 2nd line shows fitted OT map $\hat{T}(x)$, and the 3rd line shows the corresponding optimal map $T^*(x) \sim \nu$.

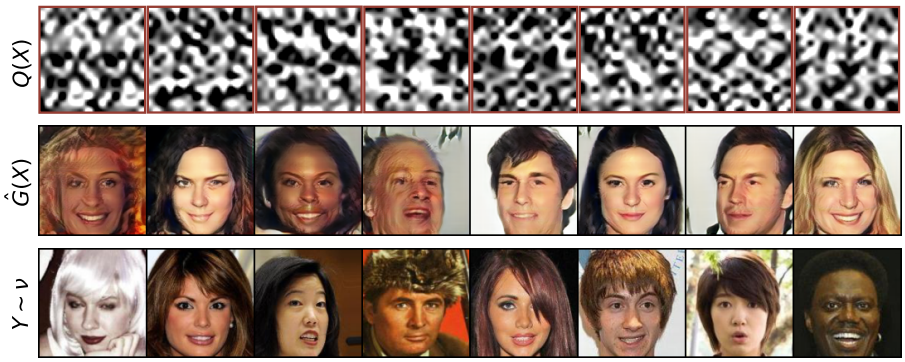


Figure 11: OTM between 128-dimensional noise and CelebA, 128×128. The 1st line shows the embedding Q (bicubic upscaling of a noise), the 2nd line shows corresponding generated samples, and the 3rd line shows random samples from the dataset.

Figure 15 illustrates latent space interpolation between the generated samples. Figure 16 shows denoising of images with varying levels of $\sigma = 0.1, 0.2, 0.3, 0.4$ by the model trained with $\sigma = 0.3$.

Toy datasets. Figure 17 shows the results of our method and related approaches (§3) on a toy 2D dataset. Figure 18 shows the results of our method applied to other toy datasets.

B.4 Neural Network Architectures

This section contains architectures on **MNIST** (Table 5), **CIFAR10** (Table 6), **CelebA** (Table 7), **CelebA** 128×128 (Figure 11) generation, **image restoration** tasks (Table 8), evaluation on the toy 2D datasets and **Wasserstein-2 images benchmark** (Table 4).

The transport map G on MNIST is the generator of DCGAN (Radford et al., 2016). In the ResNet architectures, we use the ResidualBlock of NCSN (Song & Ermon, 2019).

In the unpaired restoration tasks (§5.2), we use UNet architecture for transport map G and convolutional architecture for potential ψ. Similarly to (Song & Ermon, 2019), we use BatchNormaliation (BN) and InstanceNormalization+ (INorm+) layers.

In the toy 2D examples, we use a simple multi-layer perceptron with 3 hidden layers consisting of 128 neurons each and LeakyReLU activation. The final layer is linear without any activation.

The transport map G and potential ψ architectures on CelebA 128×128 are the generator and discriminator architectures of WGAN-QC Liu et al. (2019) respectively.

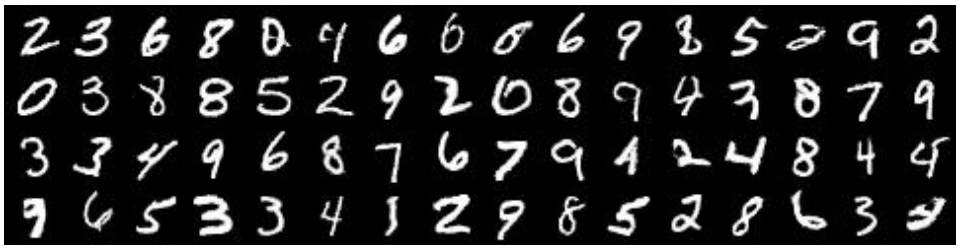


Figure 12: OTM between noise and MNIST, 28×28. Extended samples.

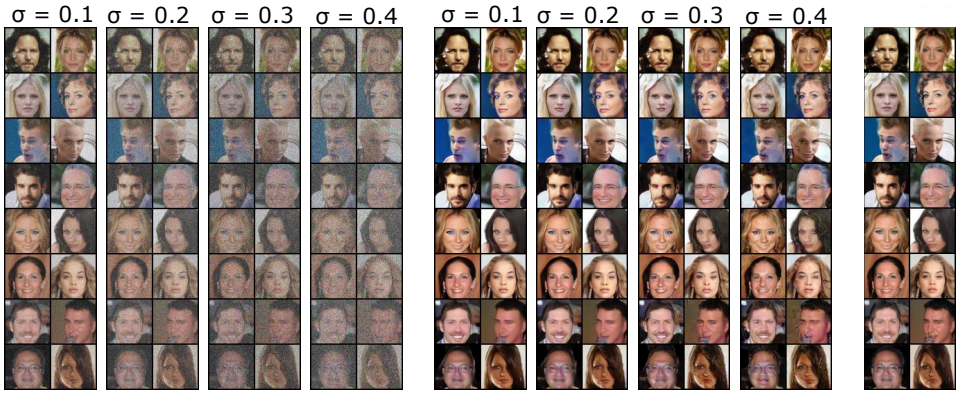
Figure 13: OTM between noise and CIFAR10, 32×32. Extended samples.

In the evaluation on the Wasserstein-2 benchmark, we use publicly available Unet[4] architecture for transport map T and WGAN-QC discriminator's architecture for ψ (Liu et al., 2019). These neural network architectures are the same as the authors of the benchmark use.

[4]https://github.com/milesial/Pytorch-UNet

Figure 14: OTM between noise and CelebA, 64×64. Extended samples.

Figure 15: OTM for latent space interpolation on CelebA, 64×64. Extended samples.



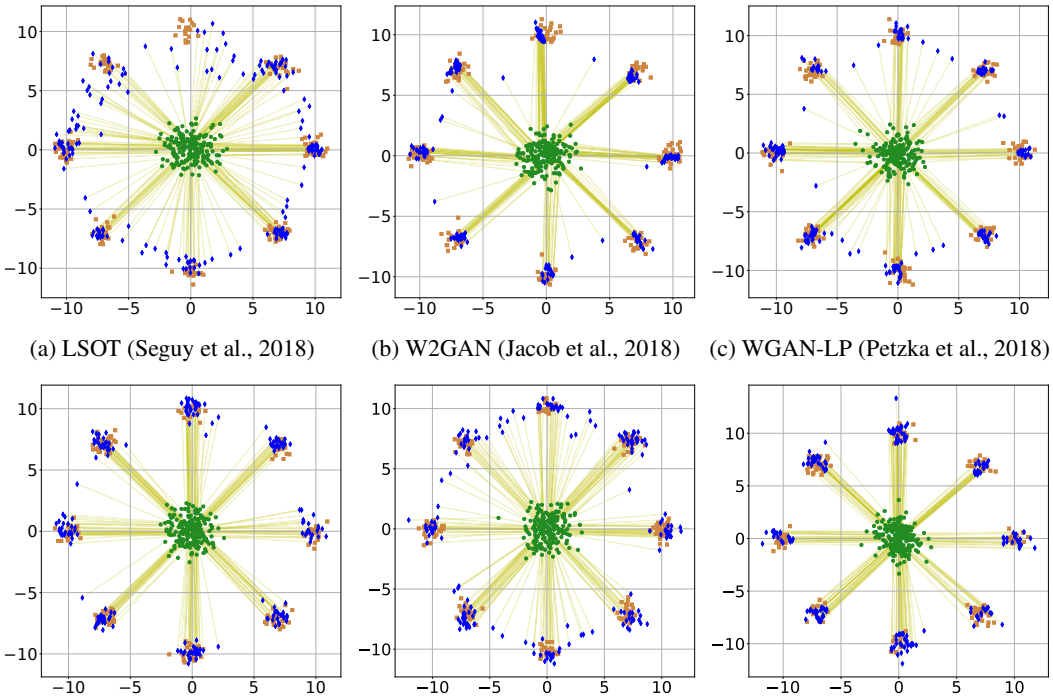
(a) Noisy samples. (b) Pushforward samples. (c) Clean.

Figure 16: OTM for image denoising for varying levels of noise on test C part of CelebA, 64×64.

Table 5: Architectures for generation task on MNIST, 28×28.

$G(.)$
Noise: $x \in \mathbb{R}^{128}$
ConvTranspose, BN, ReLU, output shape: $[128 \times 1 \times 1]$
ConvTranspose, BN, ReLU, output shape: $[256 \times 4 \times 4]$
ConvTranspose, BN, ReLU, output shape: $[512 \times 7 \times 7]$
ConvTranspose, BN, ReLU, output shape: $[1024 \times 14 \times 14]$
ConvTranspose, Tanh, output shape: $[1 \times 28 \times 28]$

$\psi(.)$
Target: $y \in \mathbb{R}^{1 \times 28 \times 28}$
Conv, INorm, LeakyReLU, output shape: $[128 \times 14 \times 14]$
Conv, INorm, LeakyReLU, output shape: $[256 \times 7 \times 7]$
Conv, INorm, LeakyReLU, output shape: $[512 \times 4 \times 4]$
Conv, output shape: $[1]$



(a) LSOT (Seguy et al., 2018) (b) W2GAN (Jacob et al., 2018) (c) WGAN-LP (Petzka et al., 2018)

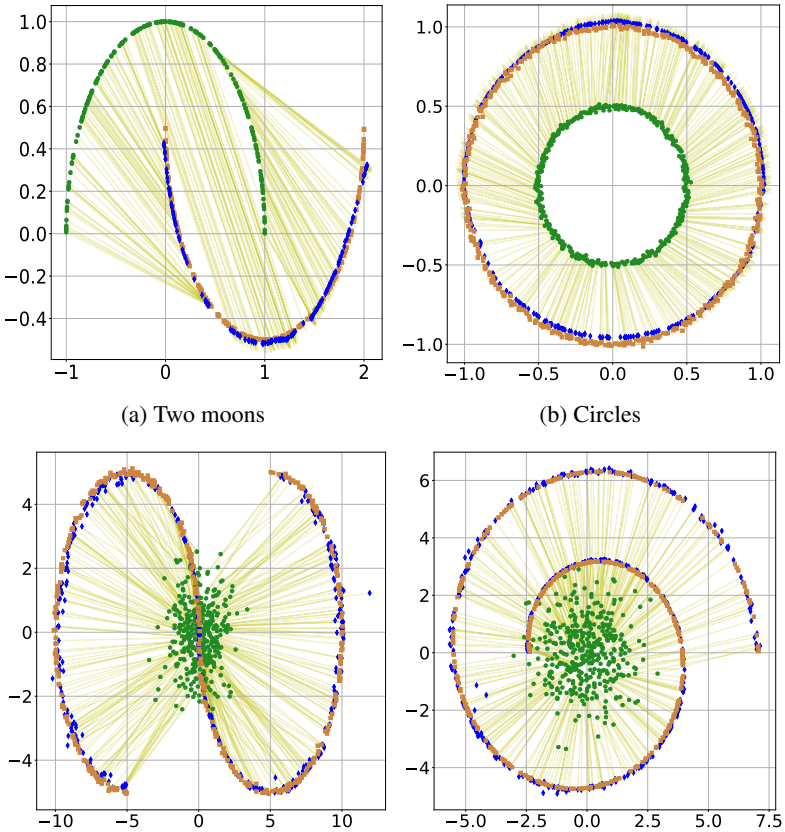
(d) ICNN-OT(Makkuva et al., 2020) (e) W2GN (Korotin et al., 2021a) (f) OTM (ours)

Figure 17: Mapping between a Gaussian and a Mixture of 8 Gaussians in 2D by various methods. The colors green, blue, and peru represent input, pushforward, and output samples respectively.

Table 6: Architectures for generation task on CIFAR10, 32×32.

$G(.)$
Noise: $x \in \mathbb{R}^{128}$
Linear, Reshape, output shape: $[128 \times 4 \times 4]$
ResidualBlock Up, output shape: $[128 \times 8 \times 8]$
ResidualBlock Up, output shape: $[128 \times 16 \times 16]$
ResidualBlock Up, output shape: $[128 \times 32 \times 32]$
Conv, Tanh, output shape: $[3 \times 32 \times 32]$

$\psi(.)$
Target: $y \in \mathbb{R}^{3\times32\times32}$
ResidualBlock Down, output shape: $[128 \times 16 \times 16]$
ResidualBlock Down, output shape: $[128 \times 8 \times 8]$
ResidualBlock, output shape: $[128 \times 8 \times 8]$
ResidualBlock, output shape: $[128 \times 8 \times 8]$
ReLU, Global sum pooling, output shape: $[128 \times 1 \times 1]$
Linear, output shape: $[1]$



(a) Two moons (b) Circles

(c) S Curve (d) Swiss Roll

Figure 18: OTM on toy datasets, $D = 2$. Here, the colors green, blue, and peru represent input, pushforward, and output samples respectively.

Table 7: Architectures for generation task on Celeba, 64×64.

$G(.)$
Noise: $x \in \mathbb{R}^{128}$
ConvTranspose, BN, LeakyReLU, output shape: $[256 \times 1 \times 1]$
ConvTranspose, BN, LeakyReLU, output shape: $[512 \times 4 \times 4]$
Conv, PixelShuffle, BN, LeakyReLU, output shape: $[512 \times 8 \times 8]$
Conv, PixelShuffle, BN, LeakyReLU, output shape: $[512 \times 16 \times 16]$
Conv, PixelShuffle, BN, LeakyReLU, output shape: $[512 \times 32 \times 32]$
ConvTranspose, Tanh, output shape: $[3 \times 64 \times 64]$

$\psi(.)$
Target: $y \in \mathbb{R}^{3\times64\times64}$
Conv, output shape: $[128 \times 64 \times 64]$
ResidualBlock Down, output shape: $[256 \times 32 \times 32]$
ResidualBlock Down, output shape: $[256 \times 16 \times 16]$
ResidualBlock Down, output shape: $[256 \times 8 \times 8]$
ResidualBlock Down, output shape: $[128 \times 4 \times 4]$
Conv, output shape: $[1]$

Table 8: Architectures for restoration tasks on CelebA, 64×64.

$G(.)$
Input: $x \in \mathbb{R}^{3\times64\times64}$
Conv, BN, LeakyReLU, output shape: $[256 \times 64 \times 64]$
Conv, LeakyReLU, AvgPool, output shape: $[256 \times 32 \times 32]$, x1
Conv, LeakyReLU, AvgPool, output shape: $[256 \times 16 \times 16]$, x2
Conv, LeakyReLU, AvgPool, output shape: $[256 \times 8 \times 8]$, x3
Conv, LeakyReLU, AvgPool, output shape: $[256 \times 4 \times 4]$, x4
Nearest Neighbour Upsample, Conv, BN, ReLU, output shape: $[256 \times 8 \times 8]$, y3
Add (y3, x3), output shape: $[256 \times 8 \times 8]$, y3
Nearest Neighbour Upsample, Conv, BN, ReLU, output shape: $[256 \times 16 \times 16]$, y2
Add (y2, x2), output shape: $[256 \times 16 \times 16]$, y2
Nearest Neighbour Upsample, Conv, BN, ReLU, output shape: $[256 \times 32 \times 32]$, y1
Add (y1, x1), output shape: $[256 \times 32 \times 32]$, y1
Nearest Neighbour Upsample, Conv, BN, ReLU, output shape: $[256 \times 64 \times 64]$, y
Add (y, x), output shape: $[256 \times 64 \times 64]$, y
ConvTranspose, Tanh, output shape: $[3 \times 64 \times 64]$

$\psi(.)$
Target: $y \in \mathbb{R}^{3\times64\times64}$
Conv, LeakyReLU, AvgPool, output shape: $[256 \times 32 \times 32]$
Conv, LeakyReLU, AvgPool, output shape: $[256 \times 16 \times 16]$
Conv, LeakyReLU, AvgPool, output shape: $[256 \times 8 \times 8]$
Conv, LeakyReLU, AvgPool, output shape: $[256 \times 4 \times 4]$
Linear, output shape: $[1]$