
Kantorovich Strikes Back!

Wasserstein GANs are not Optimal Transport?

Alexander Korotin*

Skolkovo Institute of Science and Technology
Artificial Intelligence Research Institute
Moscow, Russia
a.korotin@skoltech.ru

Alexander Kolesov*

Skolkovo Institute of Science and Technology
Moscow, Russia
a.kolesov@skoltech.ru

Evgeny Burnaev

Skolkovo Institute of Science and Technology
Artificial Intelligence Research Institute
Moscow, Russia
e.burnaev@skoltech.ru

Abstract

Wasserstein Generative Adversarial Networks (WGANs) are the popular generative models built on the theory of Optimal Transport (OT) and the Kantorovich duality. Despite the success of WGANs, it is still unclear how well the underlying OT dual solvers approximate the OT cost (Wasserstein-1 distance, \mathbb{W}_1) and the OT gradient needed to update the generator. In this paper, we address these questions. We construct 1-Lipschitz functions and use them to build ray monotone transport plans. This strategy yields pairs of continuous benchmark distributions with the analytically known OT plan, OT cost and OT gradient in high-dimensional spaces such as spaces of images. We thoroughly evaluate popular WGAN dual form solvers (gradient penalty, spectral normalization, entropic regularization, etc.) using these benchmark pairs. Even though these solvers perform well in WGANs, none of them faithfully compute \mathbb{W}_1 in high dimensions. Nevertheless, many provide a meaningful approximation of the OT gradient. These observations suggest that these solvers should not be treated as good estimators of \mathbb{W}_1 , but to some extent they indeed can be used in variational problems requiring the minimization of \mathbb{W}_1 .

The Wasserstein-1 distance [3] (\mathbb{W}_1) is a popular loss function to learn generative models. It has numerous advantages compared to the vanilla GAN loss [15]. For example, \mathbb{W}_1 is correctly defined if the distributions' supports differ [2]. Besides, it correlates with the sample quality, provides improved stability of the optimization process and does not suffer from the vanishing gradients issue [3, §4].

Generative models which employ \mathbb{W}_1 as the loss to update the generator are called the Wasserstein GANs (WGANs). To compute \mathbb{W}_1 , they use its variational approximation based on the **Kantorovich duality** [20] and the **Optimal Transport** (OT) theory [52, 47]. Since the introduction of the original WGANs with the weight clipping method [3], a lot of alternative techniques (*neural dual OT solvers*) to compute \mathbb{W}_1 have been proposed: gradient penalties [17, 39, 54], entropic regularization [46], architectural constraints [35, 1], batch-based methods [31, 29], maximin methods [38, 23], etc.

Despite the popularity of WGANs, it still remains unclear to what extent their success is connected to OT and \mathbb{W}_1 rather than, e.g., to a good choice of regularization [49, §7]. Due to the limited amount of pairs of distributions with known \mathbb{W}_1 , it is challenging to evaluate existing dual OT solvers.

Contributions. We develop a generic methodology based on the *transport rays* (§3.1) to evaluate dual OT solvers for the Wasserstein-1 distance (\mathbb{W}_1). Our main contributions are as follows:

*Equal contribution.

- We use 1-Lipschitz functions to construct pairs of *continuous* distributions that we use as a benchmark with analytically-known OT cost, map and gradient for \mathbb{W}_1 transport (§3.2, §3.3).
- We use these *benchmark distributions* to evaluate (§4) popular WGAN dual OT solvers (§2) in high-dimensional spaces, including the spaces of 32×32 CIFAR-10 images, 64×64 CelebA faces.

Related works [32, 49] consider *discrete* distributions and show that some solvers fail to estimate \mathbb{W}_1 . In contrast to them, we study how well the solvers compute the gradient of \mathbb{W}_1 (*OT gradient*), as it is the OT gradient which is used to update the generator in WGANs, not the value of \mathbb{W}_1 . We use *continuous* distributions since in the discrete case the OT gradient may be ill-defined (§1).

Notation. We work in the \mathbb{R}^D space that is endowed with the Euclidean norm $\|\cdot\|_2$. We use μ_L to denote the Lebesgue measure on \mathbb{R}^D . For a measurable map $T : \mathbb{R}^D \rightarrow \mathbb{R}^D$, we denote the associated pushforward operator by $T\#$. We consider Borel probability distributions \mathbb{P}, \mathbb{Q} on \mathbb{R}^D with finite first moments. We use $\Pi(\mathbb{P}, \mathbb{Q})$ to denote the set of probability distributions on $\mathbb{R}^D \times \mathbb{R}^D$ with marginals \mathbb{P} and \mathbb{Q} (transport plans). All the integrals are computed over \mathbb{R}^D , if not stated otherwise. We write $\|f\|_L \leq C$ if $f : \mathbb{R}^D \rightarrow \mathbb{R}$ is C -Lipschitz.

1 Background on Optimal Transport

Primal Formulation. For distributions \mathbb{P}, \mathbb{Q} , the **Monge's** formulation of the Wasserstein-1 (\mathbb{W}_1) distance, i.e., OT with the distance cost function $\|x - y\|_2$, is given by (Figure 1a)

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) \stackrel{\text{def}}{=} \min_{T\#\mathbb{P}=\mathbb{Q}} \int \|x - T(x)\|_2 d\mathbb{P}(x), \quad (1)$$

where min is taken over measurable functions $T : \mathbb{R}^D \rightarrow \mathbb{R}^D$ (transport maps) that map \mathbb{P} to \mathbb{Q} . The optimal T^* is called the *optimal transport map* (OT map). Note that (1) is not symmetric, and this formulation does not allow for mass splitting, i.e., for some \mathbb{P}, \mathbb{Q} , there is no map T that satisfies $T\#\mathbb{P} = \mathbb{Q}$ [40, Remark 2.4]. Thus, **Kantorovich** [20] proposed the following relaxation (Figure 1b):

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) \stackrel{\text{def}}{=} \min_{\pi \in \Pi(\mathbb{P}, \mathbb{Q})} \int_{\mathbb{R}^D \times \mathbb{R}^D} \|x - y\|_2 d\pi(x, y), \quad (2)$$

where min is taken over transport plans $\pi \in \Pi(\mathbb{P}, \mathbb{Q})$. The optimal $\pi^* \in \Pi(\mathbb{P}, \mathbb{Q})$ is called the *optimal transport plan* (OT plan). If $\pi^* = [\text{id}_{\mathbb{R}^D}, T^*]\#\mathbb{P}$ for some map $T^* : \mathbb{R}^D \rightarrow \mathbb{R}^D$, then T^* minimizes formulation (1). In general, there might exist more than one OT plan π^* or OT map T^* .

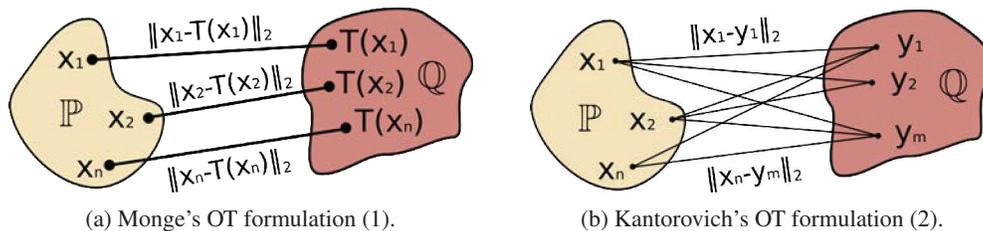


Figure 1: Monge's and Kantorovich's OT formulations of the Wasserstein-1 distance (\mathbb{W}_1).

Dual formulation. For distributions \mathbb{P}, \mathbb{Q} , the dual formulation of \mathbb{W}_1 is given by [52, Thm. 5.10]:

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) = \max_{f \oplus g \leq \|\cdot\|_2} \int f(x) d\mathbb{P}(x) + \int g(y) d\mathbb{Q}(y), \quad (3)$$

where max is taken over $f, g : \mathbb{R}^D \rightarrow \mathbb{R}$ satisfying $f(x) + g(y) \leq \|x - y\|_2$ for all $x, y \in \mathbb{R}^D$. By using the c -transform $f^c(y) \stackrel{\text{def}}{=} \min_{x \in \mathbb{R}^D} \{\|x - y\|_2 - f(x)\}$ [52, §5], one rewrites (3) as

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) = \max_f \int f(x) d\mathbb{P}(x) + \int f^c(y) d\mathbb{Q}(y). \quad (4)$$

In accordance with [52, Case 5.16], dual form (4) can be further restricted to 1-Lipschitz functions. In this case, it holds $f^c(y) = -f(y)$ [52, Case 5.4], and the alternative duality formula for \mathbb{W}_1 is

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) = \max_{\|f\|_L \leq 1} \int f(x) d\mathbb{P}(x) - \int f(y) d\mathbb{Q}(y). \quad (5)$$

In WGAN literature [17, 2], function f is typically called the *critic* (or discriminator). In OT literature [52, 47, 51], functions f, f^c, g are commonly referred as the (Kantorovich) *potentials*.

Optimal transport in GANs. Derivatives of \mathbb{W}_1 are used implicitly in generative modeling [3, 39, 35, 17, 38, 46, 48, 28] that incorporates \mathbb{W}_1 loss, in which case $\mathbb{P} = \mathbb{P}_\alpha$ is a parametric distribution and \mathbb{Q} is the data distribution. Typically, $\mathbb{P}_\alpha = G_\alpha \# \mathbb{S}$ is the distribution generated from a fixed latent distribution \mathbb{S} by a generator network G_α . The goal is to find parameters α that minimize $\mathbb{W}_1(\mathbb{P}_\alpha, \mathbb{Q})$ via gradient descent. The loss function for the generator is:

$$\mathbb{W}_1(\mathbb{P}_\alpha, \mathbb{Q}) = \int_z f^*(G_\alpha(z)) d\mathbb{S}(z) - \int f^*(y) d\mathbb{Q}(y),$$

where f^* is the optimal potential in (5). The loss derivative (Figure 2) is given by [13, Eq. 3]:

$$\frac{\partial \mathbb{W}_1(\mathbb{P}_\alpha, \mathbb{Q})}{\partial \alpha} = \int_z \mathbf{J}_\alpha G_\alpha(z)^T \nabla f^*(G_\alpha(z)) d\mathbb{S}(z),$$

where $\mathbf{J}_\alpha G_\alpha(z)^T$ is the transpose of the Jacobian matrix of $G_\alpha(z)$ w.r.t. parameters α . This result still holds without assuming the potentials are fixed [3, Theorem 3] by the envelope theorem [33].

In practice, the optimal potential f^* is unknown. Therefore, WGANs approximate it with a network $f_\theta : \mathbb{R}^D \rightarrow \mathbb{R}$ (potential) by maximizing (5), (4) or (3) via the stochastic gradient ascent (SGA). This is usually associated with *evaluating* the \mathbb{W}_1 loss. However, note that *the loss value plays no role* in generator updates. Only the gradient ∇f^* of the potential is needed. We call it the *OT gradient*. We use a generic phrase *OT solver* to refer to any algorithm which is capable of recovering ∇f^* .

Quantitative evaluation of OT solvers. Existing solvers are typically tested as the loss in WGANs without evaluating the actual OT performance. The quality of the generated samples is evaluated by standard metrics such as FID [18] or IS [45]. These metrics do not provide understanding about the quality of the solver itself since they depend on components of the model that are not related to OT.

In [41, 32, 49], the authors use discrete \mathbb{P}, \mathbb{Q} to show that some solvers imprecisely compute \mathbb{W}_1 . Their approach is *not applicable* to evaluation of the OT gradient, as ∇f^* is ill-defined in the discrete case. For example, when $\mathbb{P} = \delta_0, \mathbb{Q} = \delta_1$, it holds that $f^* = -[x]_+$ is an optimal potential, but it is not even differentiable at $x = 0 = \text{Supp}(\mathbb{P})$. The existence of the OT gradient is studied, e.g., in [19].

2 Neural Dual Solvers for the Wasserstein-1 Distance

Our proposed benchmark is useful for testing any OT solver that computes ∇f^* or \mathbb{W}_1 . We evaluate only neural solvers which are based on (5), (4), or (3) and used in WGANs. We provide an overview of these methods below. We group them by the dual formulations which they use.

Most solvers approximate the potential by a network $f_\theta : \mathbb{R}^D \rightarrow \mathbb{R}$ and learn it via maximizing (5) with SGA on batches from \mathbb{P}, \mathbb{Q} . The main challenge is to enforce the 1-Lipschitz constraint for f_θ .

[WC] In [3], the space Θ of parameters is restricted to a compact, e.g., to a hypercube $[-c, c]^{\dim \theta}$. With mild assumptions on the architecture, f_θ is provably Lipschitz continuous with some *unknown* constant C , i.e., $\|f_\theta\|_L \leq C$. The main practical issue is tuning the boundary c of the set.

[GP] The authors of [17] prove that with mild assumptions on the OT plan π^* , the equation $\|\nabla f^*(z)\|_2 = 1$ holds almost surely for $z = tx + (1-t)y$ with (x, y) distributed as the OT plan π^* and $t \sim \text{Uniform}[0, 1]$. Thus, they *softly* penalize f for being not 1-Lipschitz and optimize

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) \approx \max_f \left\{ \int f(x) d\mathbb{P}(x) - \int f(y) d\mathbb{Q}(y) - \lambda \mathcal{R}(f) \right\}, \quad \lambda > 0. \quad (6)$$

The *gradient penalty* $\mathcal{R}_{\text{GP}}(f)$ equals $\int (\|\nabla f(r)\|_2 - 1)^2 d\mu(r)$, where μ is the distribution of a random variable $r = xt + (1-t)y$ with $t \sim \text{Uniform}[0, 1]$ and $(x, y) \sim \mathbb{P} \times \mathbb{Q}$ (as π^* is unknown). In [34], it is proved that (6) with $\mathcal{R} = \mathcal{R}_{\text{GP}}$ is a specific OT formulation called *congested OT*.

[LP] In [39], the authors show that $\mathcal{R}_{\text{GP}}(f)$ suffers from instabilities and high magnitudes. They introduce the *Lipschitz penalty* $\mathcal{R}_{\text{LP}}(f) = \int (\max\{0, \|\nabla f(r)\|_2 - 1\})^2 d\mu(r)$ resolving the issues.

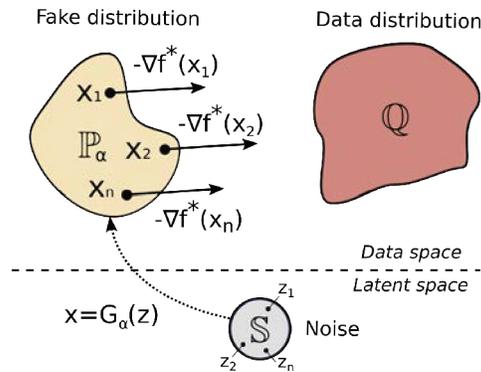


Figure 2: The anti-gradient $-\nabla f^*(x)$ shows where to move the mass of each $x = G_\alpha(z)$ to make the generated \mathbb{P}_α closer to \mathbb{Q} in \mathbb{W}_1 .

[SN] In [35], the authors optimize (5) and use the power iteration method [14] to normalize the weight matrices of linear layers of net f_θ by their spectral norms. This provably makes f globally Lipschitz continuous, which does not hold for [17, 39] which are based on the soft penalization (6).

[SO] The authors of [1] claim that spectral normalization negatively affects the expressiveness of the network. They propose to ortho-normalize weight matrices and use GroupSort activations [5]. Such networks f_θ are 1-Lipschitz and universally approximate 1-Lipschitz functions [1, §4].

Below we overview methods based on (4) or (3) which do not require enforcing 1-Lipschitz continuity for the potential. Unlike the above-mentioned methods, they mostly require 2 neural networks.

[LS] In [46, 48, 12, 6], the authors optimize the following *unconstrained* regularized form (3):

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) \approx \max_{f, g} \left\{ \int f(x) d\mathbb{P}(x) + \int g(y) d\mathbb{Q}(y) - \mathcal{R}(f, g) \right\}, \quad (7)$$

where $\mathcal{R}(f, g)$ is the *entropic* or *quadratic* regularizer [46, Eq.5] which softly penalizes the potentials f, g for disobeying $f \oplus g \leq \|\cdot\|_2$. In practice, f, g are neural networks $f_\theta, g_\omega : \mathbb{R}^D \rightarrow \mathbb{R}$.

[MM:B] The authors of [31, 32] expand the dual formulation (4) with the c -transform:

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) = \max_f \left\{ \int f(x) d\mathbb{P}(x) + \int \min_{x \in \mathbb{R}^D} [\|x - y\|_2 - f(y)] d\mathbb{Q}(y) \right\}. \quad (8)$$

During optimization, they restrict the inner minimization to the current mini-batch from \mathbb{P} . This leads to *overestimation* of the inner problem's solution since the minimum is taken over a restricted subset. Recently, a more tricky version of this approach appeared [27, §3]. We call it [MM:Bv2].

[MM] In [38], the authors use a saddle point formulation equivalent to (8):

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) = \max_f \int f(x) d\mathbb{P}(x) + \min_H \int [\|H(y) - y\|_2 - f(H(y))] d\mathbb{Q}(y), \quad (9)$$

where the minimization is performed over functions $H : \mathbb{R}^D \rightarrow \mathbb{R}^D$. The authors use neural networks f_θ and H_ω to parametrize the potential and the minimizer of the inner problem (*mover*). To train θ, ω , the authors apply stochastic gradient ascent/descent (SGAD) over mini-batches from \mathbb{P}, \mathbb{Q} .

[MM:R] One may also recover the OT gradient ∇f^* from the OT map T^* . Consider the form

$$\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) = \max_g \min_T \int [\|T(x) - x\|_2 - g(T(x))] d\mathbb{P}(x) + \int g(y) d\mathbb{Q}(y), \quad (10)$$

which is a *reversed* [23] version of (9), i.e., the roles of \mathbb{P}, \mathbb{Q} are swapped and $T : \mathbb{R}^D \rightarrow \mathbb{R}^D$. For *some* optimal saddle points (g^*, T^*) of (10) it holds that mover T^* is an OT map [25, Lemma 4], [11, Lemma 2], [50], [8]. With mild assumptions on \mathbb{P}, \mathbb{Q} , one may recover T^* and use the identity $\nabla f^*(x) = \frac{x - T^*(x)}{\|x - T^*(x)\|_2}$ [7, §1] to obtain the OT gradient from T^* .

3 Constructing Benchmark Distributions for Dual Solvers

In this section, we develop a generic methodology to construct pairs (\mathbb{P}, \mathbb{Q}) with computable ground truth OT plan, OT cost and OT gradient. Our approach is inspired by the insights about OT plans in [47, §3.1], [16], [7, §3-7]. In §3.1, we give the required preliminaries. In §3.2, we provide our method to build benchmark pairs. We construct them in §3.3. The proofs are given in Appendix A.

3.1 Ray Monotone Transport Plans and 1-Lipschitz Functions

Let $u : \mathbb{R}^D \rightarrow \mathbb{R}$ be a 1-Lipschitz function. Recall that due to the Rademacher's theorem, u is differentiable μ_L -almost everywhere. For every $x, y \in \mathbb{R}^D$ satisfying $u(x) - u(y) = \|x - y\|_2$ it holds that u is affine on the segment $[x, y]$, i.e., $u(z) = tu(x) + (1-t)u(y)$ for $z = tx + (1-t)y$ with $t \in [0, 1]$ [47, Lemma 3.5]. Moreover, u is differentiable at all $z \in (x, y)$ and $\nabla u(z) = \frac{x-y}{\|x-y\|_2}$ [47, Lemma 3.6]. Following [47, Definition 3.7], we call a *transport ray* any non-trivial (different from a singleton) segment $[x, y]$ such that $u(x) - u(y) = \|x - y\|_2$, which is maximal for the inclusion among segments of this form. The unit vector $\frac{x-y}{\|x-y\|_2}$ is called the direction of a transport ray. Two transport rays can only intersect at their boundary points [46, Corollary 3.8]. In general, not all points

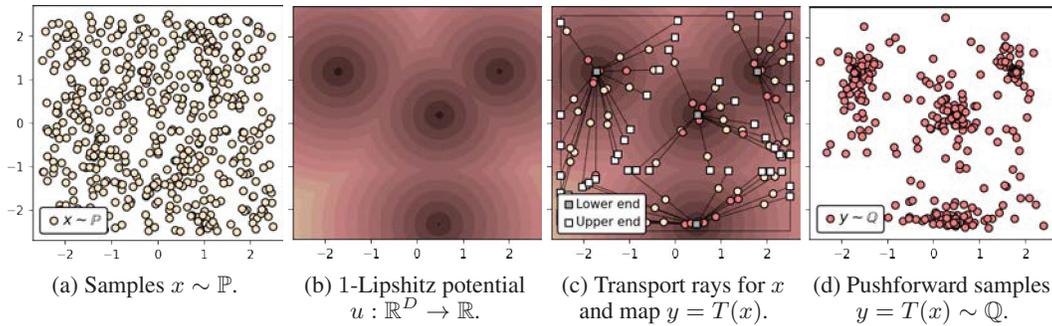


Figure 3: Our methodology to construct benchmark pairs (\mathbb{P}, \mathbb{Q}) . An example with $(D, N) = (2, 4)$. We pick a distribution \mathbb{P} and a MinFunnel function $u: \mathbb{R}^D \rightarrow \mathbb{R}$ (11). To sample $y \sim \mathbb{Q}$, we get $x \sim \mathbb{P}$, compute its transport ray and move x along the ray with the u -ray monotone map T (12).

belong to transport rays. For example, for the function $u(x) = \frac{1}{2}x$ there are no transport rays at all. In Figures 3c and 9, we provide examples of transport rays for 1-Lipshitz functions u in $D = 2$.

For a 1-Lipshitz function $u: \mathbb{R}^D \rightarrow \mathbb{R}$, we say that a transport plan $\pi \in \Pi(\mathbb{P}, \mathbb{Q})$ is u -ray monotone (decreasing) if $u(x) - u(y) = \|x - y\|_2$ holds π -almost surely for $x, y \in \mathbb{R}^D$. The idea of the definition is that such plans distribute the probability mass of $x \sim \mathbb{P}$ among $y \in \mathbb{R}^D$ such that $y = x$ (no mass movement) and/or y which lie on the same transport ray as x but below x , i.e., $u(y) < u(x)$. If x is not contained in a transport ray, then $\pi(y|x) = \delta_x$, i.e., π necessarily does not move x .

Proposition 1 (Ray monotone transport plans are optimal). *Let $\pi \in \Pi(\mathbb{P}, \mathbb{Q})$ be a u -ray monotone transport plan for a 1-Lipshitz function u . Then it is an optimal plan between \mathbb{P}, \mathbb{Q} . Besides, u is an optimal potential, i.e., it attains the maximum in dual formulation (5).*

For a distribution \mathbb{P} , we say that distribution \mathbb{Q} is a u -ray-forward of \mathbb{P} if there exists a measurable function $T: \mathbb{R}^D \rightarrow \mathbb{R}^D$ satisfying $T\# \mathbb{P} = \mathbb{Q}$ and $u(x) - u(T(x)) = \|x - T(x)\|_2$ holds \mathbb{P} -almost surely for all $x \in \mathbb{R}^D$. We say that such a T is a u -ray-monotone transport map from \mathbb{P} to \mathbb{Q} . Note that the deterministic plan $\pi = [\text{id}_{\mathbb{R}^D}, T]\# \mathbb{P}$ is u -ray monotone. We have the following corollary:

Corollary 1 (Ray monotone transport maps are optimal). *Let T be a u -ray monotone transport map from \mathbb{P} to \mathbb{Q} . Then the plan $\pi = [\text{id}_{\mathbb{R}^D}, T]\# \mathbb{P}$ is optimal and T is an OT map from \mathbb{P} to \mathbb{Q} .*

In §3.2 below, we derive our recipe to construct **benchmark pairs** (\mathbb{P}, \mathbb{Q}) such that \mathbb{Q} is a u -ray-forward of \mathbb{P} with user-defined u and analytically known T . In such pairs \mathbb{P} is accessible by samples and is also possible to sample from \mathbb{Q} by pushing $x \sim \mathbb{P}$ forward by T . Since T is an OT map, the ground truth OT cost is $\mathbb{W}_1(\mathbb{P}, \mathbb{Q}) = \int \|x - T(x)\|_2 d\mathbb{P}(x)$. It admits unbiased Monte Carlo estimates from samples $x \sim \mathbb{P}$. Moreover, with mild assumptions (§3.3), ∇u is the unique OT gradient. Our benchmark pairs can be used to test how well OT solvers recover \mathbb{W}_1 and the OT gradient.

3.2 Method to Construct Benchmark Pairs

Let u be a known 1-Lipshitz function. We aim to find its transport rays and construct a u -ray-forward map T and distribution $\mathbb{Q} = T\# \mathbb{P}$ (for a given \mathbb{P}) for testing dual OT solvers. First, we describe the parametric class of 1-Lipshitz functions u which we employ in our benchmark. Second, we explain how to compute the transport rays of u . Finally, we explain how to define u -ray monotone maps.

Part 1. Parameterizing 1-Lipshitz functions. Inspired by the distance representation of optimal potentials [16], as $u: \mathbb{R}^D \rightarrow \mathbb{R}$, we employ the following 1-Lipshitz *MinFunnel* functions:

$$u(x) \stackrel{\text{def}}{=} \min_n \{u_n(x)\} = \min_n \{\|x - a_n\|_2 + b_n\}, \quad (11)$$

where $a_n \in \mathbb{R}^D$ and $b_n \in \mathbb{R}$ are the parameters. Each *funnel* $u_n(x) = \|x - a_n\|_2 + b_n$ has $\|\nabla u_n(x)\|_2 = 1$ when $x \neq a_n$. Thus, u is also 1-Lipshitz as it is their minimum. Note that for μ_L -almost every x it holds that u is differentiable at x and $\|\nabla u(x)\|_2 = 1$.

Proposition 2 (MinFunnels are universal approximators of 1-Lipshitz functions on compact sets). *Let $\mathcal{S} \subset \mathbb{R}^D$ be a compact set and $f^*: \mathcal{S} \rightarrow \mathbb{R}$ be a 1-Lipshitz function. Then for every $\epsilon > 0$ there exists N and $\{a_n, b_n\}_{n=1}^N$ such that function (11) satisfies $\sup_{x \in \mathcal{S}} |u(x) - f^*(x)| \leq \epsilon$.*

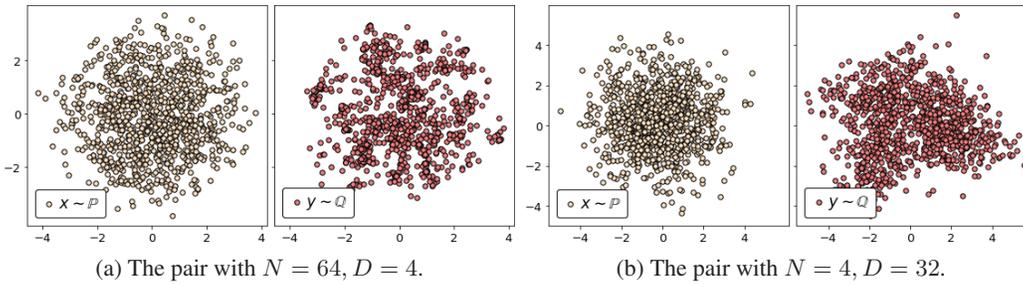


Figure 4: Visualization of constructed high-dimensional benchmark pairs. In each pair, we show random samples projected onto 2 principal components of \mathbb{Q} .

The proposition is included only for completeness of the exposition and is not principal for our construction. We do not aim to approximate a specific f^* by u . In practice, we found that simply picking a *random* MinFunnel u and using it to construct a benchmark pair is reasonable (§3.3).

Part 2. Computing the transport rays. MinFunnels are practically very convenient as the transport ray for a given $x \in \mathbb{R}^D$ can be analytically computed in $O(ND)$ time, see our proposition below.

Proposition 3 (Transport rays of a MinFunnel). *Consider MinFunnel (11) function $u : \mathbb{R}^D \rightarrow \mathbb{R}$ with n distinct centers a_n such that for all $n_1 \neq n_2$ it holds that $\|a_{n_1} - a_{n_2}\|_2 \neq |b_{n_1} - b_{n_2}|$. Let $x \in \mathbb{R}^D$ be a point such that u is differentiable at x . Then $\text{ray}(x) = [a_m, x + r \cdot v]$, where*

$$m \stackrel{\text{def}}{=} \arg \min_n \{u_n(x)\}; \quad v \stackrel{\text{def}}{=} \frac{x - a_m}{\|x - a_m\|_2}; \quad r \stackrel{\text{def}}{=} \min_n r_n \in \mathbb{R} \cup \{+\infty\},$$

and $r_n \stackrel{\text{def}}{=} \frac{1}{2} \left[\frac{\|a_n - x\|_2^2 - |u(x) - b_n|^2}{(u(x) - b_n) - \langle v, x - a_n \rangle} \right]$ for $n \neq m$ and $r_m \stackrel{\text{def}}{=} +\infty$. Here in the definition of r the min is taken only over n for which $r_n > 0$ and $r_n \geq b_n - u_m(x)$. Also, for $c \in \mathbb{R}$ we define $c/0 = +\infty$.

The condition on a_n, b_n is imposed to avoid inconvenient cases when the center of a funnel lies on some another funnel. In practice, we compute the transport rays for a batch of points x with tensor operations. We provide an example of transport rays of a random MinFunnel u in Figure 9.

Part 3. Defining the ray monotone map. Let \mathbb{P} be a distribution on \mathbb{R}^D and u be a MinFunnel. We aim to construct a u -ray monotone map $T : \mathbb{R}^D \rightarrow \mathbb{R}^D$ and define a distribution $\mathbb{Q} = T\#\mathbb{P}$.

If $\nabla u(x)$ does not exist, we define $T(x) = x$ (the set of such points is μ_L -negligible). Otherwise, since u is a MinFunnel, we have $\|\nabla u(x)\|_2 = 1$. Thus, the u -ray forward map T may take any value $T(x) \in [x_0, x]$, where x_0 is the left (lower) endpoint of $\text{ray}(x) = [x_0, x_1]$. According to the analysis in §3.1, any (measurable) map T defined by this principle is u -ray monotone. For such a map T one may put $\mathbb{Q} = T\#\mathbb{P}$. As a result, for the pair (\mathbb{P}, \mathbb{Q}) , function T is an OT map (Corollary 1), possibly non-unique. Thus, the ground truth $\mathbb{W}_1(\mathbb{P}, \mathbb{Q})$ can be estimated from samples $x \sim \mathbb{P}$.

As we are also interested in testing how well OT dual solvers recover the OT **gradient** ∇f^* , we need this gradient to exist and be \mathbb{P} -unique. From Proposition 1 we know that $f^* = u$ is an optimal potential. However, it is not necessarily unique (up to a constant). For example, in the trivial case $T(x) \equiv x$ and $\mathbb{P} = \mathbb{Q}$, any 1-Lipschitz f^* is optimal and ∇f^* is not unique. We show that with mild assumptions on \mathbb{P}, T, u , the gradient can be designed to be unique and match $\nabla u(x)$.

Proposition 4 (Uniqueness of the OT gradient). *Let \mathbb{P} be absolutely continuous and let $u : \mathbb{R}^D \rightarrow \mathbb{R}$ be 1-Lipschitz. Let T be a u -ray-monotone map for which $T(x) \neq x$ holds \mathbb{P} -almost surely, i.e., it moves every piece of mass of \mathbb{P} . Define $\mathbb{Q} = T\#\mathbb{P}$. Then for every optimal f^* which maximizes (5) the equality $\nabla f^*(x) = \nabla u(x)$ holds \mathbb{P} -almost surely, i.e., the OT gradient ∇u is \mathbb{P} -unique.*

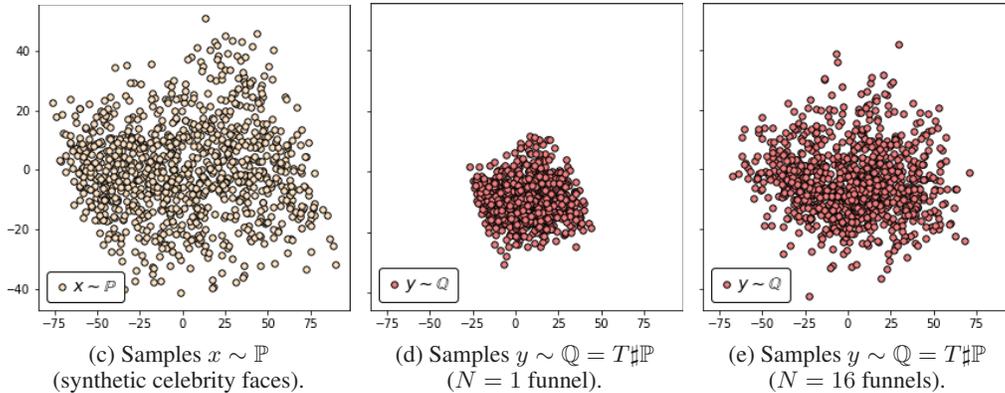
3.3 Benchmark Pairs

We use our methodology (§3.2) to construct **high-dimensional** ($D = 2, 2^2, \dots, 2^7$) benchmark pairs and benchmark pairs on the space of 64×64 RGB **images** of celebrity faces ($D = 12288$).

For convenience, we use absolutely continuous \mathbb{P} supported on a hypercube $\mathcal{S} = [-B, B]^D \subset \mathbb{R}^D$. In particular, for points $x \in \mathcal{S}$, we consider transport rays *truncated* to \mathcal{S} , i.e., $\text{ray}(x) \cap \mathcal{S}$. We



(a) Samples from the pair with $(N, p) = (1, 10)$. (b) Samples from the pair with $(N, p) = (16, 100)$.



(c) Samples $x \sim \mathbb{P}$ (synthetic celebrity faces). (d) Samples $y \sim \mathbb{Q} = T\#\mathbb{P}$ ($N = 1$ funnel). (e) Samples $y \sim \mathbb{Q} = T\#\mathbb{P}$ ($N = 16$ funnels).

Figure 5: Visualization of random samples from our constructed Celeba images benchmark pairs. In the last three plots, we show distributions projected to 2 principal components of \mathbb{P} .

construct the u -ray monotone map $T : \mathbb{R}^D \rightarrow \mathbb{R}^D$ by the following principle. Let $[x_0, x_1] \subset \mathcal{S}$ be the truncated transport ray of u for $x \in \mathcal{S}$. We pick a parameter $p > 1$ and define the map T as follows:

$$T(x) \stackrel{\text{def}}{=} \left[\frac{\|x - x_0\|_2}{\|x_0 - x_1\|_2} \right]^p x_0 + \left(1 - \left[\frac{\|x - x_0\|_2}{\|x_0 - x_1\|_2} \right]^p \right) x_1. \quad (12)$$

This is a power function, i.e., if a ray is parametrized as $[0, 1]$, it moves the mass along the ray by $t \mapsto t^p$. Since \mathbb{P} -almost all the points in \mathcal{S} belong to (truncated) rays, we have $T(x) \neq x$ on \mathcal{S} .

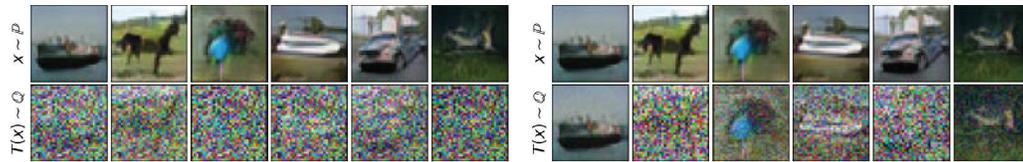
High-dimensional benchmark pairs. In dimensions $D = 2, 2^2, \dots, 2^7$, we put \mathbb{P} to be the standard uniform distribution on $\mathcal{S} = [-2.5, 2.5]^D$. In each dimension D , we consider $N = 4, 16, 64, 256$ funnels and pick random parameters $a_n \sim \text{Uniform}([-2.5, 2.5]^D)$ and $b_n \sim \mathcal{N}(0, 0.1)$. For this initialization of a_n, b_n , the assumptions of Proposition 3 hold with probability 1. The random seed is hardcoded. We use $p = 8$. For the case $D = 2, N = 4$, we visualize the input distribution \mathbb{P} , the function u , the constructed map T and the output distribution $\mathbb{Q} = T\#\mathbb{P}$ in Figure 3. We show examples of constructed (\mathbb{P}, \mathbb{Q}) for higher dimensions in Figure 4.

Images benchmark pairs for CIFAR-10 and Celeba. As \mathbb{P} we consider the synthetic distributions of generated 32×32 RGB images ($D = 3072$) and 64×64 RGB images ($D = 12288$). To generate these images, we use the WGAN-QC [28] generator model trained on CIFAR-10 [26] and Celeba [30] datasets, respectively. For CIFAR-10, we train WGAN-QC by using its publicly available code.² For Celeba, we pick a readily available pre-trained generator from the related **Wasserstein-2 benchmark**³ [23, §4.1]. To make \mathbb{P} absolutely continuous, we add the Gaussian noise with axis-wise $\sigma = 0.01$. Then we truncate the distribution to $\mathcal{S} = [-1.1, 1.1]^D$, i.e., we reject samples which are out of \mathcal{S} . We construct two benchmark pairs per each dataset (Figures 5, 6) with $(N, p) = (1, 10), (16, 100)$ and $a_n \sim \text{Uniform}([-1., 1.]^D), b_n \sim \mathcal{N}(0, 0.1)$.

In both high-dimensional and images pairs (\mathbb{P}, \mathbb{Q}) , the input \mathbb{P} is an absolutely continuous distribution supported on a hypercube \mathcal{S} . By the design, the distribution \mathbb{P} is accessible by random samples. To sample from \mathbb{Q} , we first sample $x \sim \mathbb{P}$, then compute its transport ray (Proposition 3), truncate it to \mathcal{S} , and produce $y = T(x) \sim \mathbb{Q}$ by (12). The sampler for \mathbb{Q} outputs only y , i.e., information about u, T, x is hidden from the user and employed only when estimating the ground truth $\mathbb{W}_1(\mathbb{P}, \mathbb{Q})$ or ∇u .

²<https://github.com/harryliew/WGAN-QC>

³<https://github.com/iamalexkorotin/Wasserstein2Benchmark>



(a) Samples from the pair with $(N, p) = (1, 10)$. (b) Samples from the pair with $(N, p) = (16, 100)$.

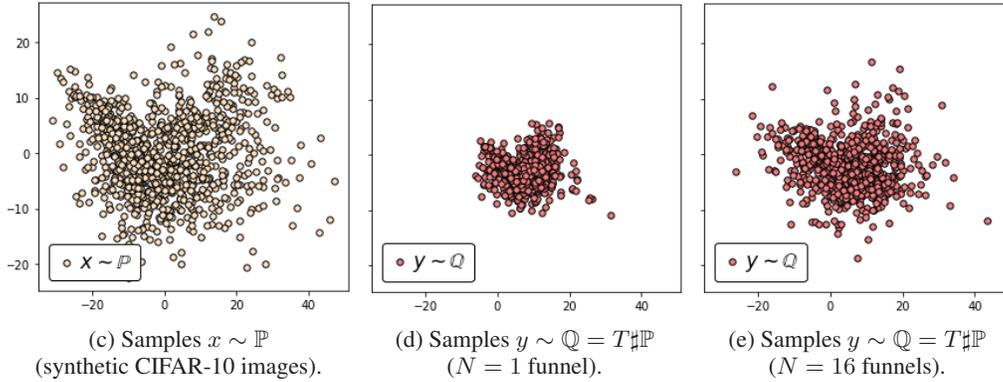


Figure 6: Visualization of random samples from our constructed CIFAR-10 images benchmark pairs. In the last three plots, we show distributions projected to 2 principal components of \mathbb{P} .

Reversed pairs. Doing preliminary tests, we found that for solvers it is more challenging to compute the OT gradient for pair (\mathbb{Q}, \mathbb{P}) rather than (\mathbb{P}, \mathbb{Q}) . In particular, in our images benchmark, the pair (\mathbb{Q}, \mathbb{P}) reflects the practical WGAN scenario better. Indeed, in WGANs, the solvers move the generated distribution (bad images, \mathbb{Q} in our construction) to the real distribution (good images, \mathbb{P}). Recall that $\mathbb{Q} = T\#\mathbb{P}$, where T is a differentiable bijection (12) along the (truncated) transport rays of a MinFunnel u . As a result, \mathbb{Q} is absolutely continuous and the OT gradient for the reverse (\mathbb{Q}, \mathbb{P}) is \mathbb{Q} -unique (Proposition 4). For this pair, the optimal potential is $-u$ and its gradient is $-\nabla u$. To conclude, in all the experiments, we feed the **reverse pair** $(\mathbb{P}, \mathbb{Q}) := (\mathbb{Q}, \mathbb{P})$ to OT solvers in view.

4 Evaluation of Dual Solvers and Discussion

In this section, we evaluate WGAN dual OT solvers (§2) on our constructed benchmark (§3). Technical details of the implementation are given in Appendix B. The code is written in PyTorch framework and is publicly available together with all the constructed benchmark distributions at

<https://github.com/justkolesov/Wasserstein1Benchmark>

Metrics. For \mathbb{W}_1 , we do not use any specific metric but simply report obtained $\widehat{\mathbb{W}}_1$ and the ground truth \mathbb{W}_1 . To quantify the recovered gradient $\nabla \hat{f}$, we use \mathcal{L}^2 and cosine similarity metrics [23, §4.2]:

$$\mathcal{L}^2(\nabla \hat{f}, \nabla f^*) \stackrel{\text{def}}{=} \|\nabla \hat{f} - \nabla f^*\|_{\mathcal{L}^2}^2; \quad \cos(\nabla \hat{f}, \nabla f^*) \stackrel{\text{def}}{=} \frac{\langle \nabla \hat{f}, \nabla f^* \rangle_{\mathcal{L}^2}}{\|\nabla \hat{f}\|_{\mathcal{L}^2} \cdot \|\nabla f^*\|_{\mathcal{L}^2}}, \quad (13)$$

where $\langle \nabla f_1, \nabla f_2 \rangle_{\mathcal{L}^2} \stackrel{\text{def}}{=} \int \langle \nabla f_1(x), \nabla f_2(x) \rangle d\mathbb{P}(x)$, $\|\nabla f\|_{\mathcal{L}^2}^2 \stackrel{\text{def}}{=} \langle \nabla f, \nabla f \rangle_{\mathcal{L}^2}$ and ∇f^* is the ground truth OT gradient. The \mathcal{L}^2 metric compares the gradients $\nabla \hat{f}, \nabla f^*$ as elements of the space $\mathcal{L}^2(\mathbb{P})$ of quadratically integrable w.r.t. \mathbb{P} functions. The cosine compares their directions regardless of the magnitude (Figure 13). To estimate the metrics, we use 2^{13} samples $x \sim \mathbb{P}$.

[DOT] For completeness, we add the empirical (batched) OT [9, 10, 36, 37] to evaluation. For batches $X \sim \mathbb{P}, Y \sim \mathbb{Q}$, we use $\widehat{\mathbb{W}}_1(X, Y)$ computed by a **discrete** solver as an estimate of $\mathbb{W}_1(\mathbb{P}, \mathbb{Q})$. The solver can be combined with the automatic differentiation to approximate ∇f^* on the batch X .

High-dimensional pairs. We test the solvers and report the estimated \mathbb{W}_1 value and metrics (13) in Tables 9, 10, 11 (Appendix C). We use fully-connected nets as potentials f_θ, g_ω and movers T_θ, H_ω (in the maximin solvers). In Figure 7, we show the potentials learned by solvers for $D = 2, N = 4$.

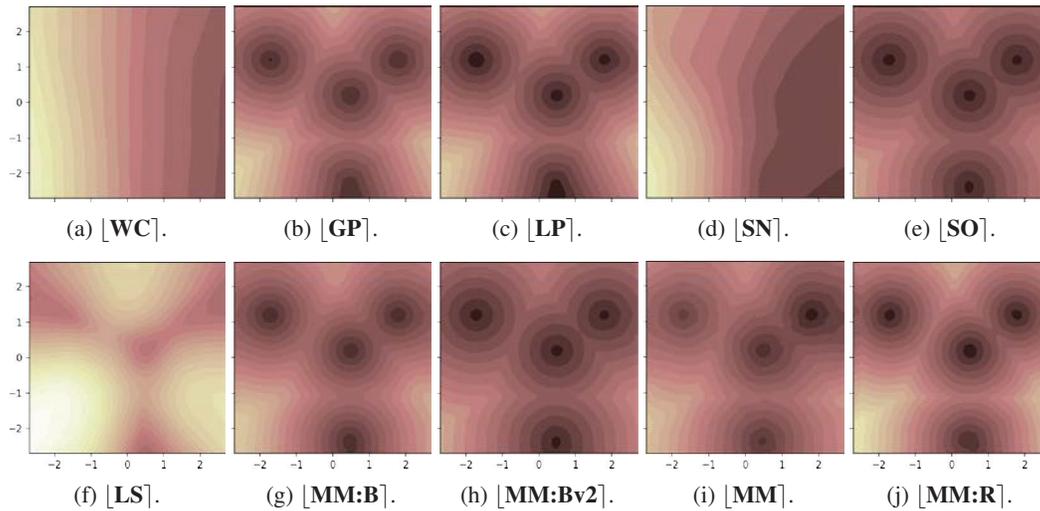


Figure 7: Surfaces of potentials f_θ learned by OT solvers on the pair with $D = 2, N = N$. For $[\mathbf{MM}:\mathbf{R}]$ we plot (minus) second potential $-g_\omega$ as the solver does not compute the first one. The ground truth optimal potential is in Figure 3b.

As Table 11 shows, all the solvers randomly over/underestimate \mathbb{W}_1 , typically with notable error. Consequently, *none of the solvers can be viewed as precise estimators of the OT cost*. Thus, our discussion below primarily focuses on the estimation of the OT gradient needed in WGANs.

Original $[\mathbf{WC}]$ leads to a pathological value surface of the learned potential (Figure 7a), which is far from the ground truth. This agrees with the claims of [17, §3]. Despite this, even in high dimensions the cos metric is positive (Table 9a), i.e., the recovered gradient correlates with the OT gradient. Popular $[\mathbf{GP}]$ notably improves upon $[\mathbf{WC}]$ and provides higher cos values (Table 9b). According to the results, its modification $[\mathbf{LP}]$ does not provide any major improvement (Table 9c).

Surprisingly, $[\mathbf{SN}]$ performs worse than $[\mathbf{GP}]$ and is comparable to $[\mathbf{WC}]$ in cos metric (Table 9d). We presume that this happens since the spectral norm negatively affects the expressive power of the neural net [1]. Even when $D = 2$ (Figure 7d), $[\mathbf{SN}]$ fails to recover the optimal potential. $[\mathbf{SO}]$ replaces spectral norm with orthogonalization and uses GroupSort activations which improve performance. The solver scores higher cos values (Table 9e) which are comparable to $[\mathbf{GP}]$.

Solver $[\mathbf{LS}]$ recovers biased potential since (7) is the duality formula for regularized OT which yields biased optimal potentials (Table 9f). The bias is huge for large D, N when the benchmark pairs are complex. This is analogous to results for the same solver in the Wasserstein-2 (\mathbb{W}_2) benchmark [23].

Batch-based $[\mathbf{MM}:\mathbf{B}]$ suffers from the bias in high dimensions (Table 9g) due to the overestimation of the value of the inner problem in (8). Note that cos metric values are even negative in high dimensions. The same is reported in the \mathbb{W}_2 benchmark [23]. $[\mathbf{MM}:\mathbf{Bv2}]$ uses a more tricky optimization scheme and yields high cos values (Table 9h). It captures the direction of the OT gradient but *extremely* overestimates its magnitude, see high values of \mathcal{L}^2 in Table 10h.

Maximin solvers $[\mathbf{MM}]$, $[\mathbf{MM}:\mathbf{R}]$ perform comparably to $[\mathbf{GP}]$, $[\mathbf{SO}]$, $[\mathbf{LP}]$, see Tables 9i, 9j. However, their training takes longer and sometimes diverges as it solves a saddle point problem. This agrees with the \mathbb{W}_2 benchmark [23, §4.3]. Using these solvers in GANs is not easy as it yields a challenging *min-max-min* optimization problem. Importantly, $[\mathbf{MM}:\mathbf{R}]$ recovers the OT map which can itself be used as a generative model (*outside* the context of GANs) in computer vision tasks. Here we refer the reader to the recent *neural optimal transport* (NOT) methods [25, 24, 4, 22, 11, 44, 8].

Empirical $[\mathbf{DOT}]$ provides precise estimates of \mathbb{W}_1 (Table 11k) and the OT gradient (Table 9k) only in small dimensions. In high dimensions, it intolerably overestimates \mathbb{W}_1 ; its gradient is almost orthogonal to the ground truth ($\cos \lesssim 0$). This is due to the exponential (in D) sample complexity of DOT [53]. Thus, (unregularized) DOT is an imprecise estimator of \mathbb{W}_1 or the OT gradient in high D .

Images pairs. Here we do not consider $[\mathbf{SO}]$ as its authors do not provide convolutional architectures with GroupSort and orthonormalization. We use DCGAN [42] as potentials f_θ, g_ω . In $[\mathbf{MM}]$ and $[\mathbf{MM}:\mathbf{R}]$, the movers T_θ, H_ω are UNets [43]. The evaluation results on Celeba benchmark pairs are given in Tables 6, 7, 8 and on CIFAR-10 benchmark pairs – in Tables 3, 4, 5 (Appendix C).

		[WC]	[GP]	[LP]	[SN]	[SO]	[LS]	[MM:B]	[MM:Bv2]	[MM]	[MM:R]	[DOT]
\cos	HD	☹️	😊	😊	☹️	😊	☹️	😊	😊	😊	😊	☹️
	IMG	😊	😊	😊	😊	-	😊	😊	😊	☹️	😊	😊
\mathcal{L}^2	HD	☹️	😊	😊	☹️	😊	☹️	😊	😊	☹️	😊	☹️
	IMG	☹️	☹️	☹️	☹️	-	☹️	☹️	☹️	☹️	😊	☹️
\mathbb{W}_1	HD	☹️	😊	😊	☹️	😊	☹️	☹️	😊	😊	😊	☹️
	IMG	☹️	☹️	☹️	☹️	-	😊	☹️	😊	😊	😊	😊

Table 1: The summary of WGAN dual OT solvers’ performance in \cos , \mathcal{L}^2 and \mathbb{W}_1 metrics on our high-dimensional (HD) and images (IMG) benchmark pairs. For details, see Appendix C.

Surprisingly, our images benchmark pairs turned to be simpler than some high-dimensional benchmark pairs. Although (\mathbb{P}, \mathbb{Q}) are absolutely continuous and supported on $[-1.1, 1.1]^D$, their actual probability mass is still concentrated around small low-dimensional sub-manifold of data. We suppose that this is one of the causes for the reasonable performance of most solvers. In particular, we see that even [DOT], [LS], [MM:B] score $\cos > 0$ in images benchmark pairs, although they struggled to produce good results on high-dimensional pairs.

Solvers [GP], [LP], [MM:R] provide very high $\cos > 0.9$ (Table 6). However, only [MM:R] provides precise approximation of ∇f^* in \mathcal{L}^2 norm (Table 7). Interestingly, maximin [MM] diverges on our images benchmark pairs (tuning the hyper-parameters did not help). Similar to the evaluation in high-dimensional pairs, [WC] and [SN] show moderate $\cos > 0$, but its value is notably smaller than that of the top-performing methods. Solver [MM:Bv2] demonstrates meaningful estimate of \mathbb{W}_1 (Table 8). Nevertheless, its recovered gradient is almost orthogonal to the ground truth ($\cos \approx 0$, see Table 6), and the values of \mathcal{L}^2 metric are **extremely** high (Table 7).

5 Discussion

Our methodology creates pairs of continuous distributions with known ground truth OT cost and gradient, filling the missing gap of benchmarking \mathbb{W}_1 dual solvers. This development allows us to evaluate the OT performance of WGAN dual methods. The experimental results are summarized in Table 1. Our evaluation shows that these solvers should **not** be considered as **meaningful estimators** of \mathbb{W}_1 as they exhibit large error. However, the OT gradient recovered by these solvers shows positive \cos with ground truth. This suggests that most methods could still be used to **minimize** \mathbb{W}_1 in variational problems, e.g., Wasserstein GANs.

Computational complexity. Evaluation of all the OT solvers on our high-dimensional and images benchmark pairs takes less than 50 hours on a single GPU GTX 1080ti (11 GB VRAM).

Potential Impact. Our benchmark distributions can be used to evaluate future dual OT solvers in high-dimensional spaces, a crucial step to improve the transparency and replicability of OT and WGAN-related research. We expect our benchmark to become a standard benchmark for \mathbb{W}_1 optimal transport as part of the ongoing effort of advancing computational OT.

Limitations (benchmark). We rely on MinFunnels as optimal Kantorovich potentials to generate benchmark pairs. Also, we limit our pairs to be absolutely continuous distributions. It is unclear whether our benchmark sufficiently reflects the real-world scenarios in which the WGAN solvers are used. Nevertheless, our methodology is generic and can be used to construct new benchmark pairs.

Limitations (evaluation). We evaluate how well the OT solvers compute OT cost and gradient but do not assess their performance in GAN settings. Studying this question is a promising future research avenue which could help to develop new OT-based methods for generative modeling.

ACKNOWLEDGEMENTS. The work was supported by the Analytical center under the RF Government (subsidy agreement 000000D730321P5Q0002, Grant No. 70-2021-00145 02.11.2021).

References

- [1] Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301. PMLR, 2019.
- [2] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.
- [3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.
- [4] Arip Asadulaev, Alexander Korotin, Vage Egiazarian, and Evgeny Burnaev. Neural optimal transport with general cost functionals. *arXiv preprint arXiv:2205.15403*, 2022.
- [5] Artem Chernodub and Dimitri Nowicki. Norm-preserving orthogonal permutation linear unit activation functions (oplu). *arXiv preprint arXiv:1604.02313*, 2016.
- [6] Grady Daniels, Tyler Maunu, and Paul Hand. Score-based generative neural networks for large-scale optimal transport. *Advances in Neural Information Processing Systems*, 34, 2021.
- [7] Lawrence C Evans and Wilfrid Gangbo. *Differential equations methods for the Monge-Kantorovich mass transfer problem*. Number 653. American Mathematical Soc., 1999.
- [8] 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.
- [9] 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.
- [10] Kilian Fatras, Younes Zine, Szymon Majewski, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Minibatch optimal transport distances; analysis and applications. *arXiv preprint arXiv:2101.01792*, 2021.
- [11] Milena Gazdieva, Litu Rout, Alexander Korotin, Andrey Kravchenko, Alexander Filippov, and Evgeny Burnaev. An optimal transport perspective on unpaired image super-resolution. *arXiv preprint arXiv:2202.01116*, 2022.
- [12] Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. In *Advances in neural information processing systems*, pages 3440–3448, 2016.
- [13] Aude Genevay, Gabriel Peyré, and Marco Cuturi. Gan and vae from an optimal transport point of view. *arXiv preprint arXiv:1706.01807*, 2017.
- [14] Gene H Golub and Henk A Van der Vorst. Eigenvalue computation in the 20th century. *Journal of Computational and Applied Mathematics*, 123(1-2):35–65, 2000.
- [15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [16] Luca Granieri. On a distance representation of kantorovich potentials. *Applied mathematics letters*, 22(4):605–610, 2009.
- [17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.
- [18] 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. In *Advances in neural information processing systems*, pages 6626–6637, 2017.
- [19] Antoine Houdard, Arthur Leclaire, Nicolas Papadakis, and Julien Rabin. On the existence of optimal transport gradient for learning generative models. *arXiv preprint arXiv:2102.05542*, 2021.

- [20] Leonid Kantorovitch. On the translocation of masses. *Management Science*, 5(1):1–4, 1958.
- [21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [22] Alexander Korotin, Vage Egiazarian, Lingxiao Li, and Evgeny Burnaev. Wasserstein iterative networks for barycenter estimation. *arXiv preprint arXiv:2201.12245*, 2022.
- [23] Alexander Korotin, Lingxiao Li, Aude Genevay, Justin M Solomon, Alexander Filippov, and Evgeny Burnaev. Do neural optimal transport solvers work? a continuous wasserstein-2 benchmark. *Advances in Neural Information Processing Systems*, 34:14593–14605, 2021.
- [24] Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Kernel neural optimal transport. *arXiv preprint arXiv:2205.15269*, 2022.
- [25] Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Neural optimal transport. *arXiv preprint arXiv:2201.12220*, 2022.
- [26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [27] Dohyun Kwon, Yeoneung Kim, Guido Montúfar, and Insoon Yang. Training wasserstein gans without gradient penalties. *arXiv preprint arXiv:2110.14150*, 2021.
- [28] Huidong Liu, Xianfeng Gu, and Dimitris Samaras. Wasserstein GAN with quadratic transport cost. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4832–4841, 2019.
- [29] Huidong Liu, GU Xianfeng, and Dimitris Samaras. A two-step computation of the exact gan wasserstein distance. In *International Conference on Machine Learning*, pages 3159–3168. PMLR, 2018.
- [30] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- [31] Anton Mallasto, Jes Frelsen, Wouter Boomsma, and Aasa Feragen. (q, p)-Wasserstein GANs: Comparing ground metrics for Wasserstein GANs. *arXiv preprint arXiv:1902.03642*, 2019.
- [32] Anton Mallasto, Guido Montúfar, and Augusto Gerolin. How well do WGANs estimate the Wasserstein metric? *arXiv preprint arXiv:1910.03875*, 2019.
- [33] Paul Milgrom and Ilya Segal. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601, 2002.
- [34] Tristan Milne and Adrian Nachman. Wasserstein gans with gradient penalty compute congested transport. *arXiv preprint arXiv:2109.00528*, 2021.
- [35] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.
- [36] Khai Nguyen, Dang Nguyen, Tung Pham, and Nhat Ho. Improving mini-batch optimal transport via partial transportation. *arXiv preprint arXiv:2108.09645*, 2021.
- [37] Khai Nguyen, Quoc Nguyen, Nhat Ho, Tung Pham, Hung Bui, Dinh Phung, and Trung Le. Bomb-ot: On batch of mini-batches optimal transport. *arXiv e-prints*, pages arXiv–2102, 2021.
- [38] 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.
- [39] Henning Petzka, Asja Fischer, and Denis Lukovnicov. On the regularization of wasserstein gans. *arXiv preprint arXiv:1709.08894*, 2017.
- [40] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

- [41] Thomas Pinetz, Daniel Soukup, and Thomas Pock. On the estimation of the Wasserstein distance in generative models. In *German Conference on Pattern Recognition*, pages 156–170. Springer, 2019.
- [42] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [44] Litu Rout, Alexander Korotin, and Evgeny Burnaev. Generative modeling with optimal transport maps. In *International Conference on Learning Representations*, 2021.
- [45] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in neural information processing systems*, pages 2234–2242, 2016.
- [46] Maziar Sanjabi, Jimmy Ba, Meisam Razaviyayn, and Jason D Lee. On the convergence and robustness of training GANs with regularized optimal transport. *arXiv preprint arXiv:1802.08249*, 2018.
- [47] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.
- [48] Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large-scale optimal transport and mapping estimation. *arXiv preprint arXiv:1711.02283*, 2017.
- [49] Jan Stanczuk, Christian Etmann, Lisa Maria Kreusser, and Carola-Bibiane Schönlieb. Wasserstein gans work because they fail (to approximate the wasserstein distance). *arXiv preprint arXiv:2103.01678*, 2021.
- [50] Akinori Tanaka. Discriminator optimal transport. *Advances in Neural Information Processing Systems*, 32, 2019.
- [51] Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.
- [52] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.
- [53] Jonathan Weed, Francis Bach, et al. Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. *Bernoulli*, 25(4A):2620–2648, 2019.
- [54] Xiang Wei, Boqing Gong, Zixia Liu, Wei Lu, and Liqiang Wang. Improving the improved training of wasserstein gans: A consistency term and its dual effect. *arXiv preprint arXiv:1803.01541*, 2018.

Checklist

1. For all authors...
 - (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?
[Yes]
 - (b) Did you describe the limitations of your work?
[Yes]
 - (c) Did you discuss any potential negative societal impacts of your work?
[N/A]
 - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
 - (a) Did you state the full set of assumptions of all theoretical results?
[Yes] All the assumptions are stated in the main text.
 - (b) Did you include complete proofs of all theoretical results?
[Yes] All the proofs are given in the appendices.
3. If you ran experiments...
 - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)?
[Yes] The code and the instructions are included in the supplementary material. The datasets and assets that we use are publicly available.
 - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)?
[Yes] See the supplementary material (appendices + code).
 - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)?
[No]
 - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)?
[Yes] See the Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
 - (a) If your work uses existing assets, did you cite the creators?
[Yes] See the discussion in section §3.3.
 - (b) Did you mention the license of the assets?
[No] We refer to the datasets' and assets' public pages.
 - (c) Did you include any new assets either in the supplemental material or as a URL?
[Yes] See the supplementary material
 - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating?
[N/A]
 - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content?
[No] For CelebA faces dataset, we refer to the original authors publication.
5. If you used crowdsourcing or conducted research with human subjects...
 - (a) Did you include the full text of instructions given to participants and screenshots, if applicable?
[N/A]
 - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable?
[N/A]
 - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation?
[N/A]