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* $\mathscr{P}_{(M)}(\mathbb{R}^d) = \{\frac{1}{M}\sum_{m=1}^M \delta_{\mathbf{x}_m} | \mathbf{x}_m \in \mathbb{R}^d \text{ for all } m \in [M]\} \subset \mathscr{P}(\mathbb{R}^d)$.

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 $\sigma : [M] \to [M]$, 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

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

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*

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

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

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

$$\mathrm{PW}_p^p(\eta,\mu) = \mathbb{E}_{\mathbf{v}\sim\mathrm{Unif}(S^{d-1})}\left[\frac{1}{M}\sum_{m=1}^{M}\|\mathbf{x}_m - \mathbf{y}_{\sigma_{\mathbf{v}}(m)}\|_2^p\right] \tag{17}$$

$$\geq \mathbb{E}_{\mathbf{v}\sim\mathrm{Unif}(S^{d-1})}\left[\min_{\sigma\in\mathcal{S}_M}\frac{1}{M}\sum_{m=1}^{M}\|\mathbf{x}_m - \mathbf{y}_{\sigma(m)}\|_2^p\right] \tag{18}$$

$$= \min_{\sigma\in\mathcal{S}_M}\frac{1}{M}\sum_{m=1}^{M}\|\mathbf{x}_m - \mathbf{y}_{\sigma(m)}\|_2^p \tag{19}$$

$$= \mathrm{W}_p^p(\eta,\mu)\,, \tag{20}$$

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: $\eta = \frac{1}{M}\sum_{m=1}^{M}\delta_{\mathbf{x}_m}$, $\mu = \frac{1}{M}\sum_{m=1}^{M}\delta_{\mathbf{y}_m}$
Sample $(\mathbf{v}_n)_{n=1}^{N} \sim \mathrm{UnifOrt}(S^{d-1};N)$
for $n = 1$ **to** N **do**
  Compute projected distributions:
  $(\Pi_{\mathbf{v}_n})_{\#}\eta = \frac{1}{M}\sum_{m=1}^{M}\delta_{\langle\mathbf{v}_n,\mathbf{x}_m\rangle}$
  $(\Pi_{\mathbf{v}_n})_{\#}\mu = \frac{1}{M}\sum_{m=1}^{M}\delta_{\langle\mathbf{v}_n,\mathbf{y}_m\rangle}$
  Compute optimal matching for projected distributions:
  $\sigma_{\mathbf{v}_n} \leftarrow \mathtt{argsort}((\langle\mathbf{v}_n,\mathbf{x}_m\rangle)_{m=1}^{M}, (\langle\mathbf{v}_n,\mathbf{y}_m\rangle)_{m=1}^{M})$
  Compute contribution from coupling:
  $\frac{1}{M}\sum_{m=1}^{M}\|\mathbf{x}_m - \mathbf{y}_{\sigma_{\mathbf{v}_n}(m)}\|^p$
end for
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

Sample $(\mathbf{A}_{ij})_{i,j=1}^{d} \overset{\text{i.i.d.}}{\sim} N(0,1)$
Normalise the norms of the rows of $\mathbf{A}$ to 1.
Perform Gram-Schmidt orthogonalisation on the rows of $\mathbf{A}$ to obtain $\widetilde{\mathbf{A}}$
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 $p_{k,l}^{(i,j)} \coloneqq \mathbb{P}(X_i \in A_k, X_j \in A_l)$, $p_k = \mathbb{P}(X \in A_k)$, and $s_k \coloneqq \mathbb{E}[f(X) \,|\, X \in A_k]$. Defining the matrix $[\mathbf{P}^{(i,j)}]_{k,l} \coloneqq p_{k,l}^{(i,j)}$ and the vector $[\mathbf{p}]_k \coloneqq p_k$ 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 $\widetilde{E}_A \coloneqq (H_{\mathbf{x}}^+ \cap H_{\mathbf{y}}^+) \cup (H_{\mathbf{x}}^- \cap H_{\mathbf{y}}^-)$ and $\widetilde{E}_B \coloneqq (H_{\mathbf{x}}^+ \cap H_{\mathbf{y}}^-) \cup (H_{\mathbf{x}}^- \cap H_{\mathbf{y}}^+)$ 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 $p \coloneqq \mathbb{P}(\phi \in \widetilde{E}_A)$. 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

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

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

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

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. □

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 $\mathbf{x} \in \mathbb{R}^{32 \times 32}$.

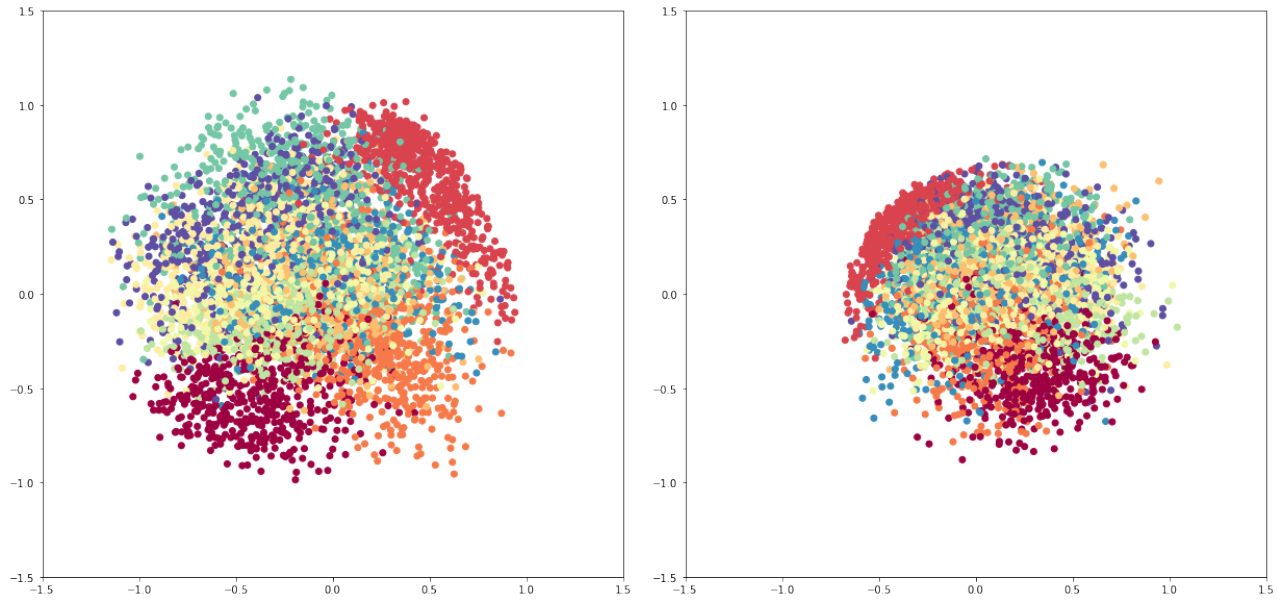

Figure 5: Training AEs using two sets of distance measure (left: sliced Wasserstein distance, right: projected Wasserstein distance): We show hidden codes $p_\theta(\mathbf{z})$ generated by the encoder after training. Though both distributions generally match the prior distribution $p(\mathbf{z})$,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

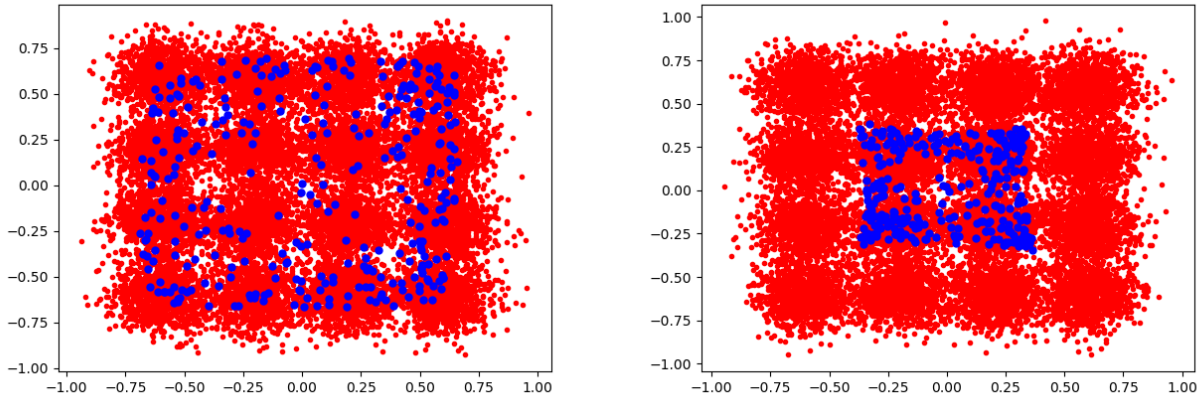

Figure 6: Training generators using two sets of distance measure (left: sliced Wasserstein distance, right: projected Wasserstein distance): Red samples are form the target distributions, which are drawn from a mixture of Gaussians with 16 mixtures. Blue samples are those generated from the generator 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.