Orthogonal Estimation of Wasserstein Distances

Mark Rowland[*1] **Jiri Hron**[*1] **Yunhao Tang**[*2] **Krzysztof Choromanski**[3] **Tamas Sarlos**[4] **Adrian Weller**[1,5]
[1]University of Cambridge, [2]Columbia University, [3]Google Brain, [4]Google Research, [5]The Alan Turing Institute

Abstract

Wasserstein distances are increasingly used in a wide variety of applications in machine learning. Sliced Wasserstein distances form an important subclass which may be estimated efficiently through one-dimensional sorting operations. In this paper, we propose a new variant of sliced Wasserstein distance, study the use of orthogonal coupling in Monte Carlo estimation of Wasserstein distances and draw connections with stratified sampling, and evaluate our approaches experimentally in a range of large-scale experiments in generative modelling and reinforcement learning.

1 INTRODUCTION

Wasserstein distances are a method of measuring distance between probability distributions, and are widely used in image processing (Rabin et al., 2011; Bonneel et al., 2015), probability (Ambrosio et al., 2005), physics (Jordan et al., 1998) and economics (Galichon, 2016), and are increasingly used in machine learning (Arjovsky et al., 2017; Gulrajani et al., 2017; Peyré and Cuturi, 2018). Wasserstein distances are popular as they take into account spatial information (unlike total variation distance), and can be defined between continuous and discrete distributions (unlike the Kullback-Leibler divergence). These factors have made Wasserstein distances particularly popular in defining objectives for generative modelling (Arjovsky et al., 2017; Gulrajani et al., 2017).

However, exact computation of Wasserstein distances is costly, as it requires the solution of an optimal transport problem. This has motivated the development of a wide variety of computationally tractable approximations (see for example (Cuturi, 2013; Genevay et al., 2016)). The sliced Wasserstein distance was proposed by Rabin et al. (2011), and exploits the fact that one-dimensional instances of optimal transport problems can be solved much quicker than the general case, working directly with one-dimensional projections of the probability distributions in question; see Section 2 for a more detailed exposition.

Although the move from Wasserstein distance to sliced Wasserstein distance often yields significant gains in computational tractability, we highlight two issues that remain. Firstly, the focus of sliced Wasserstein distance on one-dimensional marginals of probability distributions can lead to poorer quality results than true Wasserstein distance (Bonneel et al., 2015). Secondly, the evaluation of sliced Wasserstein distance itself generally requires Monte Carlo estimation, and thus enough Monte Carlo samples must be used to control the random fluctuations introduced by this stochastic method.

In this paper, we investigate these two shortcomings of the sliced Wasserstein distance and propose a new distance which incorporates the computational benefits of the sliced Wasserstein distance, whilst still retaining information from the high-dimensional distances. Given recent strong results based on Wasserstein distances and their tractable approximations, exploring related methods is an important area of study. However, our initial empirical results demonstrate only small benefits in some contexts.

We study a Monte Carlo sampling scheme for more efficient estimation of both sliced Wasserstein distance, and our novel variant. We conclude this section by highlighting our main contributions:

- In Section 3, we introduce a new variant of Wasserstein distance, which we term *projected Wasserstein distance*, which incorporates aspects of both sliced Wasserstein distance and true Wasserstein distance.
- In Section 4, we study a Monte Carlo method for more accurate estimation of sliced Wasserstein and projected Wasserstein distances, based on orthogonal couplings of projection vectors, and provide the-

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s). *Equal contribution.

oretical analysis and connections to notions of stratified sampling.

- In Section 5, we empirically evaluate the performance of projected Wasserstein distance, and orthogonally-coupled estimation, on a variety of tasks, including high-dimensional generative modelling and reinforcement learning.

2 WASSERSTEIN AND SLICED WASSERSTEIN DISTANCES

We briefly review Wasserstein and sliced Wasserstein distances, and their role in modern machine learning. For a more detailed review of theory of Wasserstein distances and optimal transport more generally, see Villani (2008). In this paper, we consider these distances in the specific case of Euclidean base space $(\mathbb{R}^d, \|\cdot\|_2)$, and first recall the definition of Wasserstein distance at this level of generality.

Definition 2.1 (Wasserstein distances)**.** *Let $p \geq 1$. The p-Wasserstein distance is defined on $\mathscr{P}_p(\mathbb{R}^d)$, the set of probability distributions over $\mathbb{R}^d$ with finite p^{th} moment, by:*

$$\mathrm{W}_p(\eta, \mu) = \left[\inf_{\gamma \in \Gamma(\eta,\mu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|\mathbf{x} - \mathbf{y}\|_2^p \gamma(\mathrm{d}\mathbf{x}, \mathrm{d}\mathbf{y}) \right]^{1/p},$$

for all $\eta, \mu \in \mathscr{P}_p(\mathbb{R}^d)$, where $\Gamma(\eta,\mu) \subseteq \mathscr{P}(\mathbb{R}^d \times \mathbb{R}^d)$ is the set of joint distributions over $\mathbb{R}^d \times \mathbb{R}^d$ for which the marginal on the first d coordinates is η, and the marginal on the last d coordinates is μ. An element $\gamma \in \Gamma(\eta,\mu)$ achieving the infimum is called an optimal transport plan between η and μ.

A common problem in the image processing literature using Wasserstein distance is that of *computing a barycentre* for a collection of measures (Cuturi and Doucet, 2014; Staib et al., 2017). In this problem, a collection of measures $\{\mu_j\}_{j=1}^J \subseteq \mathscr{P}_p(\mathbb{R}^d)$ and a collection of weightings $(\lambda_j)_{j=1}^J \in \mathbb{R}^J_{>0}$ are given, and the following objective is posed:

$$\operatorname*{arg\,min}_{\eta \in \mathscr{Q}} \sum_{j=1}^J \lambda_j \mathrm{W}_p^p(\eta, \mu_j)\,. \tag{1}$$

In some sense, this generalises the notion of finding a centre of mass of a collection of points in Euclidean space to the "centre of mass" of a collection of probability measures. A special case of the barycentre problem which is common in the machine learning literature is that of *distribution learning*, for which $J = 1$, reducing Problem (1) to the following optimisation problem

$$\operatorname*{arg\,min}_{\eta \in \mathscr{Q}} \mathrm{W}_p^p(\eta, \mu)\,, \tag{2}$$

for a given $\mu \in \mathscr{P}_p(\mathbb{R}^d)$, and a space of probability distributions $\mathscr{Q} \subseteq \mathscr{P}_p(\mathbb{R}^d)$. A typical application is in deep generative modelling, in which μ is the empirical distribution corresponding to some dataset, and $\mathscr{Q}$ is a set of distributions parametrised by a neural network (Arjovsky et al., 2017).

One of the reasons that Wasserstein distances are not more commonly used in machine learning is that their evaluation requires solving an optimisation problem. As an example, computing $\mathrm{W}_p^p(\eta,\mu)$ in the particular case where

$$\eta = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{x}_m}\,,\ \mu = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{y}_m}\,, \tag{3}$$

is expressed as the following linear program:

$$\max_{\mathbf{T} \in \mathscr{B}_M} \langle \mathbf{T}, \mathbf{D} \rangle / M\,. \tag{4}$$

Here, Dirac delta function $\delta_{\mathbf{x}_m}$ denotes the probability density function of idealized point mass at sample data point $\mathbf{x}_m$, $\mathbf{D} \in \mathbb{R}^{M\times M}$ specifies costs between elements of the supports of η and μ, so that $\mathbf{D}_{ij} = \|\mathbf{x}_i - \mathbf{y}_j\|_2^p$ for all $i, j \in [M]$, and $\mathscr{B}_M = \{\mathbf{A} \in \mathbb{R}^{M\times M} | \mathbf{A}\mathbf{1} = \mathbf{1},\ \mathbf{A}^\top \mathbf{1} = \mathbf{1}\}$ is the Birkhoff polytope. For linear programs of this specific form, the most efficient known solution methods are matching algorithms, which can achieve $\mathcal{O}(M^{5/2}\log M)$ complexity; for large M, this cost can quickly become infeasible as an inner loop in a learning algorithm. Therefore (Arjovsky et al., 2017) *heuristically* optimized equivalent dual form of W_p^p.

In contrast, for *one-dimensional* probability distributions $\eta, \mu \in \mathscr{P}(\mathbb{R})$, the optimal transport plan $\gamma^\star \in \Gamma(\eta,\mu)$ may be written down analytically, and in the case of finitely-supported distributions, the computation of Wasserstein distance amounts to sorting the support of the distributions in question on the real line, with a complexity of only $\mathcal{O}(M \log M)$. This motivates an alternative to the Wasserstein distance which exploits the computational ease of optimal transport on the real line: *sliced Wasserstein distances* (Rabin et al., 2011). For a vector $\mathbf{v} \in S^{d-1}$, where S^{d-1} is the unit sphere in $\mathbb{R}^d$, we define the projection map $\Pi_{\mathbf{v}} : \mathbb{R}^d \to \mathbb{R}$ by $\Pi_{\mathbf{v}}(\mathbf{x}) = \langle \mathbf{v}, \mathbf{x} \rangle$, for all $\mathbf{x} \in \mathbb{R}^d$, and denote projection of distribution η by $(\Pi_{\mathbf{v}})_\# \eta$. With this notion established, we may now precisely define sliced Wasserstein distances.

Definition 2.2 (Sliced Wasserstein distances)**.** *Let $p \geq 1$. The p-sliced Wasserstein distance is defined on $\mathscr{P}_p(\mathbb{R}^d)$ by*

$$\mathrm{SW}_p^p(\eta,\mu) = \mathbb{E}_{\mathbf{v}\sim \mathrm{Unif}(S^{d-1})}\left[\mathrm{W}_p^p((\Pi_{\mathbf{v}})_\#\eta, (\Pi_{\mathbf{v}})_\#\mu) \right], \tag{5}$$

for all $\eta, \mu \in \mathscr{P}_p(\mathbb{R}^d)$.

Due to the intractable expectation appearing in Equation (5), the sliced Wasserstein distance is typically estimated via Monte Carlo sampling

$$\widehat{\mathrm{SW}}_p^p(\eta, \mu) = \frac{1}{N} \sum_{n=1}^{N} \mathrm{W}_p^p((\Pi_{\mathbf{v}_n})_{\#}\eta, (\Pi_{\mathbf{v}_n})_{\#}\mu) , \quad (6)$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_N \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}(S^{d-1})$. The precise algorithmic steps for this estimation in the case of empirical distributions $\eta = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{x}_m}$, $\mu = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{y}_m}$ are given in Algorithm 1.

Algorithm 1 Sliced Wasserstein estimation

```
Require: $\eta = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{x}_m}$,  $\mu = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{y}_m}$
 1: for $n = 1$ to $N$ do
 2:     Sample $\mathbf{v}_n \sim \mathrm{Unif}(S^{d-1})$.
 3:     Compute projected distributions:
 4:         $(\Pi_{\mathbf{v}_n})_{\#}\eta = \frac{1}{M}\sum_{m=1}^M \delta_{\langle \mathbf{v}_n, \mathbf{x}_m \rangle}$
 5:         $(\Pi_{\mathbf{v}_n})_{\#}\mu = \frac{1}{M}\sum_{m=1}^M \delta_{\langle \mathbf{v}_n, \mathbf{y}_m \rangle}$
 6:     Compute sorted lists of supports:
 7:         $(w_m)_{m=1}^M \leftarrow \mathtt{sort}((\langle \mathbf{v}_n, \mathbf{x}_m \rangle)_{m=1}^M)$
 8:         $(z_m)_{m=1}^M \leftarrow \mathtt{sort}((\langle \mathbf{v}_n, \mathbf{y}_m \rangle)_{m=1}^M)$
 9:     Compute one-dimensional Wasserstein distance:
10:         $\mathrm{W}_p^p((\Pi_{\mathbf{v}_n})_{\#}\eta, (\Pi_{\mathbf{v}_n})_{\#}\mu) = \frac{1}{M}\sum_{m=1}^M |w_m - z_m|^p$
11: end for
12: return $\frac{1}{N}\sum_{n=1}^N \mathrm{W}_p^p((\Pi_{\mathbf{v}_n})_{\#}\eta, (\Pi_{\mathbf{v}_n})_{\#}\mu)$
```

Sliced Wasserstein distances were originally proposed for image processing applications (Rabin et al., 2011; Bonneel et al., 2015) to avoid expensive computation with true Wasserstein distances. More recently, sliced Wasserstein distances have also been used in deep generative modelling (Kolouri et al., 2018; Deshpande et al., 2018; Wu et al., 2017).

3 THE PROJECTED WASSERSTEIN DISTANCE

Whilst sliced Wasserstein distances bypass the computational bottleneck for Wasserstein distances (namely, solving the linear program in Problem (4)) required for each evaluation, they exhibit different behaviour from true Wasserstein distance, which in many cases may be undesirable. We offer an intuition as to why the qualitative properties of sliced Wasserstein and true Wasserstein distances differ. Inspecting Algorithm 1, we note that within the main loop, the random vector $\mathbf{v}_n$ plays *two* roles: firstly, as the determiner of the matching between the projected particles $((\langle \mathbf{v}_n, \mathbf{x}_m \rangle))_{m=1}^M$, $((\langle \mathbf{v}_n, \mathbf{y}_m \rangle))_{m=1}^M$; and secondly, in the computation of the distances between the projected particles. Roughly speaking, this may be thought of as introducing some type of "bias". This is similar in flavour to the phenomenon observed by Hasselt (2010) in the context of Q-learning, in which the maximum of a collection of samples is shown to be biased (over)estimate of the maximum of the corresponding population means. Indeed, Hasselt (2010) observes that this phenomenon has a long history (see e.g. Smith and Winkler (2006); Harrison and March (1984)).

Suppose we were to use separate projections to compute the distances at Line 10 of Algorithm 1. More precisely, suppose we sample $\mathbf{v}'_n \sim \mathrm{Unif}(S^{d-1})$ independently of $\mathbf{v}_n$, and introduce the notation $\sigma_{\mathbf{v}_n} : [M] \to [M]$ for the bijective mapping with the property that $\langle \mathbf{v}_n, \mathbf{x}_i \rangle < \langle \mathbf{v}_n, \mathbf{x}_j \rangle \implies \langle \mathbf{v}_n, \mathbf{y}_{\sigma_{\mathbf{v}_n}(i)} \rangle \le \langle \mathbf{v}_n, \mathbf{y}_{\sigma_{\mathbf{v}_n}(j)} \rangle$. Now consider replacing Line 10 of Algorithm 1 with the following computation:

$$\frac{1}{M} \sum_{m=1}^{M} |\langle \mathbf{v}'_n, \mathbf{x}_i \rangle - \langle \mathbf{v}'_n, \mathbf{y}_{\sigma_{\mathbf{v}_n}(i)} \rangle|^p , \quad (7)$$

noting that in degenerate cases there may exist more than one such $\sigma_{\mathbf{v}}$, in which case we select uniformly from this set of induced couplings. The computation in Expression (7) removes the source of "bias" identified previously.

Further, we observe that in the special case of $p = 2$, the use of a second projection to compute the costs can itself be interpreted as an unbiased estimator (up to a multiplicative scaling) of the original pairwise costs themselves. This motivates the *projected Wasserstein distance*, which we define formally below, along with the prerequisite notion of *induced couplings.*

Definition 3.1 (Induced couplings). *Given two empirical distributions* $\eta = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{x}_m}$ $\mu = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{y}_m}$ *and a vector* $\mathbf{v} \in S^{d-1}$*, we define the* couplings induced by $\mathbf{v}$ *to be the set* $\Sigma_{\mathbf{v}}$ *of bijective maps* $[M] \to [M]$ *that specify an optimal matching for the projected particles* $(\langle \mathbf{v}, \mathbf{x}_m \rangle)_{m=1}^M$, $(\langle \mathbf{v}, \mathbf{y}_m \rangle)_{m=1}^M$*, in the sense that a matching* $\sigma : [M] \to [M]$ *is optimal when the condition* $\langle \mathbf{v}, \mathbf{x}_i \rangle < \langle \mathbf{v}, \mathbf{x}_j \rangle$ *iff* $\langle \mathbf{v}, \mathbf{y}_{\sigma_{\mathbf{v}}(i)} \rangle < \langle \mathbf{v}, \mathbf{y}_{\sigma_{\mathbf{v}}(j)} \rangle$ *is satisfied. Note that typically* $\Sigma_{\mathbf{v}}$ *is a set of size one, but in degenerate cases, there may be more than one induced coupling* $\sigma_{\mathbf{v}}$ *for a given vector* $\mathbf{v}$.

Definition 3.2 (Projected Wasserstein distances). *For* $p \ge 1$*, the p-projected Wasserstein distance between* η *and* μ *is defined as:*

$$\mathrm{PW}_p^p(\eta, \mu) = \mathbb{E}_{\mathbf{v} \sim \mathrm{Unif}(S^{d-1})}\left[\frac{1}{M}\sum_{m=1}^{M} \|\mathbf{x}_m - \mathbf{y}_{\sigma_{\mathbf{v}}(m)}\|_2^p\right], \quad (8)$$

where $\sigma_{\mathbf{v}} : [M] \to [M]$ *is the coupling induced by* $\mathbf{v}$.

Projected Wasserstein distances are thus an alternative to Wasserstein distances that enjoy similar computational efficiency to sliced Wasserstein distances, but correct for the "bias" implicit in the definition of sliced Wasserstein distances. In the remainder of this section, we explore a variety of theoretical and computational aspects of projected Wasserstein distances, and in Section 5 we explore the use of projected Wasserstein distance as an objective in deep generative modelling and reinforcement learning.

3.1 Theoretical properties

Having motivated the projected Wasserstein distance, we now establish some of its basic properties.

Proposition 3.3. *Projected Wasserstein distance* PW_p *is a metric on the space* $\mathscr{P}_{(M)}(\mathbb{R}^d) = \{\frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{x}_m} | \mathbf{x}_m \in \mathbb{R}^d$ *for all* $m \in [M]\} \subset \mathscr{P}(\mathbb{R}^d)$.

Proposition 3.4. *We have the following inequalities*

$$\mathrm{SW}_p(\eta, \mu) \leq \mathrm{W}_p(\eta, \mu) \leq \mathrm{PW}_p(\eta, \mu)\,, \qquad (9)$$

for all $\eta, \mu \in \mathscr{P}_{(M)}(\mathbb{R}^d)$, *for all* $p \geq 1$.

3.2 Monte Carlo estimation of PW distance

Just as sliced Wasserstein distance requires Monte Carlo estimation, so too does projected Wasserstein distance. The estimation algorithm is similar to Algorithm 1 for sliced Wasserstein distance (as our motivation for projected Wasserstein distance might suggest), and is presented as Algorithm 2; the crucial difference is the contribution calculation at Line 9. Within Algorithm 2, `argsort` can be taken to be any subroutine that computes an induced coupling between the projected samples. One implementation that runs in $\mathcal{O}(M \log M)$ time consists of sorting the two lists of real numbers, keeping track of the permutations that sort them, and then computing one with the inverse of the other.

Algorithm 2 thus allows any method utilising Wasserstein or sliced Wasserstein distances, such as Problems (1) and (2) instead to use projected Wasserstein distance.

3.3 Johnson-Lindenstrauss estimation of PW distance

To conclude this section, we discuss several variations on Algorithm (2) which may allow for more efficient estimation of projected Wasserstein distance. These variants are motivated by the difference in computational burden between sliced Wasserstein and projected Wasserstein distances; whilst Algorithm 1 for Monte Carlo estimation of sliced Wasserstein distances deals entirely with one-dimensional projections, Algorithm 2 for estimation of the projected Wasserstein distance requires computation of distances between the original (unprojected) datapoints.

Algorithm 2 Projected Wasserstein estimation

```
Require: $\eta = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{x}_m},\ \mu = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{y}_m}$
 1: Sample $(\mathbf{v}_n)_{n=1}^N \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}(S^{d-1})$
 2: for $n = 1$ to $N$ do
 3:     Compute projected distributions:
 4:         $(\Pi_{\mathbf{v}_n})_\# \eta = \frac{1}{M}\sum_{m=1}^M \delta_{\langle \mathbf{v}_n, \mathbf{x}_m \rangle}$
 5:         $(\Pi_{\mathbf{v}_n})_\# \mu = \frac{1}{M}\sum_{m=1}^M \delta_{\langle \mathbf{v}_n, \mathbf{y}_m \rangle}$
 6:     Compute optimal matching for projected distributions:
 7:         $\sigma_{\mathbf{v}_n} \leftarrow \mathtt{argsort}((\langle \mathbf{v}_n, \mathbf{x}_m \rangle)_{m=1}^M, (\langle \mathbf{v}_n, \mathbf{y}_m \rangle)_{m=1}^M)$
 8:     Compute contribution from coupling:
 9:         $\frac{1}{M}\sum_{m=1}^M \|\mathbf{x}_m - \mathbf{y}_{\sigma_{\mathbf{v}_n}(m)}\|_2^p$
10: end for
11: return $\frac{1}{N}\sum_{n=1}^N \frac{1}{M}\sum_{m=1}^M \|\mathbf{x}_m - \mathbf{y}_{\sigma_{\mathbf{v}_n}(m)}\|_2^p$
```

However, a key observation is that in the course of computing induced couplings in Algorithm 2, many one-dimensional projections of the support of the distributions concerned have been computed. These projections can be pooled, and thus considered collectively as constituting a *random projection* of the support of the distribution, in the vein of the Johnson-Lindenstrauss transform (Johnson and Lindenstrauss, 1984). This random projection can then be used to estimate distances between support points of the distributions, without having to work with the original high-dimensional points themselves. More concretely, this can be achieved by replacing Line 9 of Algorithm 2 with the following computation:

$$\frac{1}{M}\sum_{m=1}^M \frac{d}{N}\sum_{n'=1}^N |\langle \mathbf{v}_{n'}, \mathbf{x}_m - \mathbf{y}_{\sigma_{\mathbf{v}_n}(M)} \rangle|^p\,. \qquad (10)$$

In particular, in the case $p = 2$, this yields an unbiased estimator of the distance computed in Line 9 of Algorithm 2.

4 ORTHOGONAL ESTIMATION FOR SLICED AND PROJECTED WASSERSTEIN DISTANCES

Having introduced the projected Wasserstein distance, we now turn to the second contribution in this paper: developing understanding of improved methods for estimating both sliced Wasserstein and projected Wasserstein distances. Pitié et al. (2007) argue implicitly for using an orthogonal coupling of projection vectors in estimating the sliced Wasserstein distance, and Wu et al. (2017) put forward an approach to generative modelling where a set of orthogonal projection directions are *learnt* from data, to be used in the context of sliced Wasserstein estimation.

Here, we consider the general approach of using orthogonal projection directions within sliced Wasserstein and projected Wasserstein distances. We show the (perhaps surprising) results that: (i) there is a strong connection between orthogonal coupling of projection directions and notions of stratified sampling; (ii) contrary to the intuition of Pitié et al. (2007), using orthogonal projection directions can actually *worsen* the performance of sliced Wasserstein estimation (as measured by estimator variance); but (iii) orthogonal projection directions *always* lead to an improvement in estimator variance for the projected Wasserstein distance in the case $M = 2$; we conjecture that this holds more generally. Besides the motivation presented in Section 3, the projected Wasserstein distances therefore serve an important role in our theoretical understanding of the impact of orthogonally coupled projection directions on estimation of the sliced Wasserstein distances. Details on how to perform practical Monte Carlo sampling from $\text{UnifOrt}(S^{d-1}; N)$, as well as computationally efficient approximate sampling algorithms, are provided in the Appendix.

4.1 Orthogonal couplings

To make precise the notion of orthogonal projection directions, we first make a preliminary definition.

Definition 4.1. *Let $N \leq d$. The probability distribution $\text{UnifOrt}(S^{d-1}; N) \in \mathscr{P}((S^{d-1})^N)$ is defined as the joint distribution of N rows of a random orthogonal matrix drawn from Haar measure on the orthogonal group $\mathscr{O}(d)$. If N is a multiple of d, we define the distribution $\text{UnifOrt}(S^{d-1}; N)$ to be that given by concatenating N/d independent copies of random variables drawn from $\text{UnifOrt}(S^{d-1}; d)$.*

A collection of random vectors $(\mathbf{v}_n)_{n=1}^N$ drawn from $\text{UnifOrt}(S^{d-1}; N)$ has the property that each random vector $\mathbf{v}_n$ is marginally distributed as $\text{Unif}(S^{d-1})$, and all vectors are mutually orthogonal almost surely. The broad idea is to replace the i.i.d. projection directions $(\mathbf{v}_n)_{n=1}^N$ appearing in Algorithms 1 and 2 with a sample from $\text{UnifOrt}(S^{d-1}; N)$; Algorithm 3 specifies this adjustment precisely in the case of sliced Wasserstein estimation, with the new sampling mechanism shown in red. The adjustment for projected Wasserstein estimation is analogous, and is given in the appendix due to space constraints.

4.2 Analysis of orthogonal couplings

We now compare the *mean squared error* (MSE) of i.i.d. and orthogonal estimation of the sliced and projected Wasserstein distance between distributions $(\eta, \mu) \in \mathscr{P}_{(M)}(\mathbb{R}^d) \times \mathscr{P}_{(M)}(\mathbb{R}^d)$. As the MSE of an unbiased estimator is equal to its variance, improving upon i.i.d. requires the cross-covariance induced by sampled directions to be negative. This motivates us to first study a class of *stratified* estimators which is proved to be statistically superior to the i.i.d. approach. The main drawback of the stratification scheme is its $\mathcal{O}((M!)^2)$ computational complexity.

Algorithm 3 Orthogonal sliced Wasserstein estimation

```
Require: $\eta = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{x}_m}$,  $\mu = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{y}_m}$
 1: Sample $(\mathbf{v}_n)_{n=1}^N \sim \text{UnifOrt}(S^{d-1}; N)$
 2: for $n = 1$ to $N$ do
 3:     Compute projected distributions:
 4:         $(\Pi_{\mathbf{v}_n})_\# \eta = \frac{1}{M}\sum_{m=1}^M \delta_{\langle \mathbf{v}_n, \mathbf{x}_m \rangle}$
 5:         $(\Pi_{\mathbf{v}_n})_\# \mu = \frac{1}{M}\sum_{m=1}^M \delta_{\langle \mathbf{v}_n, \mathbf{y}_m \rangle}$
 6:     Compute sorted lists of supports:
 7:         $(w_m)_{m=1}^M \leftarrow \texttt{sort}((\langle \mathbf{v}_n, \mathbf{x}_m \rangle)_{m=1}^M)$
 8:         $(z_m)_{m=1}^M \leftarrow \texttt{sort}((\langle \mathbf{v}_n, \mathbf{y}_m \rangle)_{m=1}^M)$
 9:     Compute one-dimensional Wasserstein distance:
10:         $\text{W}_p^p((\Pi_{\mathbf{v}_n})_\# \eta, (\Pi_{\mathbf{v}_n})_\# \mu) = \frac{1}{M}\sum_{m=1}^M |w_m - z_m|^p$
11: end for
12: return $\frac{1}{N}\sum_{n=1}^N \text{W}_p^p((\Pi_{\mathbf{v}_n})_\# \eta, (\Pi_{\mathbf{v}_n})_\# \mu)$
```

The importance of stratified estimators for our purposes comes from the fact that their improved accuracy is due to the increase in average diversity of induced couplings (cf. Definition 3.1), a property that is also typical for the orthogonally coupled estimators. Orthogonal coupling can therefore be seen as a computationally tractable approximation to stratification. As we observe in experiments, this approximation usually indeed leads to improved MSE. However, we prove that the improvement in the case of sliced Wasserstein estimation is not universal over all pairs of distributions $(\eta, \mu) \in \mathscr{P}_p(\mathbb{R}^d) \times \mathscr{P}_p(\mathbb{R}^d)$, contrary to the intuition of Pitié et al. (2007).

4.2.1 Improving MSE by stratification

We begin by formalising stratification and establishing its dominance over i.i.d. estimation in terms of MSE.

Definition 4.2 (Stratified estimation). *Let $(\mathcal{X}, \mathcal{A})$ be a measurable space, X a random variable with probability distribution $\text{Law}(X)$ taking values in $\mathcal{X}$, and $f \colon \mathcal{X} \to \mathbb{R}$ an integrable function with $\theta := \mathbb{E}[f(X)]$. Assume $\{A_k\}_{k=1}^K \subseteq \mathcal{A}$ is a finite disjoint partition of $\mathcal{X}$. An estimator of θ*

$$\hat{\theta}_N = \frac{1}{N}\sum_{i=1}^N f(X_i) \, ,$$

will be called stratified *if* $\mathrm{Law}(X_i) = \mathrm{Law}(X)$, $\forall i$, *and all bivariate marginals* $\mathrm{Law}((X_i, X_j))$, $i \neq j$, *satisfy:*

(i) X_i *and* X_j *are conditionally independent given the* σ*-algebra generated by* $\{A_k\}_{k=1}^K$*;*

(ii) $\mathbb{P}(X_i \in A_k, X_j \in A_l)$ *is less than (resp. greater than) or equal to* $\mathbb{P}(X \in A_k)\mathbb{P}(X \in A_l)$ *when* $k = l$ *(resp.* $k \neq l$*), for any* $k, l \in [K]$*;*

(iii) X_i *and* X_j *are pairwise exchangeable.*

The inequality in (ii) is required to be strict for at least one $(i,j), i \neq j$*, in the* $k = l$ *case for some* $k \in [K]$*.*

Theorem 4.3. *The MSE of any stratified estimator is lower or equal to that of an i.i.d. estimator. A stratified estimator for which the inequality is strict exists whenever* $\exists\, k, l \in [K]$ *such that* $\mathbb{E}[f(X) \,|\, X \in A_k] \neq \mathbb{E}[f(X) \,|\, X \in A_l]$ *and* $\mathbb{P}(X \in A_k) > 0$, $\mathbb{P}(X \in A_l) > 0$.

Stratification is a well-established means of achieving variance reduction in Monte Carlo (see e.g. Owen (2013)). However, it is a particularly appealing approach in the context of sliced Wasserstein and projected Wasserstein estimation, as there is a natural partition of the space S^{d-1} to consider. Bringing Definition 4.2 into the context of sliced and projected Wasserstein estimation, we take $\mathcal{X} = S^{d-1}$, $X_n = \mathbf{v}_n$, $\mathrm{Law}(X) = \mathrm{Unif}(S^{d-1})$, and f to be the function computed in the inner loop of Algorithms 1 and 2 respectively.

Revisiting Definition 3.1, it is natural to consider partitioning S^{d-1} into sets $\{E_\tau\}_{\tau \in \mathcal{S}_M}$, where $E_\tau := \{\mathbf{v} \in S^{d-1} \,|\, \tau \in \Sigma_\mathbf{v}\}$ with $\Sigma_\mathbf{v}$ denoting the set of optimal matchings for direction v, and $\mathcal{S}_M$ is the set of all permutations of $[M]$. These sets need not be disjoint which we amend using the following observation: multiple couplings are optimal iff either (a) $\mathbf{x}_i = \mathbf{x}_j$ or $\mathbf{y}_i = \mathbf{y}_j$, for some $i \neq j$; or (b) $\langle \mathbf{v}, \mathbf{x}_i \rangle = \langle \mathbf{v}, \mathbf{x}_j \rangle$ or $\langle \mathbf{v}, \mathbf{y}_i \rangle = \langle \mathbf{v}, \mathbf{y}_j \rangle$, for some $i \neq j$. In (a), we are free to deterministically pick any of the optimal couplings as the contribution to both the sliced and projected Wasserstein integrals will be the same under any of them. The events in (b) are then null sets and we can thus again safely pick any of the available couplings.

Stratification with the modified $\{E_\tau\}_{\tau \in \mathcal{S}_M}$ partition can therefore be applied to estimation of projected and sliced Wasserstein distances and by Theorem 4.3 will lead to improved MSE in all but degenerate cases.

4.2.2 Orthogonal coupling of directions as approximate stratification

The last section presented a sampling scheme which renders i.i.d. sampling *inadmissible* in terms of MSE. However, the proposed stratification approach crucially relies on knowledge of the E_σ regions and our ability to sample uniformly from these.

Remark 4.4. *Each region* E_σ*, can be written as a finite union of simply-connected sets.*

Specifically, a coupling $\sigma \in \mathcal{S}_M$ *is optimal for a given* $\mathbf{v}$ *iff it corresponds to the coupling implied by associating the projected points according to their order. The region where a particular fixed ordering* $\tau \in \mathcal{S}_M$ *of* $\{\langle \mathbf{v}, \mathbf{x}_i \rangle\}_{i=1}^M$ *is achieved can be obtained as follows:*

(i) for $i = 1, \dots, M-1$*, define the half-spaces*

$$H^{\mathbf{x}}_{\tau(i),\tau(i+1)} := \{\mathbf{v} \in S^{d-1} |\, \langle \mathbf{v}, \mathbf{x}_{\tau(i)} - \mathbf{x}_{\tau(i+1)} \rangle \leq 0\}\,;$$

(ii) obtain the region $B^{\mathbf{x}}_\tau = \bigcap_{i=1}^{M-1} H^{\mathbf{x}}_{\tau(i),\tau(i+1)}$.

Defining $B^{\mathbf{y}}_\tau$ *analogously and using the definition* $E_\sigma = \{\mathbf{v} \in S^{d-1} \,|\, \sigma \in \Sigma_\mathbf{v}\}$*, we can write* E_σ *as a finite union of intersections of half-spaces:*

$$E_\sigma = \bigcup_{\tau \in \mathcal{S}_M} \left(B^{\mathbf{x}}_\tau \cap B^{\mathbf{y}}_{\tau \circ \sigma}\right),$$

where $\tau \circ \sigma$ *denotes composition of the two mappings.*

By Remark 4.4, the structure of E_σ quickly grows in complexity as M increases. Practical implementation of the algorithm is thus computationally intractable for all but very small problems.

However, we might view the orthogonal coupling of the $\{\mathbf{v}_n\}_{n=1}^N$ directions as an approximation to stratification, since: (a) the directions are pairwise exchangeable and marginally $\mathrm{Unif}(S^{d-1})$, and (b) the orthogonal coupling of the directions should intuitively decrease the chance of sampling the same induced coupling because the individual E_σ are finite unions of simply-connected sets (cf. Remark 4.4). However, even if we assumed that Condition (ii) from Definition 4.2 holds, Condition (i) will only be satisfied in the case of the projected Wasserstein distance for which $f(\mathbf{v})$ is piecewise constant (cf. Definition 4.2).

The following result for the simplified case $M = 2, d = 2$ supports the above intuition, showing that orthogonal coupling improves MSE in the projected but not necessarily in the sliced Wasserstein case.

Proposition 4.5. *Let* $M = 2$ *and* $d = 2$*. Then orthogonally coupled estimator of projected Wasserstein distance satisfies Definition 4.2. For the sliced Wasserstein distance, neither i.i.d. nor orthogonal estimation dominates the other in terms of MSE.*

5 EXPERIMENTS

We now present empirical evaluation of the theory introduced in the previous sections.

5.1 Distance estimation

We begin with a test bed of small-scale problems, which will aid intuition as to the advantages of orthog-

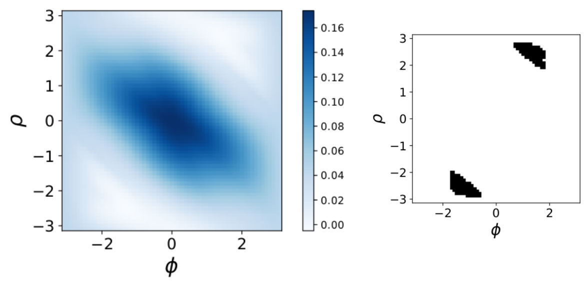

Figure 1: Difference between MSE of the orthogonally coupled and i.i.d. estimator of 1-sliced Wasserstein distance with $N = 2$, for various datasets in the $M = 2, d = 2$ scenario. For all datasets, $\mathbf{x}_1 = [0,0]^\top$, $\mathbf{x}_2 = [1,0]^\top$ is fixed, and $\mathbf{z} = \mathbf{x}_1 - \mathbf{y}_1$ and $\mathbf{r} = \mathbf{y}_1 - \mathbf{y}_2$ have unit norm. The only difference between the datasets are the angles between the x-axis and the vectors $\mathbf{z}$ and $\mathbf{r}$, which we respectively denote by ρ and ϕ. The numerical difference between MSEs is on the left (the higher the better is orthogonal). The two connected black regions on the right highlight the configurations of ρ and ϕ in which i.i.d. is better than orthogonal.

onal estimation. The problems we consider consist of computing the sliced Wasserstein distance

$$\mathrm{SW}_p\left(\frac{1}{M}\sum_{m=1}^{M}\delta_{\mathbf{x}_i}, \frac{1}{M}\sum_{m=1}^{M}\delta_{\mathbf{y}_i}\right). \tag{11}$$

To generate a collection of problems, we sample $(\mathbf{x}_m)_{m=1}^M$, $(\mathbf{y}_m)_{m=1}^M$ independently from distributions $N(\mu_\mathbf{x}, I)$, $N(\mu_\mathbf{y}, I)$. In Figure 2, we plot comparisons of the MSE achieved by estimators using i.i.d. and orthogonally coupled projection directions for a variety of values of the parameters d and N, in the case $p = 1$, $\mu_\mathbf{x} = \mu_\mathbf{y}$, and use a number of projection direction N equal to the dimensionality d. Each pair of sampled distributions in Expression (11) corresponds to a scatter point in the relevant graph in Figure 2. The results for the case $d = 2$ illustrate our theoretical results that orthogonality does not always guarantee improved MSE, although in the vast majority of cases, there is indeed an improvement. The case of $d = 50$ illustrates that as dimensionality increases, the improved performance of orthogonally-coupled projection directions becomes more robust.

5.2 Generative modelling

We study the effect of orthogonal estimation in the context of generative modeling. In particular, we study the sliced Wasserstein Auto-Encoders Kolouri et al. (2018) on MNIST dataset. The auto-encoder

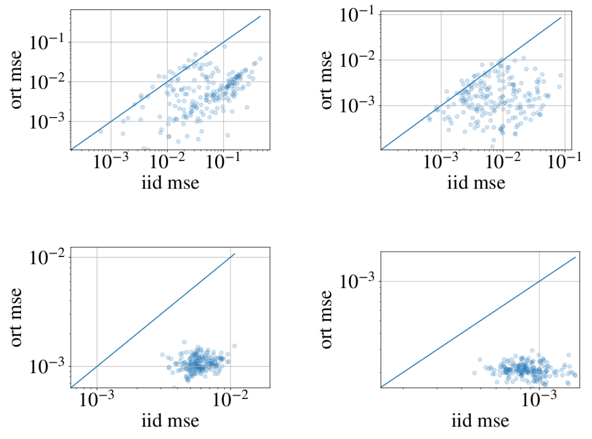

Figure 2: Scatter plots of orthogonal estimator MSE vs. i.i.d. estimator MSE for $d = 2$ (left column) and $d = 50$ (right column), and $M = 2$ (top row) and $M = 10$ (bottom row).

consists of an encoder $f_\theta(\cdot)$ with parameter θ and a decoder $g_\phi(\cdot)$ with parameter ϕ. For a given observation $\mathbf{x} \in \mathbb{R}^n$, the encoder computes an hidden code $\mathbf{z} = f_\theta(\mathbf{x}) \in \mathbb{R}^h$. With a hidden code $\mathbf{z}$, the decoder computes a generated sample $\tilde{\mathbf{x}} = g_\phi(\mathbf{z})$. Given a distribution of m observations $\{\mathbf{x}_i\}_{i=1}^m$, let $P(X) = \frac{1}{m}\sum_{i=1}^m \delta_{\mathbf{x}_i}$ be the empirical distribution, we hope to jointly train an encoder $f_\theta(\cdot)$ that uncovers the hidden structure of $P(X)$ and a decoder $g_\phi(\cdot)$ that generates samples similar to $P(X)$. Let $p_\theta(\mathbf{z})$ be the push-forward distribution from $\mathbf{z} = f_\theta(\mathbf{x}), \mathbf{x} \sim P(X)$ and $p(\mathbf{z})$ be a prior distribution over hidden codes. The loss of the auto-encoder is defined as

$$L_{\mathrm{ae}}(\theta,\phi) = \mathbb{E}_{\mathbf{x}\sim P(X)}[\|g_\phi(f_\theta(\mathbf{x})) - \mathbf{x}\|] + \mathrm{SW}_1(p_\theta(\mathbf{z}), p(\mathbf{z}))$$

where the first term is a reconstruction error and the second term is to enforce that the generated hidden codes $p_\theta(\mathbf{z})$ be close to the prior distribution $p(\mathbf{z})$. We then update θ, ϕ by approximating gradient descent $(\theta,\phi) \leftarrow (\theta,\phi) - \alpha\nabla_{(\theta,\phi)}L_{\mathrm{ae}}(\theta,\phi)$ with learning rate α. To estimate $\mathrm{SW}_1(p_\theta(\mathbf{z}), p(\mathbf{z}))$, we compare orthogonal vs. i.i.d. Monte Carlo samples and we expect that the benefits from a more accurate estimate of the sliced Wasserstein distance translates into higher quality gradient updates. Experimental details are in the Appendix.

In Figure 3, we present the learning curves of auto-encoders using three methods: i.i.d. Monte Carlo estimate, orthogonal estimate and an approximation to orthogonal estimation using Hadamard-Rademacher (HD) random matrices (see Appendix) to approximate orthogonal estimate. We observe that the effect of orthogonality depends on the hyperparameters in the learning procedure: when the learning rate is large

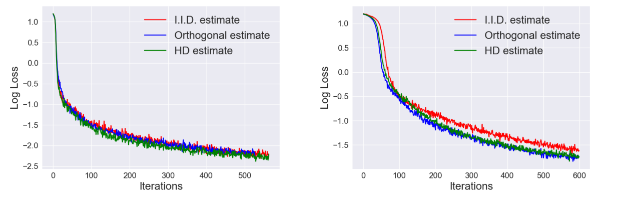

Figure 3: Training curves of Sliced Wasserstein Auto-encoders with three methods to compute Sliced Wasserstein distance: i.i.d. Monte Carlo estimate (red), orthogonal estimate (blue) and HD matrix for orthogonal estimate (green). Vertical axis is the log training loss, horizontal axis is the number of iterations. Left uses a learning rate of $\alpha = 1.0 \cdot 10^{-4}$ and right uses a learning rate of $\alpha = 1.0 \cdot 10^{-5}$.

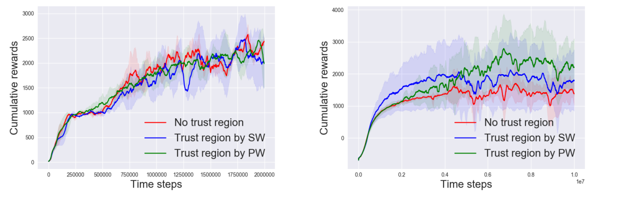

Figure 4: Training curves of RL with three methods to compute policy gradients updates on benchmark tasks (left: Hopper, right: HalfCheetah): no trust region (red), trust region by sliced Wasserstein (blue) and trust region by projected Wasserstein (green). Training curves show the mean ± std performance across 5 random seeds. Vertical axis is the cumulative reward, horizontal axis is the number of time steps.

$\alpha = 10^{-4}$ (Figure 3, left), both estimates behave similarly; when the learning rate is small $\alpha = 10^{-5}$ (Figure 3, right), orthogonal estimate leads to a slightly faster convergence than i.i.d. estimate. When using HD matrices as a proxy to compute orthogonal estimates, we always benefit from the computational benefit at training time. Additional results including a comparison of sliced Wasserstein and projected Wasserstein distance as objective for generative modelling can be found in the Appendix.

5.3 Reinforcement learning

In reinforcement learning (RL), at time t an agent is in state $\mathbf{s}_t$, takes an action $\mathbf{a}_t$, receives an instant reward r_t and transitions to next state $\mathbf{s}_{t+1}$. The objective is to search for a policy $\pi_\theta : \mathbf{s}_t \mapsto \mathbf{a}_t$ parameterized by θ such that the expected discounted cumulative reward $J(\pi_\theta) = \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{\infty} \gamma^t r_t]$ for some discount factor $\gamma \in (0, 1)$ is maximized. Policy gradient algorithms apply (an approximation of) the gradient update $\theta_{\text{new}} \leftarrow \theta_{\text{old}} + \alpha \nabla_{\theta_{\text{old}}} J(\pi_{\theta_{\text{old}}})$ to iteratively improve the policy. Trust region policy optimization (Schulman et al., 2015) requires that $D(\pi_{\theta_{\text{old}}} || \pi_{\theta_{\text{new}}}) \le \epsilon$ for some $\epsilon > 0$ to ensure that the updates are stable, where $D(\cdot, \cdot)$ is some discrepancy measure between two policies. Previously, Schulman et al. (2015) propose to set $D(\cdot, \cdot) = \mathbb{KL}[\cdot||\cdot]$ as the KL divergence, while Zhang et al. (2018) set $D(\cdot, \cdot) = \mathrm{W}_1(\cdot, \cdot)$ as the 1-Wasserstein distance. As alternates to the discrepancy measure, we take $D(\cdot, \cdot)$ to be the sliced Wasserstein distance $\mathrm{SW}_1(\cdot, \cdot)$ or projected Wasserstein distance $\mathrm{PW}_1(\cdot, \cdot)$. For fast optimization, instead of constructing an explicit constraint, we adopt a penalty formulation of the trust region (Schulman et al., 2017; Zhang et al., 2018) and update $\theta_{\text{new}} \leftarrow \theta_{\text{old}} + \alpha \nabla_{\theta_{\text{old}}} (J(\pi_{\theta_{\text{old}}}) - \lambda D(\theta_{\text{old}}, \theta_{\text{new}}))$ for some penalty constant $\lambda > 0$. We present all algorithmic and implementation details in the Appendix.

Since projected Wasserstein corrects for the implicit "bias" introduced by sliced Wasserstein, we expect the trust region by projected Wasserstein lead to more stable training. In Figure 4, we show the training curves on benchmark tasks HalfCheetah (right) and Hopper (left) (Brockman et al., 2016). We compare the training curves of three schemes: no trust region (red), trust region by sliced Wasserstein distance (blue) and trust region by projected Wasserstein distance (green). In most tasks with simple dynamics as Hopper, we do not see significant difference between three methods; however, in tasks with more complex dynamics such as HalfCheetah, we observe that trust region updates with projected Wasserstein distance leads to slightly more stable updates than the other two baselines, achieving higher cumulative rewards within a fixed number of training steps.

6 CONCLUSION

We have considered projected Wasserstein distance, a variant of sliced Wasserstein distance, and studied orthogonal couplings of projection directions in estimators of sliced and projected Wasserstein distances. In doing so, we have also given an interpretation of orthogonal coupling as an efficient, approximate means of performing stratified sampling. Our empirical evaluations show that orthogonality can dramatically reduce estimator variance, and these benefits are translated over to downstream tasks such as generative modelling in certain circumstances. Important areas for future work include deepening our understanding of the relationship between improvements in estimation of Wasserstein distances themselves and improvements in downstream tasks such as distribution learning, and strengthening our understanding of the effectiveness of orthogonal couplings in Monte Carlo estimators of Wasserstein distances.

Acknowledgements

MR acknowledges support by EPSRC grant EP/L016516/1 for the Cambridge Centre for Analysis. JH acknowledges support by a Nokia CASE Studentship. YHT acknowledges the cloud credits provided by Amazon Web Services. AW acknowledges support from the David MacKay Newton research fellowship at Darwin College, The Alan Turing Institute under EPSRC grant EP/N510129/1 & TU/B/000074, and the Leverhulme Trust via the CFI.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). TensorFlow: A System for Large-Scale Machine Learning. In *Operating Systems Design and Implementation (OSDI)*.

Ambrosio, L., Gigli, N., and Savare, G. (2005). *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*.

Andoni, A., Indyk, P., Laarhoven, T., Razenshteyn, I., and Schmidt, L. (2015). Practical and Optimal LSH for Angular Distance. In *Neural Information Processing Systems (NeurIPS)*.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning, (ICML)*.

Bonneel, N., Rabin, J., Peyré, G., and Pfister, H. (2015). Sliced and Radon Wasserstein Barycenters of Measures. *Journal of Mathematical Imaging and Vision*, 1(51):22–45.

Bonnotte, N. (2013). *Unidimensional and Evolution Methods for Optimal Transportation*. PhD thesis.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). OpenAI Gym. *arXiv*.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Choromanski, K. M., Rowland, M., and Weller, A. (2017). The Unreasonable Effectiveness of Structured Random Orthogonal Embeddings. In *Neural Information Processing Systems (NeurIPS)*.

Cuturi, M. (2013). Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Neural Information Processing Systems (NeurIPS)*.

Cuturi, M. and Doucet, A. (2014). Fast Computation of Wasserstein Barycenters. In *International Conference on Machine Learning (ICML)*.

Deshpande, I., Zhang, Z., and Schwing, A. G. (2018). Generative Modeling using the Sliced Wasserstein Distance. *arXiv*.

Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., and Zhokhov, P. (2017). OpenAI Baselines. `https://github.com/openai/baselines`.

Galichon, A. (2016). *Optimal Transport Methods in Economics*. Princeton University Press.

Genevay, A., Cuturi, M., Peyré, G., and Bach, F. (2016). Stochastic Optimization for Large-scale Optimal Transport. In *Neural Information Processing Systems (NeurIPS)*.

Genz, A. (1999). Methods for Generating Random Orthogonal Matrices. *Monte Carlo and quasi-Monte Carlo methods (MCQMC)*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Nets. In *Neural Information Processing Systems (NeurIPS)*.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved Training of Wasserstein GANs. In *Neural Information Processing Systems (NeurIPS)*.

Harrison, J. R. and March, J. G. (1984). Decision Making and Postdecision Surprises. *Administrative Science Quarterly*, 29(1):26–42.

Hasselt, H. V. (2010). Double Q-learning. In *Neural Information Processing Systems (NeurIPS)*.

Johnson, W. and Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability*, volume 26, pages 189–206.

Jordan, R., Kinderlehrer, D., and Otto, F. (1998). The Variational Formulation of the Fokker-Planck Equation. *SIAM J. Math. Anal.*, 29(1):1–17.

Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*.

Kolouri, S., Martin, C. E., and Rohde, G. K. (2018). Sliced-Wasserstein Autoencoder: An Embarrassingly Simple Generative Model. *arXiv*.

Owen, A. B. (2013). *Monte Carlo Theory, Methods and Examples*.

Peyré, G. and Cuturi, M. (2018). Computational Optimal Transport. *arXiv*.

Pitié, F., Kokaram, A. C., and Dahyot, R. (2007). Automated Colour Grading Using Colour Distribution Transfer. *Comput. Vis. Image Underst.*, 107(1-2):123–137.

Rabin, J., Peyr, G., Delon, J., and Bernot, M. (2011). Wasserstein Barycenter and Its Application to Texture Mixing. In *SSVM*, volume 6667 of *Lecture Notes in Computer Science*, pages 435–446. Springer.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning (ICML)*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal Policy Optimization Algorithms. *arXiv*.

Smith, J. E. and Winkler, R. L. (2006). The Optimizers Curse: Skepticism and Postdecision Surprise in Decision Analysis. *Manage. Sci.*, 52(3):311–322.

Staib, M., Claici, S., Solomon, J. M., and Jegelka, S. (2017). Parallel Streaming Wasserstein Barycenters. In *Neural Information Processing Systems (NeurIPS)*.

Villani, C. (2008). *Optimal Transport: Old and New*. Springer.

Wu, J., Huang, Z., Li, W., Thoma, J., and Van Gool, L. (2017). Sliced Wasserstein Generative Models. *arXiv*.

Zhang, R., Chen, C., Li, C., and Carin, L. (2018). Policy Optimization as Wasserstein Gradient Flows. *arXiv*.

APPENDIX: Orthogonal Estimation of Wasserstein Distances

7 Proofs of results in Section 3

Proposition 3.3. *Projected Wasserstein distance* PW_p *is a metric on the space* $\mathscr{P}_{(M)}(\mathbb{R}^d) = \{\frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{x}_m} | \mathbf{x}_m \in \mathbb{R}^d$ *for all* $m \in [M]\} \subset \mathscr{P}(\mathbb{R}^d)$.

Proof. Symmetry and non-negativity are immediate. We thus turn our attention to proving: (i) $\mathrm{PW}_p(\eta,\mu)=0$ iff $\eta=\mu$; and (ii) the triangle inequality.

For (i), first let $\eta = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{x}_m}$ and $\mu = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{y}_m}$ be distinct. Then for *any* bijective map $\sigma : [M] \rightarrow [M]$, we have $\sum_{m=1}^M \|\mathbf{x}_m - \mathbf{y}_{\sigma(m)}\|_2^p > 0$, and hence immediately we have $\mathrm{PW}(\eta,\mu) > 0$. The converse direction is clear.

For (ii), let $\eta = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{x}_m}$, $\mu = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{y}_m}$, and $\zeta = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{z}_m}$. Fix $\mathbf{v} \in S^{d-1}$, and without loss of generality, assume that the points $(\mathbf{x}_m)_{m=1}^M$, $(\mathbf{y}_m)_{m=1}^M$, $(\mathbf{z}_m)_{m=1}^M$ are indexed so that

$$\langle \mathbf{v}, \mathbf{x}_1\rangle \leq \langle \mathbf{v}, \mathbf{x}_2\rangle \leq \cdots \leq \langle \mathbf{v}, \mathbf{x}_M\rangle\,, \quad \langle \mathbf{v}, \mathbf{y}_1\rangle \leq \langle \mathbf{v}, \mathbf{y}_2\rangle \leq \cdots \leq \langle \mathbf{v}, \mathbf{y}_M\rangle\,, \quad \langle \mathbf{v}, \mathbf{z}_1\rangle \leq \langle \mathbf{v}, \mathbf{z}_2\rangle \leq \cdots \leq \langle \mathbf{v}, \mathbf{z}_M\rangle\,. \tag{12}$$

Now observe that with this indexing notation, the value of the integrand in the definition of projected Wasserstein distances $\mathrm{PW}_p(\eta,\mu)$, $\mathrm{PW}_p(\eta,\zeta)$, and $\mathrm{PW}_p(\mu,\zeta)$ (Equation (8)) for this particular projection vector $\mathbf{v}$ are

$$\frac{1}{M}\sum_{m=1}^M \|\mathbf{x}_m - \mathbf{y}_m\|_2^p\,, \quad \frac{1}{M}\sum_{m=1}^M \|\mathbf{x}_m - \mathbf{z}_m\|_2^p\,, \quad \frac{1}{M}\sum_{m=1}^M \|\mathbf{y}_m - \mathbf{z}_m\|_2^p\,, \tag{13}$$

respectively. Thus, the full projected Wasserstein distances may be expressed as follows:

$$\mathrm{PW}_p(\eta,\mu) = \left[\sum_{(\sigma,\tau,\pi)\in\mathcal{S}_M^3} q(\sigma,\tau,\pi)\sum_{m=1}^M \|\mathbf{x}_{\sigma(m)} - \mathbf{y}_{\tau(m)}\|_2^p\right]^{1/p}, \tag{14}$$

$$\mathrm{PW}_p(\eta,\zeta) = \left[\sum_{(\sigma,\tau,\pi)\in\mathcal{S}_M^3} q(\sigma,\tau,\pi)\sum_{m=1}^M \|\mathbf{x}_{\sigma(m)} - \mathbf{z}_{\pi(m)}\|_2^p\right]^{1/p}, \tag{15}$$

$$\mathrm{PW}_p(\mu,\zeta) = \left[\sum_{(\sigma,\tau,\pi)\in\mathcal{S}_M^3} q(\sigma,\tau,\pi)\sum_{m=1}^M \|\mathbf{y}_{\tau(m)} - \mathbf{z}_{\pi(m)}\|_2^p\right]^{1/p}, \tag{16}$$

where $\sigma,\tau,\pi \in \mathcal{S}_M$ are the permutations needed to re-index $(\mathbf{x}_m)_{m=1}^M$, $(\mathbf{y}_m)_{m=1}^M$, and $(\mathbf{z}_m)_{m=1}^M$, respectively, so that Equation (12) holds, and $q(\sigma,\tau,\pi)$ is the probability that permutations σ,τ,π are required, given that $\mathbf{v}$ is drawn from $\mathrm{Unif}(S^{d-1})$. With these alternative expressions established, the triangle inequality for PW_p now follows from the standard Minkowski inequality. □

Proposition 3.4. *We have the following inequalities*

$$\mathrm{SW}_p(\eta,\mu) \leq \mathrm{W}_p(\eta,\mu) \leq \mathrm{PW}_p(\eta,\mu)\,, \tag{9}$$

for all $\eta,\mu \in \mathscr{P}_{(M)}(\mathbb{R}^d)$, *for all* $p \geq 1$.

Proof. The inequality between sliced Wasserstein and Wasserstein distances is well-known, and a short proof is given by e.g. Bonnotte (2013). For the inequality between Wasserstein and projected Wasserstein distances, write $\eta = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{x}_m}$ and $\mu = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{y}_m}$. Now note that

$$\begin{aligned}
\mathrm{PW}_p^p(\eta,\mu) &= \mathbb{E}_{\mathbf{v}\sim\mathrm{Unif}(S^{d-1})}\left[\frac{1}{M}\sum_{m=1}^M \|\mathbf{x}_m - \mathbf{y}_{\sigma_{\mathbf{v}}(m)}\|_2^p\right] && (17)\\
&\geq \mathbb{E}_{\mathbf{v}\sim\mathrm{Unif}(S^{d-1})}\left[\min_{\sigma\in\mathcal{S}_M}\frac{1}{M}\sum_{m=1}^M \|\mathbf{x}_m - \mathbf{y}_{\sigma(m)}\|_2^p\right] && (18)\\
&= \min_{\sigma\in\mathcal{S}_M}\frac{1}{M}\sum_{m=1}^M \|\mathbf{x}_m - \mathbf{y}_{\sigma(m)}\|_2^p && (19)\\
&= \mathrm{W}_p^p(\eta,\mu)\,, && (20)
\end{aligned}$$

where $\mathcal{S}_M$ is the symmetric group, i.e. the space of bijective mappings from $[M]$ to itself. □

8 Additional material relating to Section 4

8.1 Orthogonal projected Wasserstein estimation

We present the full algorithm applying orthogonal projection directions to estimation of the projected Wasserstein distance in Algorithm 4

Algorithm 4 Projected Wasserstein estimation

```
Require: $\eta = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{x}_m},\ \ \mu = \frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{y}_m}$
 1: Sample $(\mathbf{v}_n)_{n=1}^N \sim \mathrm{UnifOrt}(S^{d-1}; N)$
 2: for $n = 1$ to $N$ do
 3:     Compute projected distributions:
 4:         $(\Pi_{\mathbf{v}_n})_{\#}\eta = \frac{1}{M}\sum_{m=1}^M \delta_{\langle \mathbf{v}_n, \mathbf{x}_m\rangle}$
 5:         $(\Pi_{\mathbf{v}_n})_{\#}\mu = \frac{1}{M}\sum_{m=1}^M \delta_{\langle \mathbf{v}_n, \mathbf{y}_m\rangle}$
 6:     Compute optimal matching for projected distributions:
 7:         $\sigma_{\mathbf{v}_n} \leftarrow \mathtt{argsort}((\langle \mathbf{v}_n, \mathbf{x}_m\rangle)_{m=1}^M, (\langle \mathbf{v}_n, \mathbf{y}_m\rangle)_{m=1}^M)$
 8:     Compute contribution from coupling:
 9:         $\frac{1}{M}\sum_{m=1}^M \|\mathbf{x}_m - \mathbf{y}_{\sigma_{\mathbf{v}_n}(m)}\|^p$
10: end for
11: return $\widehat{\mathrm{PW}}_p^p(\eta,\mu) = \frac{1}{N}\sum_{n=1}^N \frac{1}{M}\sum_{m=1}^M \|\mathbf{x}_m - \mathbf{y}_{\sigma_{\mathbf{v}_n}(m)}\|^p$
```

8.2 Sampling from $\mathrm{UnifOrt}(S^{d-1}; N)$

As described in Definition 4.1, the primary task in sampling from $\mathrm{UnifOrt}(S^{d-1}; N)$ is sampling an orthogonal matrix from Haar measure on the orthogonal group $\mathscr{O}(d)$. This is a well-studied problem (see e.g. Genz (1999)), and we briefly review a method for exact simulation. Algorithm 5 generates such matrices, and can be understood as follows. Initially, the rows of $\mathbf{A}$ are independent with uniformly random directions. Normalising and performing Gram-Schmidt orthogonalisation results in an ordered set of unit vectors that are uniformly distributed on the Steifel manifold, and hence the matrix obtained by taking these vectors as rows is distributed according to Haar measure on the orthogonal group $\mathscr{O}(d)$.

Algorithm 5 Gram-Schmidt orthogonal matrix generation

```
1: Sample $(\mathbf{A}_{ij})_{i,j=1}^d \overset{\text{i.i.d.}}{\sim} N(0,1)$
2: Normalise the norms of the rows of $\mathbf{A}$ to 1.
3: Perform Gram-Schmidt orthogonalisation on the rows of $\mathbf{A}$ to obtain $\widetilde{\mathbf{A}}$
4: return $\widetilde{\mathbf{A}}$
```

8.3 Approximate sampling from $\mathrm{UnifOrt}(S^{d-1}; N)$

The Gram-Schmidt subroutine described in Section 8.2 has computational cost $\mathcal{O}(d^3)$. Whilst in general, this cost would be dominated by the cost of computing a full Wasserstein distance between point clouds (costing $\mathcal{O}(M^{5/2}\log M)$ in the special case of matching, and at least $\mathcal{O}(M^4)$ more generally), it is desirable to reduce the cost of sampling from $\mathrm{UnifOrt}(S^{d-1}; N)$ further, to make projected/sliced Wasserstein estimation more computationally efficient. A variety of methods for *approximately* sampling from $\mathrm{UnifOrt}(S^{d-1}; N)$ at a cost of $\mathcal{O}(d^2 \log d)$ exist (see for example (Genz, 1999; Choromanski et al., 2017; Andoni et al., 2015)), reducing the cost to approximately that of sampling independent projection directions (i.e. $\mathcal{O}(d^2)$). In our experiments, we use Hadamard-Rademacher random matrices to this end; further details are given in Section 9.5.

8.4 Proofs

Theorem 4.3. *The MSE of any stratified estimator is lower or equal to that of an i.i.d. estimator. A stratified estimator for which the inequality is strict exists whenever* $\exists\, k, l \in [K]$ *such that* $\mathbb{E}[f(X) \,|\, X \in A_k] \neq \mathbb{E}[f(X) \,|\, X \in A_l]$ *and* $\mathbb{P}(X \in A_k) > 0$, $\mathbb{P}(X \in A_l) > 0$.

Proof. Recalling the notation of Definition 4.2, the MSE of any unbiased estimator is equal to its variance

$$\mathbb{V}(\hat{\theta}_N) = \frac{\mathbb{V}(f(X))}{N} + \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j\neq i}\mathbb{E}[f(X_i)f(X_j)] - \{\mathbb{E}[f(X)]\}^2 \,.$$

The latter term on the r.h.s. of the above equation is equal to zero for the i.i.d. estimator and thus MSE can only be improved if it is negative. By Definition 4.2, $\mathbb{E}[f(X_i)f(X_j) \,|\, X_i \in A_k, X_j \in A_l] = \mathbb{E}[f(X) \,|\, X \in A_k]\mathbb{E}[f(X) \,|\, X \in A_l]$ whenever $i \neq j$. We can thus rewrite the cross covariance as

$$\mathbb{E}[f(X_i)f(X_j)] - \{\mathbb{E}[f(X)]\}^2 = \sum_{k=1}^{K}\sum_{l=1}^{K}(p_{k,l}^{(i,j)} - p_k p_l) s_k s_l \,,$$

where $p_{k,l}^{(i,j)} := \mathbb{P}(X_i \in A_k, X_j \in A_l)$, $p_k = \mathbb{P}(X \in A_k)$, and $s_k := \mathbb{E}[f(X) \,|\, X \in A_k]$. Defining the matrix $[\mathbf{P}^{(i,j)}]_{k,l} := p_{k,l}^{(i,j)}$ and the vector $[\mathbf{p}]_k := p_k$ we have that the cross-covariance is non-positive for all integrable f iff $\mathbf{P}^{(i,j)} - \mathbf{p}\mathbf{p}^\top$ is negative semi-definite. Observing that the constraints on bivariate marginals in Definition 4.2 ensure that each $\mathbf{P}^{(i,j)}$ is a diagonally dominant Hermitian matrix with non-positive entries on the main diagonal, implying negative semi-definiteness.

To prove existence of a stratified estimator which strictly improves upon i.i.d., consider the matrix $\mathbf{P}^{(1,2)}$ and let $k, l \in [K]$ be the indices for which $\mathbb{E}[f(X) \,|\, X \in A_k] \neq \mathbb{E}[f(X) \,|\, X \in A_l]$ and $\mathbb{P}(X \in A_k) > 0$, $\mathbb{P}(X \in A_l) > 0$. Equate $\mathbf{P}^{(1,2)} = \mathbf{p}\mathbf{p}^\top$ except for setting $\mathbf{P}_{k,k}^{(1,2)} = p_k p_k - \varepsilon$, $\mathbf{P}_{l,l}^{(1,2)} = p_l p_l - \varepsilon$, and $\mathbf{P}_{k,l}^{(1,2)} = \mathbf{P}_{k,l}^{(1,2)} = p_k p_l + \varepsilon$, for some $\varepsilon > 0$ which preserves non-negativity of the entries of $\mathbf{P}^{(1,2)}$. If X_1, X_2 are sampled independently given $\{A_k\}_{k=1}^K$, and $X_3, \ldots, X_N$ i.i.d. $\mathrm{Unif}(S^{d-1})$ (if $N > 2$), then $\mathbb{E}[f(X_i)f(X_j)] - \{\mathbb{E}[f(X)]\}^2 < 0$. □

Proposition 4.5. *Let $M = 2$ and $d = 2$. Then orthogonally coupled estimator of projected Wasserstein distance satisfies Definition 4.2. For the sliced Wasserstein distance, neither i.i.d. nor orthogonal estimation dominates the other in terms of MSE.*

Proof. We begin by observing that for $d = 2$, $\mathbf{v} \in S^{d-1}$ can be parametrised by single parameter $\phi \in [0, 2\pi)$ as $\mathbf{v} = [\cos(\phi), \sin(\phi)]^\top \in \mathbb{R}^2$. Denoting $\{\sigma_A, \sigma_B\} = \mathcal{S}_2$ with $\sigma_A(i) = i, i = 1, 2$ and $\sigma_B(1) = 2, \sigma_B(2) = 1$, we can characterise the sets $\widetilde{E}_A = \{\phi \in \mathbb{R} \,|\, \sigma_A \in \Sigma_{\mathbf{v}}\}$ and $\widetilde{E}_B = \{\phi \in \mathbb{R} \,|\, \sigma_B \in \Sigma_{\mathbf{v}}\}$ as follows: a matching is optimal iff it agrees with the ordering of $\langle \mathbf{v}, \mathbf{x}_1 \rangle, \langle \mathbf{v}, \mathbf{x}_2 \rangle$ and $\langle \mathbf{v}, \mathbf{y}_1 \rangle, \langle \mathbf{v}, \mathbf{y}_2 \rangle$; therefore we can define

$$\begin{aligned} H_{\mathbf{x}}^{+} &= \{\phi \in \mathbb{R} \,|\, \langle \mathbf{v}, \mathbf{x}_1 - \mathbf{x}_2 \rangle \geq 0\}\,, \quad H_{\mathbf{x}}^{-} = \{\phi \in \mathbb{R} \,|\, \langle \mathbf{v}, \mathbf{x}_1 - \mathbf{x}_2 \rangle \leq 0\}\,, \\ H_{\mathbf{y}}^{+} &= \{\phi \in \mathbb{R} \,|\, \langle \mathbf{v}, \mathbf{y}_1 - \mathbf{y}_2 \rangle \geq 0\}\,, \quad H_{\mathbf{y}}^{-} = \{\phi \in \mathbb{R} \,|\, \langle \mathbf{v}, \mathbf{y}_1 - \mathbf{y}_2 \rangle \leq 0\}\,, \end{aligned} \tag{21}$$

and observe $\widetilde{E}_A := (H_{\mathbf{x}}^{+} \cap H_{\mathbf{y}}^{+}) \cup (H_{\mathbf{x}}^{-} \cap H_{\mathbf{y}}^{-})$ and $\widetilde{E}_B := (H_{\mathbf{x}}^{+} \cap H_{\mathbf{y}}^{-}) \cup (H_{\mathbf{x}}^{-} \cap H_{\mathbf{y}}^{+})$ As argued in the main text, $\{\phi \in \mathbb{R} \,|\, \langle \mathbf{v}, \mathbf{x}_1 - \mathbf{x}_2 \rangle = 0\} \cap [0, 2\pi)$ and $\{\phi \in \mathbb{R} \,|\, \langle \mathbf{v}, \mathbf{y}_1 - \mathbf{y}_2 \rangle = 0\} \cap [0, 2\pi)$ are null events except for the degenerate case when $\mathbf{x}_1 = \mathbf{x}_2$ or $\mathbf{y}_1 = \mathbf{y}_2$ for which both couplings are equivalent in terms of transportation cost, and thus we can safely treat $\{\widetilde{E}_A \cap [0, 2\pi), \widetilde{E}_B \cap [0, 2\pi)\}$ as a disjoint partition of $[0, 2\pi)$, selecting a single coupling deterministically if both are optimal.

Observe that $|\langle \mathbf{v}, \mathbf{x}_i - \mathbf{y}_j\rangle| = |\langle -\mathbf{v}, \mathbf{x}_i - \mathbf{y}_j\rangle|$ and thus $\mathbf{v}$ and $-\mathbf{v}$ always induce the same optimal couplings. This means that orthogonal coupling of $\mathbf{v}_1 = [\cos(\phi_1), \sin(\phi_1)]^\top$ and $\mathbf{v}_2 = [\cos(\phi_2), \sin(\phi_2)]^\top$ is equivalent to setting $\phi_2 = \phi_1 \pm \frac{\pi}{2}$, which means both of the orthogonal vectors induce the same set of optimal couplings. Therefore

$$\begin{aligned}\mathbb{P}((\phi_1, \phi_2) \in \widetilde{E}_k \times \widetilde{E}_l \cap [0, 2\pi)^2) &= \tfrac{1}{2}\mathbb{P}(\phi_1 \in \widetilde{E}_k \cap \{\widetilde{E}_l + \tfrac{\pi}{2}\} \cap [0, 2\pi)) + \tfrac{1}{2}\mathbb{P}(\phi_1 \in \widetilde{E}_k \cap \{\widetilde{E}_l - \tfrac{\pi}{2}\} \cap [0, 2\pi)) \\ &= \ \mathbb{P}(\phi_1 \in \widetilde{E}_k \cap \{\widetilde{E}_l + \tfrac{\pi}{2}\} \cap [0, 2\pi))\,,\end{aligned}$$

as $\widetilde{E}_k \cap \{\widetilde{E}_l + \frac{\pi}{2}\} = \widetilde{E}_k \cap \{\widetilde{E}_l - \frac{\pi}{2}\}$, for any $k, l \in \{A, B\}$. Because $\mathbf{v} \sim \mathrm{Unif}(S^1)$ is equivalent to $\phi \sim \mathrm{Unif}([0, 2\pi))$,

$$\begin{aligned}&\mathbb{P}(\phi \in \widetilde{E}_A \cap \{\widetilde{E}_A + \tfrac{\pi}{2}\} \cap [0, 2\pi)) = 2[(p - \tfrac{1}{2}) \vee 0]\,, \\ &\mathbb{P}(\phi \in \widetilde{E}_B \cap \{\widetilde{E}_B + \tfrac{\pi}{2}\} \cap [0, 2\pi)) = 2[(\tfrac{1}{2} - p) \vee 0]\,, \\ &\mathbb{P}(\phi \in \widetilde{E}_A \cap \{\widetilde{E}_B + \tfrac{\pi}{2}\} \cap [0, 2\pi)) = \mathbb{P}(\phi \in \widetilde{E}_B \cap \{\widetilde{E}_A + \tfrac{\pi}{2}\} \cap [0, 2\pi)) = \tfrac{1}{2} - |\tfrac{1}{2} - p|\,,\end{aligned}$$

with $p := \mathbb{P}(\phi \in \widetilde{E}_A)$. The above equations combined with the definition of orthogonal sampling and the piecewise constant character of $f(\mathbf{v})$ in the case of the orthogonal estimation of the projected Wasserstein distance implies that all conditions of Definition 4.2 are satisfied, proving the first part of our proposition.

Turning to estimation of the sliced Wasserstein distance, we will reduce the computation of the MSE for both the i.i.d. and the orthogonal case to analytically solvable integrals, and use those to find examples of datasets for which either i.i.d. or orthogonal estimation is superior to the other. First, the expectation

$$\mathbb{E}[\mathrm{W}_p^p((\Pi_\mathbf{v})_\#\eta, (\Pi_\mathbf{v})_\#\mu)] = \frac{1}{2}\sum_{k \in \{A,B\}} \mathbb{P}(\mathbf{v} \in E_k)\, \mathbb{E}[|\langle \mathbf{v}, \mathbf{x}_1 - \mathbf{y}_{\sigma_k(1)}\rangle|^p + |\langle \mathbf{v}, \mathbf{x}_2 - \mathbf{y}_{\sigma_k(2)}\rangle|^p \mid \mathbf{v} \in E_k]\,,$$

can be solved by using the harmonic addition identity $\langle \mathbf{v}, \mathbf{z}\rangle = \|\mathbf{z}\| \cos(\phi - \rho)$ with ρ the angle between $\mathbf{z}$ and the x-axis $[1, 0]^\top \in \mathbb{R}^2$, for $\mathbf{v} = [\cos(\phi), \sin(\phi)]^\top$ and any $\mathbf{z} \in \mathbb{R}^2$. Evaluation of the above expectation then reduces to computation of a weighted sum of integrals of the form $\int_{\widetilde{E}_k \cap [0,2\pi)} |\cos(\phi - \rho)|^p \mathrm{d}\phi$ which can be solved using basic identities. The approach is analogous for the second moment, with the only difference being that the integrals will be of the form $\int_{\widetilde{E}_k \cap [0,2\pi)} |\cos(\phi - \rho)|^p |\cos(\phi - \gamma)|^p \mathrm{d}\phi$, where ρ and γ are the relevant angles between $\mathbf{x}_i - \mathbf{y}_j$ and the x-axis. Finally, the expression

$$\begin{aligned}&\mathbb{E}[\mathrm{W}_p^p((\Pi_{\mathbf{v}_1})_\#\eta, (\Pi_{\mathbf{v}_1})_\#\mu)\mathrm{W}_p^p((\Pi_{\mathbf{v}_2})_\#\eta, (\Pi_{\mathbf{v}_2})_\#\mu)] \\ &= \frac{1}{2^2}\sum_{k,l \in \{A,B\}} \sum_{m,n}^{2} \mathbb{P}((\mathbf{v}_1 \mathbf{v}_2) \in E_k \times E_l)\, \mathbb{E}[|\langle \mathbf{v}, \mathbf{x}_m - \mathbf{y}_{\sigma_k(m)}\rangle|^p |\langle \mathbf{v}, \mathbf{x}_n - \mathbf{y}_{\sigma_l(n)}\rangle|^p \mid (\mathbf{v}_1 \mathbf{v}_2) \in E_k \times E_l]\,,\end{aligned}$$

can be computed in fashion similar to that of the second moment, with the integrals now being $\int_{\widetilde{E}_k \cap \{\widetilde{E}_l + \frac{\pi}{2}\} \cap [0,2\pi)} |\cos(\phi - \rho)|^p |\cos(\phi - \gamma)|^p \mathrm{d}\phi$.

Putting all these together, an example of a dataset for which i.i.d. estimation of the 1-sliced Wasserstein distance strictly dominates orthogonal is $\mathbf{x}_1 = [1.23, -2.17]^\top$, $\mathbf{x}_2 = [-2, -0.65]^\top$, $\mathbf{y}_1 = [-0.14, -0.93]^\top$, $\mathbf{y}_2 = [-0.82, 0.43]^\top$, where the MSEs of the i.i.d. and orthogonal estimators for $N = 2$ are respectively ≈ 0.011900 and ≈ 0.017085, and the distance itself equals ≈ 1.082029. A dataset for which the orthogonal coupling dominates is $\mathbf{x}_1 = [1, 1]^\top$, $\mathbf{x}_2 = [0, 0]^\top$, $\mathbf{y}_1 = [0, -\frac{1}{2}]^\top$, $\mathbf{y}_2 = [1, -1]^\top$, where the MSEs of the i.i.d. and orthogonal estimators for $N = 2$ are respectively ≈ 0.073996 and ≈ 0.006047, and the distance itself equals ≈ 0.795774. Neither i.i.d. nor orthogonal estimation thus strictly dominates the other in terms of MSE across all p-sliced Wasserstein distances. $\square$

9 Appendix on Experiments

9.1 Generative Modelling: Auto-encoders

We consider sliced Wasserstein Auto-encoders (AE) (Kolouri et al., 2018) on MNIST dataset. MNIST dataset contains 50000 training gray images each with dimension 28×28. To facilitate HD projection, we augment the images to 32×32 by padding zeros. Hence in our case the observations have dimension $\mathbf{x} \in \mathbb{R}^{32 \times 32}$.

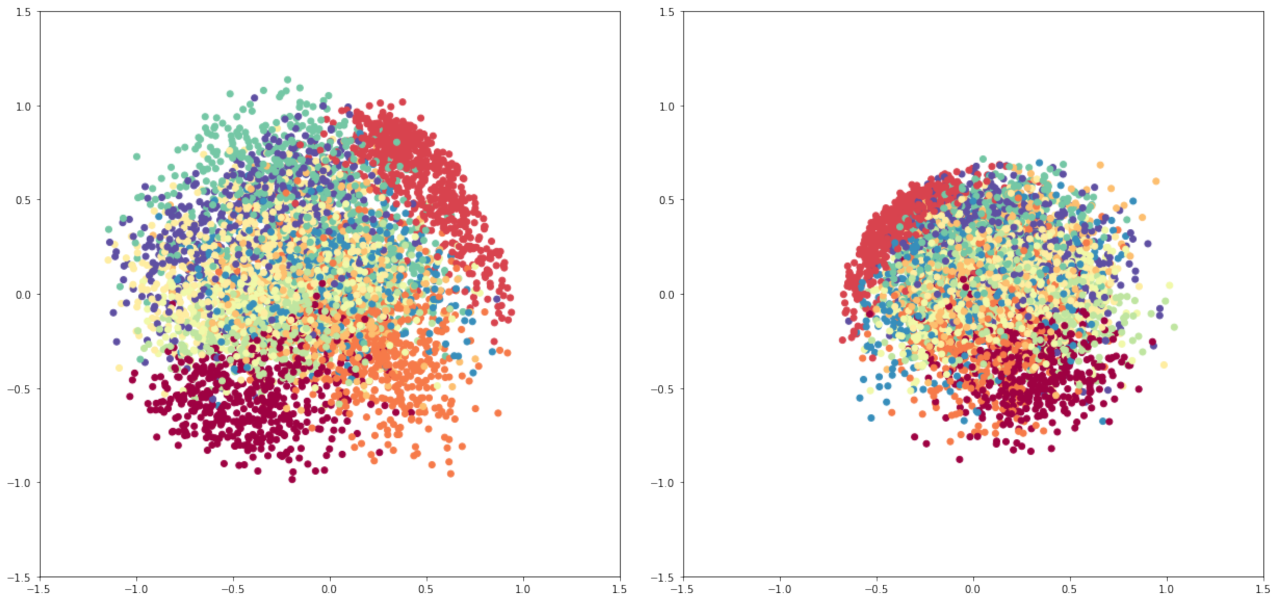

Figure 5: Training AEs using two sets of distance measure (left: sliced Wasserstein distance, right: projected Wasserstein distance): We show hidden codes $p_\theta(\mathbf{z})$ generated by the encoder after training. Though both distributions generally match the prior distribution $p(\mathbf{z})$, the distribution trained with projected Wasserstein distance tends to collapse to the center.

Implementation Details. The AEs have the same architecture as introduced in Kolouri et al. (2018). The hidden code $\mathbf{z}$ has dimension $h = 128$. The prior distribution $p(\mathbf{z})$ is chosen to be a uniform distribution inside $[-1, 1]^h$. For each iteration, we take a full sweep over the dataset in a random order. All implementations are in Tensorflow (Abadi et al., 2016) and Keras (Chollet et al., 2015), we also heavily refer to the code of the original authors of (Kolouri et al., 2018) [1].

9.2 Generative Modelling: sliced Wasserstein vs. projected Wasserstein

Background. Here we present the comparison between training AE using sliced Wasserstein distance vs. projected Wasserstein distance. Recall that the training objective of the AE is in general

$$L_{\text{ae}}(\theta, \phi) = \mathbb{E}_{\mathbf{x} \sim P(X)}[\|g_\phi(f_\theta(\mathbf{x})) - \mathbf{x}\|] + D(p_\theta(\mathbf{z}), p(\mathbf{z})),$$

where $D(\cdot, \cdot)$ can be some proper discrepancy measure between two distributions. In practice, $D(\cdot, \cdot)$ can be KL-divergence (Kingma and Welling, 2013), sliced Wasserstein distance (Kolouri et al., 2018) or projected Wasserstein distance (this work). All the aforementioned alternates allow for fast optimization using gradient descent on the discrepancy measures.

Implementation Details. The AEs have the same architecture as introduced in Kolouri et al. (2018). The hidden code $\mathbf{z}$ has dimension $h = 2$. The prior distribution $p(\mathbf{z})$ is chosen to be a uniform distribution on the interior of the 2D circle with radius 1. The other implementation details are the same as above.

Results. We compare the posterior hidden codes $\mathbf{z} \sim p_\theta(\mathbf{z})$ generated by the trained encoders under sliced Wasserstein distance (left) vs. projected Wasserstein distance (right) in Figure 5. Though both hidden code distributions largely match that of the prior distribution $p(\mathbf{z})$, the hidden codes trained by sliced Wasserstein distance tend to collapse to the center (the distribution has a slightly smaller effective support). This observation is compatible with our observations in the generator network experiments presented in the next section.

[1] `https://github.com/skolouri/swae`

9.3 Generative Modelling: Generator Networks

Background. A generator network G_θ takes as input a noise sample $\epsilon \sim \rho_0(\cdot)$ from an elementary distribution ρ_0 (e.g. Gaussian) and output a sample in the target domain $\mathcal{X}$ (e.g. images), i.e. $X = G_\theta(\epsilon) \in \mathcal{X}$. Let $P_\theta(X)$ be the implicit distribution induced by the network G_θ and noise source ρ_0 over samples X. We also have a target distribution $\hat{P}(X)$ (usually an empirical distribution constructed using samples) that we aim to model. The objective of generative modeling is to find parameters θ such that $P_\theta(X) \approx \hat{P}(X)$ by minimizing certain discrepancies

$$D(P_\theta(X), \hat{P}(X)), \tag{22}$$

for some discrepancy measure $D(\cdot,\cdot)$. When $D(\cdot,\cdot)$ is taken to be the Jenson-Shannon divergence, we recover the objective of Generative Adversarial Networks (GAN) (Goodfellow et al., 2014). Recently, Wu et al. (2017); Deshpande et al. (2018) propose to take $D(\cdot,\cdot)$ to be the sliced Wasserstein distance, so as to bypass the potential instability due to min-max optimization formulation with GAN. Similarly, $D(\cdot,\cdot)$ can be projected Wasserstein distance. Here, we show the empirical differences of these two generative models (sliced Wasserstein generative network vs. projected Wasserstein generative network) with an illustrating example.

Setup. We take $\mathcal{X} = \mathbb{R}^2$ and $\hat{P}(X)$ to be the empirical distribution formed by samples drawn from a mixture of Gaussians. The mixture contains 16 components with centers evenly spaced on the 2-D grid with horizontal/vertical distance between neighboring centers to be 0.3. Each Gaussian is factorized wit diagonal variance 0.1^2. The samples are illustrated as the red points in Figure 6 below.

Implementation Details. The generators are parameterized as neural networks which take $2-$dimensional noise (drawn from a standard factorized Gaussian) as input and output samples in $\mathcal{X} = \mathbb{R}^2$. The networks have two hidden layers each with 256 units, with relu nonlinear function activation in between. The final output layer has tanh nonlinear activation. We train all models with Adam Optimizer and learning rate 10^{-4} until convergence. All implementations are in Tensorflow (Abadi et al., 2016), we also heavily refer to a set of wonderful open source projects [2] [3].

Results. The results for generative modelling are in Figure 6 (left for sliced Wasserstein distance, right for project Wasserstein distance). Red samples are those generated from the target distribution. Blue samples are those generated from the generator network after training until convergence. We observe that samples generated from these two models exhibit distinct features: under sliced Wasserstein distance, the samples tend to be more widespread and in this case capture the modes on the perimeter of the Gaussian mixtures. On the other hand, under projected Wasserstein distance, the samples tend to collapse to the center of the target distribution and only capture modes in the middle.

9.4 Reinforcement Learning

Background. Vanilla policy gradient updates $\theta_{\text{new}} \leftarrow \theta_{\text{old}} + \alpha \nabla_{\theta_{\text{old}}} J(\pi_{\theta_{\text{old}}})$ suffer from occasionally large step sizes, which lead the policy to collect bad samples from which one could never recover (Schulman et al., 2015). Trust region policy optimization (TRPO) (Schulman et al., 2015) propose to constrain the update using KL divergence $\mathbb{KL}[\pi_{\theta_{old}} || \pi_{\theta_{new}}] \leq \epsilon$ for some $\epsilon > 0$, which can be shown to optimize a lower bound of the original objective and significantly stabilize learning in practice. Recently, Zhang et al. (2018) interpret policy optimization as discretizing the differential equation of the Wasserstein gradient flows, and propose to construct trust regions using Wasserstein distance. In general, trust region constraints are enforced by $D(\pi_{\theta_{\text{old}}}, \pi_{\theta_{\text{new}}}) \leq \epsilon$ for some ϵ, where Schulman et al. (2015) use KL divergence while Zhang et al. (2018) use Wasserstein distance.

Instead of constructing the constraints explicitly, one can adopt a penalty formulation of the trust region and apply (Schulman et al., 2017; Zhang et al., 2018)

$$\theta_{\text{new}} \leftarrow \theta_{\text{old}} + \alpha \nabla_{\theta_{\text{old}}} (J(\pi_{\theta_{\text{old}}}) - \lambda D(\pi_{\theta_{\text{old}}}, \pi_{\theta_{\text{new}}})),$$

where $\lambda > 0$ is a trade-off constant. The above updates encourage the new policy $\pi_{\theta_{\text{new}}}$ to achieve higher rewards but also stay close to the old policy for stable updates.

[2]`https://github.com/kvfrans/generative-adversial`
[3]`https://github.com/ishansd/swg`

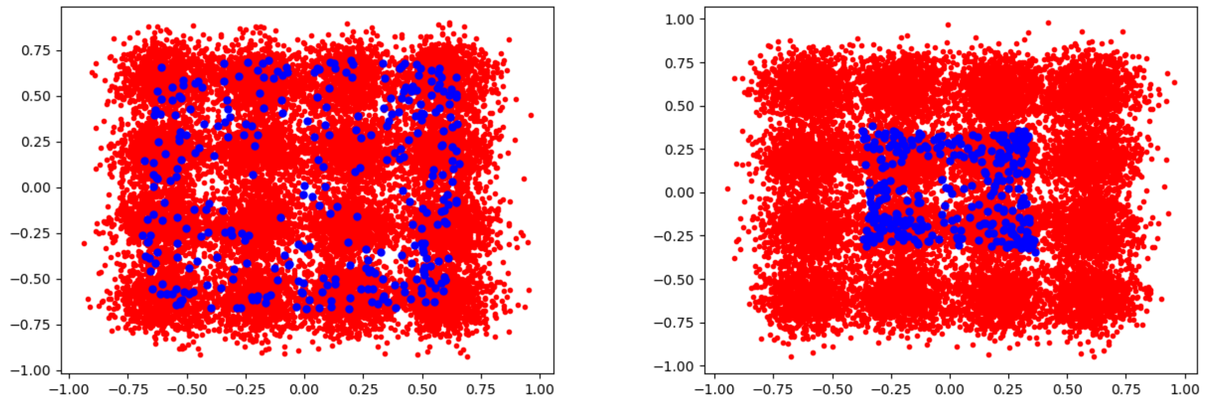

Figure 6: Training generators using two sets of distance measure (left: sliced Wasserstein distance, right: projected Wasserstein distance): Red samples are form the target distributions, which are drawn from a mixture of Gaussians with 16 mixtures. Blue samples are those generated from the generator network after convergence. Under sliced Wasserstein distance the samples tend to spread out and capture modes at the perimeter of the mixtures, while under projected Wasserstein distance the samples tend to cluster in the center and capture modes in the middle.

Due to the close connections between various Wasserstein distance measures, we propose to set $D(\cdot,\cdot)$ as either sliced Wasserstein distance or projected Wasserstein distance. Since projected Wasserstein distance corrects for the implicit "bias" in the sliced Wasserstein distance, we expect the corresponding trust region to be more robust and can better stabilize on-policy updates.

Implementation Details. The policy π_θ is parameterized as feed-forward neural networks with two hidden layers, each with $h = 64$ units with tanh non-linear activations. The value function baseline is a neural network with similar architecture. To implement vanilla policy optimization, we use PPO with very large clipping rate $\epsilon = 10.0$, which is equivalent to no clipping. We set the learning rate to be $\alpha = 3 \cdot 10^{-5}$ and the trade-off constant to be $\lambda = 0.001$ for the trust region. All implementations are based on OpenAI baseline (Dhariwal et al., 2017) and benchmark tasks are from OpenAI gym (Brockman et al., 2016).

9.5 Hadamard-Rademacher random matrices

Here, we give brief details around Hadamard-Rademacher random matrices, which are studied in Section 5.2 as an approximate alternative to using random orthogonal matrices drawn from $\mathrm{UnifOrt}(S^{d-1}; d)$. These random matrices have been used as computationally cheap alternatives to exact sampling from $\mathrm{UnifOrt}(S^{d-1}; d)$ in a variety of applications recently; see e.g. (Choromanski et al., 2017; Andoni et al., 2015). A 1-block Hadamard-Rademacher matrix is simulated by taking $\mathbf{H}$ to be a normalised Hadamard matrix in $\mathbb{R}^{d\times d}$, and $\mathbf{D}$ to be a random diagonal matrix, with independent Rademacher ($\mathrm{Unif}(\{\pm 1\})$) random variables along the diagonal. The Hadamard-Rademacher matrix is then given by the product $\mathbf{HD}$. Multi-block Hadamard-Rademacher random matrices are given by taking the product of several independent 1-block Hadamard-Rademacher random matrices.