Supplement

This supplement is structured as follows:

- Appendix A contains the proofs of the theory developed in the main text: Proposition 1 (Appendix A.1), Proposition 2 (Appendix A.2), Theorem 2 (Appendix A.3) and Proposition 3 (Appendix A.4).
- Appendix B contains the computational details and extensions to the experiments detailed in Section 4. Appendix B.1 explains how the Wasserstein-1 distance was computed as our performance metric. Appendix B.2 provides the full details of the synthetic test-bed experiments used in Section 4.1. Similarly, Appendix B.3 and Appendix B.4 provides the full details of the biochemical oxygen demand experiment of Section 4.2 and Section 4.3 respectively.
- Appendix C contains further investigations into the methods presented in the main paper. Appendix C.1 discusses the sensitivity to initialisation and explains how the transport maps were initialised for the experiments of Section 4. Appendix C.2 explores variations on the stochastic optimisation method. Appendix C.3 explores the use of quasi Monte Carlo in the stochastic approximation of gradients for KSD. In Appendix C.4, we investigate the effect of the changing the reference distribution. In Appendix C.5, we explore the effect of the length-scale parameter ℓ on KSD-based measure transport. In Appendix C.6, we investigate KSD-based measure transport on high-dimensional targets. Appendix C.7 explores the effect of the input dimension in the ReLU network transport map. In Appendix C.8, we investigate KSD-based measure transport using a (biased) V-statistic estimator of KSD against the unbiased U-statistic estimator that was used for the experiments presented in the main text. Finally, in Appendix C.9, we document certain pathological behaviours experienced when using KSD for measure transport and offer potential remedies.

A Proof of Theoretical Results

This section contains the proofs for all novel theoretical results stated in the main text. In Appendix A.1 we present the proof of Proposition 1; in Appendix A.2 we present the proof of Proposition 2; the proof of Theorem 2 is contained in Appendix A.3, and finally, the proof of Proposition 3 is in Appendix A.4.

A.1 Proof of Proposition 1

The argument involved in the proof of Proposition 1 requires differentiation under an integral. The measure-theoretic calculus result that we exploit to justify the interchange of differentiation and integration (Lemma 2 below) requires the following mathematical concepts:

Definition 2. *Let Ω be a measurable space and let Θ be a topological space. A function $f : \Theta \times \Omega \to \mathbb{R}$ is a* Carathéodory function *if for each $\theta \in \Theta$ the map $\omega \mapsto f(\theta, \omega)$ is measurable and for each $\omega \in \Omega$ the map $\theta \mapsto f(\theta, \omega)$ is continuous.*

Definition 3. *Let Ω be a measurable space equipped with a measure μ and let Θ be a topological space. A function $f : \Theta \times \Omega \to \mathbb{R}$ is* locally uniformly integrably bounded *if for every $\theta \in \Theta$ there is a non-negative measurable function $h_\theta : \Omega \to \mathbb{R}$ such that $\int_\Omega h_\theta(\omega) \mathrm{d}\mu(\omega) < \infty$, and there exists a neighbourhood U_θ of θ such that for all $\vartheta \in U_\theta$ we have $|f(\vartheta, \omega)| \leq h_\theta(\omega)$.*

The following sufficient condition for a function to be locally uniformly integrably bounded will be used:

Lemma 1. *In the setting of Definition 3, let $\Theta \subseteq \mathbb{R}^d$ be an open set and assume further that, for each $\omega \in \Omega$, the function $\theta \mapsto f(\theta, \omega)$ is continuous and that, for each $\theta \in \Theta$, the integral $\int_\Omega |f(\theta, \omega)| \mathrm{d}\mu(\omega) < \infty$ exists. Then f is locally uniformly integrably bounded.*

Proof. Fix $\theta \in \Theta$ and $\omega \in \Omega$. Since $f(\theta, \omega)$ is continuous in θ and Θ is open, we can find a neighbourhood U_θ of θ on which $f(\vartheta, \omega) \leq f(\theta, \omega) + 1$ for all $\vartheta \in U_\theta$. Take $h_\theta(\omega) := |f(\theta, \omega)| + 1$, recalling that the absolute value of a measurable function is measurable and sums of measurable functions are measurable. Then $\int_\Omega h_\theta(\omega) \mathrm{d}\mu(\omega) = \int_\Omega |f(\theta, \omega)| \mathrm{d}\mu(\omega) + 1 < \infty$ and $|f(\theta, \omega)| \leq h_\theta(\omega)$, as required. □

Lemma 2 (Differentiate under the integral)**.** *Let Ω be a measurable space equipped with a measure μ, let $\Theta \subseteq \mathbb{R}^d$ be an open set and let $f : \Theta \times \Omega \to \mathbb{R}$ be a Carathéodory function. Assume further that f is locally uniformly integrably bounded and that, for each i and each ω, the function $\theta \mapsto \partial_{\theta_i} f(\theta, \omega)$ is locally uniformly integrably bounded. Then the function $g : \Theta \to \mathbb{R}$ defined by*

$$g(\theta) := \int_\Omega f(\theta, \omega) \mathrm{d}\mu(\omega)$$

is continuously differentiable and

$$\nabla_\theta g(\theta) = \int_\Omega \nabla_\theta f(\theta, \omega) \mathrm{d}\mu(\omega).$$

Proof. This standard result can be found, for example, in Aliprantis and Burkinshaw (1998, Theorem 24.5, p.193), Billingsley (1979, Theorem 16.8, pp.181–182), with the statement here based on the account in Border (2016). □

The proof of Proposition 1 can now be presented:

Proof of Proposition 1. Using (5) and the reparametrisation trick (Glynn, 1986; L'Ecuyer, 1995; Kingma and Welling, 2013):

$$\begin{aligned} \nabla_\theta \left[\mathcal{D}_{\mathrm{S}}(P, T^\theta_\# Q)^2 \right] &= \nabla_\theta \mathbb{E}_{Y,Y' \sim T^\theta_\# Q} \left[u_p(Y, Y') \right] \\ &= \nabla_\theta \mathbb{E}_{X,X' \sim Q} \left[u_p(T^\theta(X), T^\theta(X')) \right] \end{aligned} \tag{13}$$

From Lemma 2, the preconditions of Proposition 1 justify the interchange of the derivative and the expectation in (13). Indeed, in the setting of Lemma 2 we identify $\Omega = \mathcal{X} \times \mathcal{X}$, $\mu = Q \times Q$ and $f(\theta, \omega) = u_p(T^\theta(x), T^\theta(x'))$ where $\omega = (x, x')$. That f is a Carathéodory function follows from (A1) and (A4) of Proposition 1, where we note that (A4) implies $\theta \mapsto \nabla_\theta u_p(T^\theta(x), T^\theta(x'))$ is continuous. That f is locally uniformly integrably bounded follows from assumptions (A2) and (A4) together with Lemma 1. Similarly, that $\partial_{\theta_i} f$ is locally uniformly integrably bounded follows from assumptions (A3) and (A4) together with Lemma 1. Thus the preconditions of Lemma 2 hold.

Interchanging the derivative with the expectation gives that

$$\begin{aligned} \nabla_\theta \left[\mathcal{D}_{\mathrm{S}}(P, T^\theta_\# Q)^2 \right] &= \mathbb{E}_{X,X' \sim Q} \left[\nabla_\theta u_p(T^\theta(X), T^\theta(X')) \right] \\ &= \mathbb{E} \left[\frac{1}{n(n-1)} \sum_{i \neq j}^{n} \nabla_\theta u_p(T^\theta(x_i), T^\theta(x_j)) \right], \end{aligned}$$

as claimed. □

Note that we presented stronger conditions in Proposition 1 than are required. This was to control the length of the main text, but it is immediately clear from the proof of Proposition 1 that these conditions can be weakened to those that are required for Lemma 2 to hold.

A.2 Proof of Proposition 2

First we present an existence result in Proposition 4, before considering regularity of the associated transport map. Recall that in this paper all measurable spaces $\mathcal{X}$ and $\mathcal{Y}$ are equipped with their respective Borel σ-algebras $\Sigma_\mathcal{X}$ and $\Sigma_\mathcal{Y}$. A separable complete metric space equipped with its Borel σ-algebra is called a *standard Borel space*. A measure $Q \in \mathcal{P}(\mathcal{X})$ is said to be *continuous* if $Q(\{x\}) = 0$ for all $x \in \mathcal{X}$. A map $f : \mathcal{X} \to \mathcal{Y}$ is called a *Borel isomorphism* if f is a bijection and both f and f^{-1} are Borel measurable. A fundamental result that we will exploit is known as the *isomorphism theorem for measures*:

Theorem 3 (Isomorphism Theorem)**.** *Let $\mathcal{X}$ be a standard Borel space and $Q \in \mathcal{P}(\mathcal{X})$ be continuous. Then there is a Borel isomorphism $f : \mathcal{X} \to [0, 1]$ with $f_\# Q = m|_{[0,1]}$, where $m|_{[0,1]}$ is the Lebesgue measure restricted to $[0, 1]$.*

Proof. This result can be found as Theorem 17.41 in Kechris (1995). □

Proposition 4. *Suppose that $\mathcal{X}$ and $\mathcal{Y}$ are separable complete metric spaces and suppose that $Q \in \mathcal{P}(\mathcal{X})$ and $P \in \mathcal{P}(\mathcal{Y})$ are such that $Q(\{x\}) = 0$ and $P(\{y\}) = 0$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Then there exists a measurable function $T : \mathcal{X} \to \mathcal{Y}$ such that $T_\# Q = P$.*

Proof. Our assumptions imply that $\mathcal{X}$, $\mathcal{Y}$ are standard Borel spaces and $Q \in \mathcal{P}(\mathcal{X})$, $P \in \mathcal{P}(\mathcal{Y})$ are continuous. Thus from Theorem 3, there exists a Borel isomorphism $f : \mathcal{X} \to [0, 1]$ such that $f_\# Q = m|_{[0,1]}$ and a Borel isomorphism $g : \mathcal{Y} \to [0, 1]$ such that $g_\# P = m|_{[0,1]}$. Then $T := g^{-1} \circ f : \mathcal{X} \to \mathcal{Y}$ is measurable and satisfies $T_\# Q = P$, as required. □

Now we can present the proof of Proposition 2.

Proof of Proposition 2. Assumption 1 ensures that $\mathcal{X}$ is a separable complete metric space and $Q(\{x\}) = 0$ for all $x \in \mathcal{X}$. Assumption 2 restricts attention to $\mathcal{Y} = \mathbb{R}^d$, meaning that $\mathcal{Y}$ is a separable complete metric space, and requires P to admit a density on $\mathcal{Y}$, meaning that $P(\{y\}) = 0$ for all $y \in \mathcal{Y}$. Thus the existence of a transport map T from Q to P is guaranteed by Proposition 4.

It remains to show that, for *any* such transport map, $T \in \prod_{i=1}^d L^2(Q)$. To this end, we have that

$$\begin{aligned}\|T\|^2_{\prod_{i=1}^d L^2(Q)} = \sum_{i=1}^d \|T_i\|^2_{L^2(Q)} = \sum_{i=1}^d \int_{\mathcal{X}} T_i(x)^2 \, \mathrm{d}Q(x) &= \int_{\mathcal{X}} \|T(x)\|^2 \, \mathrm{d}Q(x) \\ &\overset{(*)}{=} \int_{\mathbb{R}^d} \|x\|^2 \, \mathrm{d}(T_\# Q)(x) = \int_{\mathbb{R}^d} \|x\|^2 \, \mathrm{d}P(x) < \infty,\end{aligned}$$

where a change of variables was used at $(*)$ and the final inequality follows from the assumption that $P \in \mathcal{P}_2(\mathbb{R}^d)$ in Assumption 2. □

A.3 Proof of Theorem 2

Recall that for $P, P' \in \mathcal{P}_1(\mathbb{R}^d)$ the (first) *Wasserstein distance* is defined as (Villani, 2009, Remark 6.5)

$$W_1(P, P') := \sup_{f \in \mathcal{F}} \left| \mathbb{E}_{Y \sim P}[f(Y)] - \mathbb{E}_{Y \sim P'}[f(Y)] \right|, \tag{14}$$

where $\mathcal{F} := \left\{ f : \mathbb{R}^d \to \mathbb{R} \,|\, \|f\|_L \le 1 \right\}$ and $\|f\|_L := \sup_{x \neq y} \frac{|f(x) - f(y)|}{\|x - y\|}$ is the Lipschitz seminorm on $\mathbb{R}^d$. Our proof of Theorem 2 is based on the following result that relates convergence in W_1 to convergence in KSD:

Proposition 5 (Wasserstein Controls KSD)**.** *Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be symmetric positive definite with $(x, y) \mapsto k(x, y)$, $(x, y) \mapsto \partial_{x_i} \partial_{y_i} k(x, y)$ and $(x, y) \mapsto \partial_{x_i} \partial_{x_j} \partial_{y_i} \partial_{y_j} k(x, y)$ continuous and bounded for all $i, j \in \{1, \dots, d\}$. Let $P \in \mathcal{P}(\mathbb{R}^d)$ admit a density function p such that $\nabla \log p$ is Lipschitz with $\mathbb{E}_{X \sim P}[\|\nabla \log p(X)\|_2^2] < \infty$. Let $\mathcal{D}_{\mathrm{S}}$ denote the KSD based on P and k, as defined in (7). Then a sequence $(Q_n)_{n \in \mathbb{N}} \subset \mathcal{P}(\mathbb{R}^d)$ satisfies $\mathcal{D}_{\mathrm{S}}(P, Q_n) \to 0$ whenever $W_1(P, Q_n) \to 0$.*

Proof. This result is Proposition 9 of Gorham and Mackey (2017). □

Recall that in this paper $\mathcal{X}$ is always assumed to be a Borel space. The following result is also required:

Lemma 3 (L^2 Controls Wasserstein)**.** *Let $Q \in \mathcal{P}(\mathcal{X})$ and let $S, T \in \prod_{i=1}^d L^2(Q)$. Then we have the bound $W_1(S_\# Q, T_\# Q) \le \|S - T\|_{\prod_{i=1}^d L^2(Q)}$.*

Proof. From the definition of the (first) Wasserstein distance, we have

$$\begin{aligned} W_1(S_\# Q, T_\# Q) &= \sup_{\|f\|_L \le 1} \left| \int_{\mathcal{X}} f(x) \, \mathrm{d}S_\# Q(x) - \int_{\mathcal{X}} f(x) \, \mathrm{d}T_\# Q(x) \right| \\ &= \sup_{\|f\|_L \le 1} \left| \int_{\mathcal{X}} f(S(x)) - f(T(x)) \, \mathrm{d}Q(x) \right|. \end{aligned}$$

If $\|f\|_L \leq 1$ then $|f(a) - f(b)| \leq \|a - b\|$ for all $a, b \in \mathbb{R}^d$, and so

$$\begin{aligned} W_1(S_\# Q, T_\# Q) &\leq \int_{\mathcal{X}} \|S(x) - T(x)\| \, \mathrm{d}Q(x) \\ &\leq \left(\int_{\mathcal{X}} \|S(x) - T(x)\|^2 \, \mathrm{d}Q(x) \right)^{1/2} = \|S - T\|_{\prod_{i=1}^d L^2(Q)} \end{aligned}$$

where the second inequality is Jensen's inequality. □

Our final ingredient is a basic result on the inverse multi-quadric kernel:

Lemma 4 (Derivatives of the Inverse Multi-quadric Kernel)**.** *The inverse multi-quadric kernel in* (3)*, $k(x, y) = (c^2 + \|x - y\|^2)^\beta$, with $c > 0$ and $\beta \in (-1, 0)$, satisfies*

$$\sup_{x,y \in \mathbb{R}^d} \left| \partial_{x_1}^{\alpha_1} \dots \partial_{x_d}^{\alpha_d} \partial_{y_1}^{\alpha_1} \dots \partial_{y_d}^{\alpha_d} \, k(x, y) \right| < \infty$$

for all multi-indices $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}_0^d$.

Proof. For $\alpha \in \mathbb{N}_0^d$ let $|\alpha| := \alpha_1 + \cdots + \alpha_d$. Recall that a polynomial $\prod_{|\alpha| \leq s} c_\alpha z_1^{\alpha_1} \dots z_d^{\alpha_d}$ is said to have *maximal degree* s, where $s = |\alpha|$ is the largest integer for which $c_\alpha \neq 0$ for some $\alpha \in \mathbb{N}_0^d$. Let

$$\mathcal{F} := \left\{ (x, y) \mapsto k(x, y) \frac{r_m(x - y)}{(c^2 + \|x - y\|^2)^m} : r_m \text{ is a polynomial of maximal degree } 2m, \ m \in \mathbb{N}_0 \right\}.$$

Then $k(x, y) \in \mathcal{F}$ and $\mathcal{F}$ is closed under the action of each of the differential operators $\partial_{x_i} \partial_{y_i}$, $i \in \{1, \dots, d\}$. Indeed, we have from the product rule that

$$\partial_{x_i} \left[k(x, y) \tfrac{r_m(x-y)}{(c^2+\|x-y\|^2)^m} \right] = \tfrac{2\beta(x_i - y_i) k(x,y) r_m(x-y)}{(c^2+\|x-y\|^2)^{m+1}} + \tfrac{k(x,y) \partial_{x_i} r_m(x-y)}{(c^2+\|x-y\|^2)^m} - \tfrac{2m(x_i-y_i)k(x,y)r_m(x-y)}{(c^2+\|x-y\|^2)^{m+1}}$$

and

$$\begin{aligned} \partial_{x_i} \partial_{y_i} \left[k(x, y) \tfrac{r_m(x-y)}{(c^2+\|x-y\|^2)^m} \right] = &\left[-\tfrac{2\beta k(x,y) r_m(x-y)}{(c^2+\|x-y\|^2)^{m+1}} - \tfrac{4\beta^2 (x_i-y_i)^2 k(x,y) r_m(x-y)}{(c^2+\|x-y\|^2)^{m+2}} \right. \\ &\quad \left. + \tfrac{2\beta(x_i-y_i)k(x,y)\partial_{y_i} r_m(x-y)}{(c^2+\|x-y\|^2)^{m+1}} + \tfrac{4(m+1)\beta(x_i-y_i)^2 k(x,y) r_m(x-y)}{(c^2+\|x-y\|^2)^{m+2}} \right] \\ &+ \left[-\tfrac{2\beta(x_i-y_i)k(x,y)\partial_{x_i} r_m(x-y)}{(c^2+\|x-y\|^2)^{m+1}} + \tfrac{k(x,y)\partial_{x_i}\partial_{y_i} r_m(x-y)}{(c^2+\|x-y\|^2)^m} \right. \\ &\quad \left. + \tfrac{2m(x_i-y_i)k(x,y)\partial_{x_i} r_m(x-y)}{(c^2+\|x-y\|^2)^{m+1}} \right] \\ &+ \left[\tfrac{2mk(x,y)r_m(x-y)}{(c^2+\|x-y\|^2)^{m+1}} + \tfrac{4\beta m(x_i-y_i)^2 k(x,y) r_m(x-y)}{(c^2+\|x-y\|^2)^{m+2}} \right. \\ &\quad \left. - \tfrac{2m(x_i-y_i)k(x,y)\partial_{y_i} r_m(x-y)}{(c^2+\|x-y\|^2)^{m+1}} + \tfrac{4m(m+1)(x_i-y_i)^2 k(x,y) r_m(x-y)}{(c^2+\|x-y\|^2)^{m+2}} \right] \\ &= k(x, y) \tfrac{r_{m+2}(x-y)}{(c^2+\|x-y\|^2)^{m+2}} \end{aligned} \tag{15}$$

where $r_{m+2}(x - y)$ has been implicitly defined. Since $\partial_{x_i}(x_i - y_i)^s = s(x_i - y_i)^{s-1}$, it follows that the terms $\partial_{x_i} r_m(x - y)$, $\partial_{y_i} r_m(x - y)$ and $\partial_{x_i} \partial_{y_i} r_m(x - y)$ appearing in (15) are polynomials in $x - y$ of maximal degree $2m$. Thus, from (15), $r_{m+2}(x - y)$ is a polynomial of maximal degree $2(m + 2)$, showing that the set $\mathcal{F}$ is closed under the action of $\partial_{x_i} \partial_{y_i}$.

Since the differential operator $\partial_{x_1}^{\alpha_1} \dots \partial_{x_d}^{\alpha_d}$ is obtained by repeated application of operators of the form $\partial_{x_i} \partial_{y_i}$, and since it is clear that all elements of $\mathcal{F}$ are bounded on $\mathbb{R}^d \times \mathbb{R}^d$, the claim is established. □

Now we can prove Theorem 2:

Proof of Theorem 2. First note that our preconditions are a superset of those required for Theorem 1. Thus the conclusion of Theorem 1 holds; namely, if $\mathcal{D}_\mathrm{S}(P, (T_n)_\# Q) \to 0$ then $(T_n)_\# Q \Rightarrow P$. From (11) we have that $\mathcal{D}_{\mathrm{S}}(P,(T_n)_{\#}Q)$ and $\inf_{T\in\mathcal{T}_n}\mathcal{D}_{\mathrm{S}}(P,T_{\#}Q)$ agree in the $n\to\infty$ limit. Thus it is sufficient to show that $\inf_{T\in\mathcal{T}_n}\mathcal{D}_{\mathrm{S}}(P,T_{\#}Q)\to 0$ in the $n\to\infty$ limit.

Second, note that our preconditions are also a superset of those required for Proposition 5. Indeed, from Lemma 4 the inverse multi-quadric kernel in (3) is infinitely differentiable with derivatives of all orders bounded. Thus it is sufficient to show that $\inf_{T\in\mathcal{T}_n}W_1(P,T_{\#}Q)\to 0$ in the $n\to\infty$ limit.

From Assumptions 1 and 2 and Proposition 2, there exists a map $T\in\prod_{i=1}^d L^2(Q)$ with $T_{\#}Q=P$. From Assumption 3, there is a set $\mathfrak{T}$ such that $T\in\mathfrak{T}\subseteq\prod_{i=1}^d L^2(Q)$ and the set $\mathcal{T}_\infty$ is dense in $\mathfrak{T}$. Thus there exists a sequence $(S_n)_{n\in\mathbb{N}}\subset\mathcal{T}_\infty$ with $S_n\to T$ in $\prod_{i=1}^d L^2(Q)$.

For each $n\in\mathbb{N}$, let $m_n\in\mathbb{N}$ denote the smallest integer m for which $S_n\in\mathcal{T}_m$, which is well-defined since $S_n\in\mathcal{T}_\infty=\cup_{i\in\mathbb{N}}\mathcal{T}_i$. Let $M_n:=\max\{m_1,\dots,m_n,n\}$, so that M_n is a non-decreasing sequence with $M_n\to\infty$ in the $n\to\infty$ limit. Note that, since $\mathcal{T}_i\subseteq\mathcal{T}_j$ for all $i\le j$, we have $S_n\in\mathcal{T}_{M_n}$.

Thus from Lemma 3 we conclude that

$$0\le\inf_{T\in\mathcal{T}_{M_n}}W_1(P,T_{\#}Q)\le W_1(P,(S_n)_{\#}Q)\le\|T-S_n\|_{\prod_{i=1}^d L^2(Q)}\to 0 \tag{16}$$

in the $n\to\infty$ limit. Again, since $\mathcal{T}_i\subseteq\mathcal{T}_j$ for $i\le j$, the sequence $n\mapsto\inf_{T\in\mathcal{T}_n}W_1(P,T_{\#}Q)$ is non-increasing and, from (16), it has a subsequence that converges to 0. It follows that $\lim_{n\to\infty}\inf_{T\in\mathcal{T}_n}W_1(P,T_{\#}Q)=0$, as required.

□

A.4 Proof of Proposition 3

Recall the *rectified linear unit* activation function $\sigma(x)=\max(0,x)$, which we consider to be applied componentwise when $x\in\mathbb{R}^d$.

Definition 4 (Deep ReLU Neural Network). *A* deep ReLU neural network *with l* hidden layers *from $\mathbb{R}^p$ to $\mathbb{R}^d$ is a function $f:\mathbb{R}^p\to\mathbb{R}^d$ of the form*

$$f=F_{l+1}\circ\sigma\circ F_l\circ\cdots\circ F_2\circ\sigma\circ F_1$$

where $F_i:\mathbb{R}^{w_{i-1}}\to\mathbb{R}^{w_i}$, $i=1,\dots,l$, is an affine transformation, $F_{l+1}:\mathbb{R}^{w_l}\to\mathbb{R}^{w_{l+1}}$ is a linear transformation, $w_0=p$ is the input dimension*, $w_{l+1}=d$ is the* output dimension*, and $w_i\in\mathbb{N}$, $i=1,\dots,l$, is the* width *of the ith* hidden layer*. The set of all deep ReLU neural networks with l* hidden layers *from $\mathbb{R}^p$ to $\mathbb{R}^d$ with maximum width $\max\{w_1,\dots,w_l\}\le n$ is denoted $\mathcal{R}_{l,n}(\mathbb{R}^p\to\mathbb{R}^d)$ and we let $\mathcal{R}_{l,\infty}(\mathbb{R}^p\to\mathbb{R}^d):=\lim_{n\to\infty}\mathcal{R}_{l,n}(\mathbb{R}^p\to\mathbb{R}^d)$.*

The following, essentially trivial observation will be useful:

Proposition 6. $\prod_{i=1}^d\mathcal{R}_{l,\infty}(\mathbb{R}^p\to\mathbb{R})\subset\mathcal{R}_{l,\infty}(\mathbb{R}^p\to\mathbb{R}^d)$.

Proof. For fixed $n\in\mathbb{N}$, there is a canonical injection from $\prod_{i=1}^d\mathcal{R}_{l,n}(\mathbb{R}^p\to\mathbb{R})$ into $\mathcal{R}_{l,nd}(\mathbb{R}^p\to\mathbb{R}^d)$ that concatenates the d neural networks width-wise, to form a single neural network with width nd. Since every element of $\prod_{i=1}^d\mathcal{R}_{l,\infty}(\mathbb{R}^p\to\mathbb{R})$ belongs to $\prod_{i=1}^d\mathcal{R}_{l,n}(\mathbb{R}^p\to\mathbb{R})$ for a sufficiently large $n\in\mathbb{N}$, the claim is established. □

Here we introduce the shorthand $L^2(\mathbb{R}^p)$ for $L^2(\lambda_{\mathbb{R}^p})$ where $\lambda_{\mathbb{R}^p}$ is the Lebesgue measure on $\mathbb{R}^p$. The following result on the approximation properties of deep ReLU neural networks, which derives from the fact that the set of continuous piecewise linear functions $f:\mathbb{R}\to\mathbb{R}$ is dense in $L^2(\mathbb{R})$, will be required.

Proposition 7. *For every function $f\in\prod_{i=1}^d L^2(\mathbb{R}^p)$ and every $\epsilon>0$, there exists a function $g\in\mathcal{R}_{l,\infty}(\mathbb{R}^p\to\mathbb{R}^d)$ such that $\|f-g\|_{\prod_{i=1}^d L^2(\mathbb{R}^p)}<\epsilon$, where $l=\lceil\log_2(p+1)\rceil$.*

Proof. For $d=1$, this result is a special case of Theorem 2.3 in Arora et al. (2018), which derives from the fact that continuous piecewise linear functions are dense in $L^2(\mathbb{R})$.

For general d, we observe that for each component $f_i \in L^2(\mathbb{R}^p)$ we can find a function $g_i \in \mathcal{R}_{l,\infty}(\mathbb{R}^p \to \mathbb{R})$ with $\|f_i - g_i\|_{L^2(\mathbb{R}^p)} < \epsilon/\sqrt{d}$. Then, letting $g = (g_1, \dots, g_d) : \mathbb{R}^p \to \mathbb{R}^d$, we have that $g \in \prod_{i=1}^d \mathcal{R}_{l,\infty}(\mathbb{R}^p \to \mathbb{R})$ and

$$\|f - g\|_{\prod_{i=1}^d L^2(\mathbb{R}^p)} = \sqrt{\sum_{i=1}^d \|f_i - g_i\|^2_{L^2(\mathbb{R}^p)}} < \sqrt{\sum_{i=1}^d \frac{\epsilon^2}{d}} = \epsilon.$$

Finally, we note from Proposition 6 that $g \in \prod_{i=1}^d \mathcal{R}_{l,\infty}(\mathbb{R}^p \to \mathbb{R}) \subset \mathcal{R}_{l,\infty}(\mathbb{R}^p \to \mathbb{R}^d)$. □

Now we present the proof of Proposition 3:

Proof of Proposition 3. From Assumptions 1 and 2 and Proposition 2 there exists $T \in \prod_{i=1}^d L^2(Q)$ such that $T_\# Q = P$. Let $\mathfrak{T} = \prod_{i=1}^d L^2(Q)$, so that $T \in \mathfrak{T}$ is satisfied.

From the statement of Proposition 3 we have $\mathcal{T}_n := \mathcal{R}_{l,n}(\mathbb{R}^p \to \mathbb{R}^d)$ with $l = \lceil \log_2(p+1) \rceil$. From Definition 4 it is therefore clear that $\mathcal{T}_n \subseteq \mathcal{T}_m$ whenever $n \le m$ and that $\mathcal{T}_\infty = \mathcal{R}_{l,\infty}(\mathbb{R}^p \to \mathbb{R}^d)$.

Thus all parts of Assumption 3 have been verified except the part that requires $\mathcal{T}_\infty$ to be dense in $\mathfrak{T}$; i.e. that the set $\mathcal{R}_{l,\infty}(\mathbb{R}^p \to \mathbb{R}^d)$ is dense in the Hilbert space $\prod_{i=1}^d L^2(Q)$. To establish this last part, we will make use of Proposition 7:

Let $T \in \prod_{i=1}^d L^2(Q)$ and $\epsilon > 0$. From the definition of $L^2(Q)$, there exists $c \ge 0$ such that, for each of the coordinates $i \in \{1, \dots, d\}$,

$$\int_{\mathbb{R}^p \setminus [-c,c]^p} T_i(x)^2 \, \mathrm{d}Q(x) < \frac{\epsilon}{4d}.$$

Let

$$f_i(x) := \begin{cases} T_i(x), & x \in [-c,c]^p, \\ 0, & x \in \mathbb{R}^p \setminus [-c,c]^p. \end{cases}$$

Our assumption that Q admits a positive and continuous density q on $\mathbb{R}^p$ ensures that $f \in \prod_{i=1}^d L^2(\mathbb{R}^p)$, since

$$\begin{aligned} \|f\|^2_{\prod_{i=1}^d L^2(\mathbb{R}^p)} = \sum_{i=1}^d \int_{[-c,c]^p} T_i(x)^2 \mathrm{d}x = \sum_{i=1}^d \int_{[-c,c]^p} \frac{T_i(x)^2}{q(x)} \mathrm{d}Q(x) &\le \left[\sup_{x \in [-c,c]^d} \frac{1}{q(x)} \right] \sum_{i=1}^d \int_{[-c,c]^p} T_i(x)^2 \mathrm{d}Q(x) \\ &\le \left[\sup_{x \in [-c,c]^p} \frac{1}{q(x)} \right] \sum_{i=1}^d \|T_i\|^2_{L^2(Q)} \\ &= \left[\sup_{x \in [-c,c]^p} \frac{1}{q(x)} \right] \|T\|^2_{\prod_{i=1}^d L^2(Q)}, \qquad (17) \end{aligned}$$

where the supremum in (17) is finite, since q^{-1} is well-defined and continuous on the compact set $[-c,c]^p$. Let also $q_{\max} := \sup_{x \in \mathbb{R}^p} q(x)$, which is well-defined since we assumed q to be continuous and bounded on $\mathbb{R}^p$. Then, since $f \in \prod_{i=1}^d L^2(\mathbb{R}^p)$, we may evoke Proposition 7 to find a function $g \in \mathcal{R}_{l,\infty}(\mathbb{R}^p \to \mathbb{R}^d)$ such that $\|f - g\|^2_{\prod_{i=1}^d L^2(\mathbb{R}^p)} < \epsilon/(4q_{\max})$. It remains to check that g approximates T in $\prod_{i=1}^d L^2(Q)$. To this end, we can use the triangle inequality in $\prod_{i=1}^d L^2(Q)$ and the fact that $(a+b)^2 \le 2(a^2 + b^2)$ to see that

$$\begin{aligned} \|T - g\|^2_{\prod_{i=1}^d L^2(Q)} &\le 2\|T - f\|^2_{\prod_{i=1}^d L^2(Q)} + 2\|f - g\|^2_{\prod_{i=1}^d L^2(Q)} \\ &= 2 \sum_{i=1}^d \int_{\mathbb{R}^p \setminus [-c,c]^p} T_i(x)^2 \mathrm{d}Q(x) + 2 \sum_{i=1}^d \int (f_i(x) - g_i(x))^2 q(x) \mathrm{d}x \\ &\le 2 \sum_{i=1}^d \int_{\mathbb{R}^p \setminus [-c,c]^p} T_i(x)^2 \mathrm{d}Q(x) + 2q_{\max} \sum_{i=1}^d \int (f_i(x) - g_i(x))^2 \mathrm{d}x < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon. \end{aligned}$$

Since $\epsilon > 0$ was arbitrary, this argument shows that the set $\mathcal{R}_{l,\infty}(\mathbb{R}^p \to \mathbb{R}^d)$ is dense in $\prod_{i=1}^d L^2(Q)$, as required. □

B Computational Details

This section provides full details for the experiments presented in Section 4. Note that we never attempt to verify that our target distributions are distantly dissipative (Definition 1).

B.1 Performance Metric

To estimate the Wasserstein-1 distance between the target distribution and approximations we computed the *earth mover distance* (EMD) between two uniformly weighted empirical measures, each formed from 10^4 samples from their respective distributions and with the Euclidean distance between the samples used to construct the cost matrix. The EMD was computed using an implementation in the Python Optimal Transport (`POT`) package (Flamary and Courty, 2017). For the target distribution, independent samples were used in the synthetic test bed in Section 4.1 and thinned samples from a long HMC chain were used for the real examples in Sections 4.2 and 4.3. For the approximate distribution, independent samples from $T^{\theta}_{\#}Q$ were used.

B.2 Details of the Synthetic Test Bed

To assess the proposed methods, we considered the following bivariate densities

$$\begin{aligned} p_1(x,y) &:= \mathcal{N}(x;0,\eta_1^2)\mathcal{N}(y;\sin(ax),\eta_2^2), \\ p_2(x,y) &:= \mathcal{N}(x;0,\sigma_1^2)\mathcal{N}(y;bx^2,\sigma_2^2), \\ p_3(x,y) &:= \tfrac{1}{n}\textstyle\sum_{i=1}^{n}\mathcal{N}(x,y;\mu_i,\sigma^2 I_2), \end{aligned}$$

where $\mathcal{N}(x;\mu,\sigma^2)$ is the univariate Gaussian density with mean μ and variance σ^2, and $\mathcal{N}(x,y;\mu,K)$ is the bivariate Gaussian density with mean vector μ and covariance matrix K. The parameter choices for the sinusoidal experiment p_1 were $\eta_1^2 = 1.3^2, \eta_2^2 = 0.001^2$ and $a = 1.2$. The parameter choices for the banana experiment p_2 were $\sigma_1^2 = 1, \sigma_2^2 = 0.1^2$ and $b = 0.5$. The parameter choices for the multi-modal experiment p_3 were $n = 4, \mu_1 = (1,1), \mu_2 = (1,-1), \mu_3 = (-1,-1), \mu_4 = (-1,1)$ and $\sigma^2 = 0.2^2$. The target densities can be seen in Figure 3.

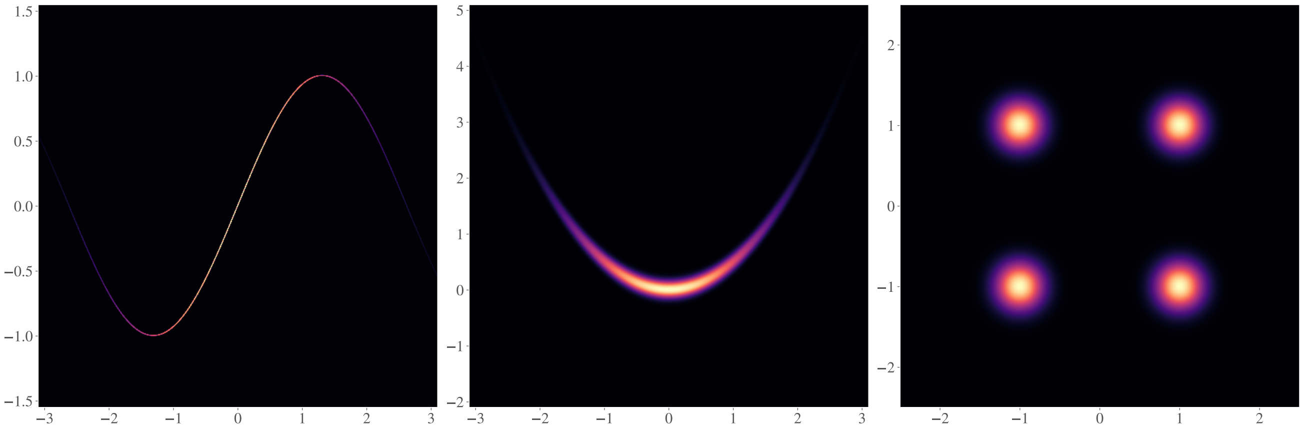


Figure 3: Contour plots of the three synthetic test densities p_1, p_2 and p_3 from left to right used in Section 4.1.

Computational costs for each method in terms of number of target evaluations and CPU wall-clock time against performance are shown, respectively, in Figure 4 and Figure 5. From Figure 4, there is no clear sense in which KSD or KLD out-performs the other across the different synthetic tests; this is in line with the conclusion of Section 4.1. For CPU wall-clock time in Figure 5, KSD-based measure transport is approximately three to five times slower than its KLD counterpart. However, note that our implementation of KSD is not production code and further performance gains can certainly be achieved.

We now discuss the implementations details of all the methods used in Section 4.1. In all our measure transport implementations, unless specified otherwise, we used existing implementations in Pyro (Bingham et al., 2018). Furthermore, the reference measure used for synthetic tests was the standard Gaussian on $\mathbb{R}^p$.

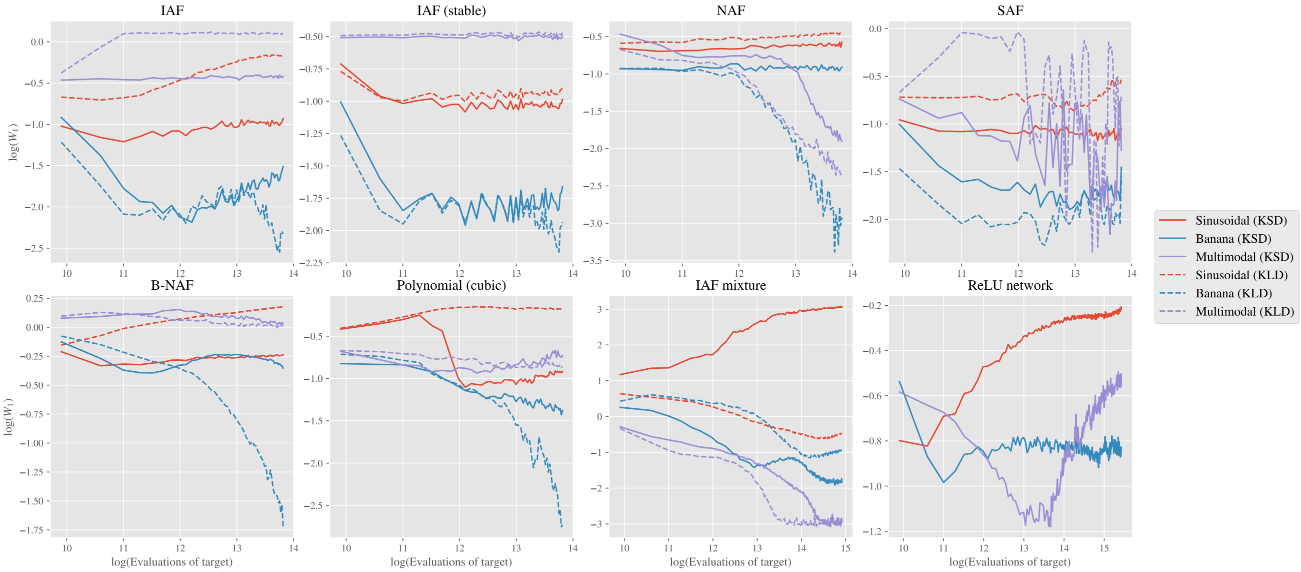


Figure 4: Wasserstein-1 metric, W_1, as a function of the total number of evaluations of either p or its gradient, for each synthetic test experiment.

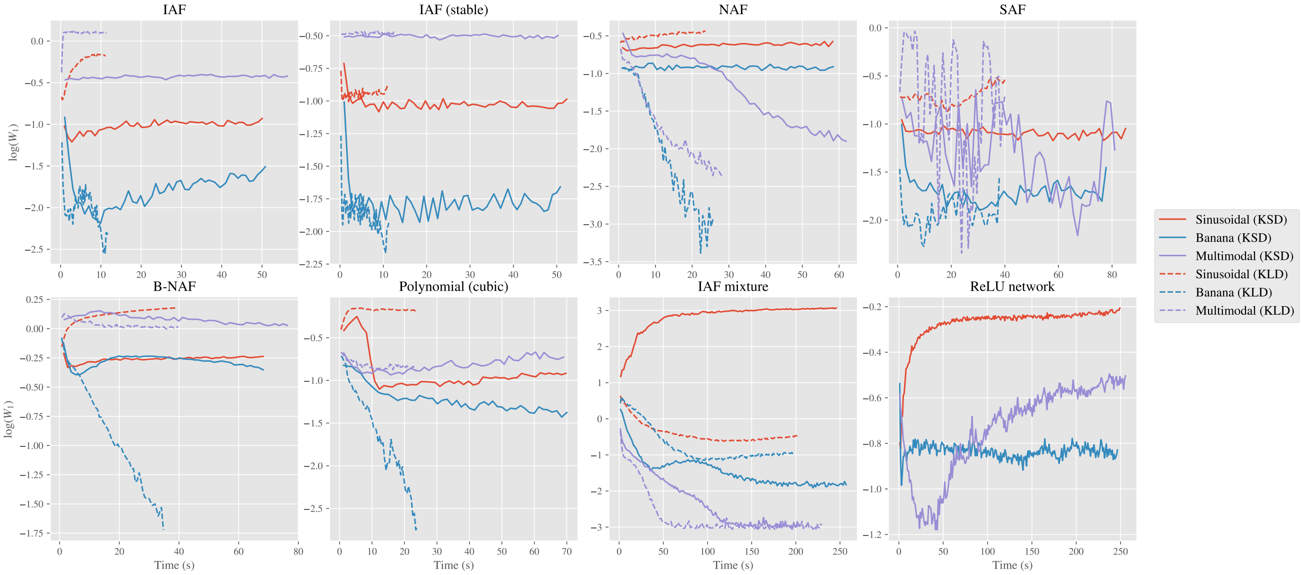


Figure 5: Wasserstein-1 metric, W_1, as a function of the CPU wall-clock time for each synthetic test experiment.

Hamiltonian Monte Carlo: We used an adaptive HMC algorithm in which the integrator step size was automatically adjusted using a dual-averaging algorithm in a warm-up phase to give an average acceptance statistic of 0.8 (Betancourt et al., 2014) and the number of integrator steps per transition was set dynamically by expanding the trajectory until a termination criterion was met (Hoffman and Gelman, 2014; Betancourt, 2017). We used the HMC implementations in the Python package (Graham, 2020), with the dual-averaging adaptation algorithm settings following the defaults used in Stan (Carpenter et al., 2017). Only the post-warm-up samples were included in estimates of the discrepancies and density plots.

In the following, the autoregressive neural networks that we specify are the Masked Autoencoders for Density Estimation (MADE) of Germain et al. (2015); the only difference being, that there is no sigmoidal non-linearity applied to the output layer.

Inverse Autoregressive Flow (IAF): Recall from Section 2.5, that an autoregressive flow $T : \mathbb{R}^d \to \mathbb{R}^d$ is of the form $T_i(x) = \tau(c_i(x_1, \ldots, x_{i-1}), x_i)$, where $T = (T_1, \ldots, T_d)$ and $x = (x_1, \ldots, x_d)$. The IAF of Kingma et al. (2016) takes c_i to be the ith output of an autoregressive neural network and τ to be an affine transform of the form

$$\tau(c_i(x_1, \ldots, x_{i-1}), x_i) = \mu_i + \exp(s_i)x_i,$$

where $\mu_i \in \mathbb{R}$ and $s_i \in \mathbb{R}$ are outputs from $c_i(x_1, \ldots, x_{i-1})$. Note that the coefficient of x_i is forced to be positive, this ensures the resulting transport map is monotonic.

For each synthetic test problem, we used a single IAF where the dimensionality of the hidden units in the single hidden layer of the underlying autoregressive neural network was 40. The underlying autoregressive neural network used the ReLU activation function. The IAF was initialised using the same default random initialisation in both the KSD and KLD experiments. $10,000$ iterations of Adam were used, with learning rate 0.001.

Stable Inverse Autoregressive Flow (IAF stable): Closely related to an IAF, a stable IAF was introduced in Kingma et al. (2016) in order to improve numerical stability. The only difference being that the τ is of the form

$$\tau(c_i(x_1, \ldots, x_{i-1}), x_i) = \text{sigmoid}(s_i)x_i + (1 - \text{sigmoid}(s_i))\mu_i,$$

where $\mu_i \in \mathbb{R}$ and $s_i \in \mathbb{R}$ are outputs from $c_i(x_1, \ldots, x_{i-1})$ and

$$\text{sigmoid}(x) = \frac{e^x}{1 + e^x},$$

where, for $x \in \mathbb{R}^d$, we consider sigmoid to be applied component-wise. Since sigmoid is monotonic, the resulting transport map is again monotonic. The restriction $\text{sigmoid}(s_i) \in (0, 1)$ may limit the expressibility of the transport map compared to standard IAF, but this at the expense of increased numerical stability.

For each synthetic test problem, we used a single stable IAF where the dimensionality of the hidden units in the single hidden layer of the underlying autoregressive neural network was 40. The underlying autoregressive neural network used the ReLU activation function. The stable IAF was initialised using the same default random initialisation in both the KSD and KLD experiments. $10,000$ iterations of Adam were used, with learning rate 0.001.

Neural Autoregressive Flow (NAF): A NAF, introduced in Huang et al. (2018), is again an autoregressive flow which, compared to the preceding IAF and stable IAF, offers greater flexibility and a universality guarantee. The autoregressive conditioner c is again taken as an autoregressive neural network and $\tau : \mathbb{R} \to \mathbb{R}$ takes the form of a monotonic neural network whose weights and biases are the output of the conditioner c. Monotonicity of τ is guaranteed by using strictly positive weights and strictly monotonic activation functions.

The particular implementation of τ that we used in Pyro uses what is termed a *deep sigmoidal flow* (DSF) in Huang et al. (2018). A DSF is a single layer dense neural network with a sigmoidal activation function. Furthermore, in order to increase the effective range of τ, an inverse sigmoid function is taken on the output layer. However, this inverse sigmoid function has domain $(0, 1)$ and thus the weights and biases of the output layer must be constrained such that this composition can be well defined. In a DSF, this is achieved by having no bias term on the output layer and constraining the output layer's weights, w_{ij}, to satisfy $\sum_i w_{ij} = 1$. Thus the output term of a DSF is a convex combination of the output of the hidden layer. The overall transformation of a DSF is then of the form

$$f_{\text{DSF}} = \text{sigmoid}^{-1} \circ C \circ \text{sigmoid} \circ F,$$

where $F : \mathbb{R} \to \mathbb{R}^k$ is an affine transformation and $C : \mathbb{R}^k \to \mathbb{R}$ is a convex combination. $10,000$ iterations of Adam were used, with learning rate 0.001.

Following Huang et al. (2018), for each synthetic test problem, we used a single DSF style NAF, where the dimensionality of the hidden sigmoid units in each DSF was 16 and the dimensionality of the hidden units in the single hidden layer of the underlying autoregressive neural network was 40. The underlying autoregressive neural network used the ReLU activation function. Note that the dimensionality of ith output c_i of this autoregressive neural network is 48, due to the 2×16 weight terms and the 16 bias terms in each DSF. The NAF was initialised using the same default random initialisation in both the KSD and KLD experiments.

Spline Autoregressive Flow (SAF): A SAF, developed in Durkan et al. (2019) and Dolatabadi et al. (2020), is again an autoregressive flow that takes $\tau : \mathbb{R} \to \mathbb{R}$ as a piecewise monotonic rational polynomial function (a spline) on an interval $[-a, a]$ and the identity otherwise. A rational polynomial function is the ratio of two polynomials. In Durkan et al. (2019), the polynomial was taken as quadratic polynomial and in Dolatabadi et al. (2020), the polynomial was taken as a linear polynomial. The parameters controlling each rational polynomial function are computed from the output of an autoregressive neural network. We also note that the spline transform was first implemented in the context of coupling flows, rather than autoregressive flows.

For each synthetic test problem, we used a single SAF based on rational linear splines with 8 pieces defined on the interval $[-3, 3]$. The underlying autoregressive neural network had two hidden layers, each of dimension 20. The SAF was initialised using the same default random initialisation in both the KSD and KLD experiments. $10,000$ iterations of Adam were used, with learning rate 0.001.

Block Neural Autoregressive Flow (B-NAF): A B-NAF, introduced in Cao et al. (2019), is similar in spirit to the NAF. It is an autoregressive flow, where τ is a neural network. The difference is now that the weights and biases of τ are not the output of an autoregressive conditioner network c; instead, the parameters of the neural network are learned directly. In a B-NAF, the affine transformations $L : \mathbb{R}^{ad} \to \mathbb{R}^{bd}$, for $a, b \in \mathbb{Z}^+$, used at a given layer are always in a lower triangular block form

$$L(x) = \begin{pmatrix} u(B_{11}) & 0 & \dots & 0 \\ B_{21} & u(B_{22}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ B_{d1} & B_{d1} & \dots & u(B_{dd}) \end{pmatrix} x + \mu,$$

where $u : \mathbb{R} \to \mathbb{R}^+$, each $B_{ii} \in \mathbb{R}^{a \times b}$ and $\mu \in \mathbb{R}^{bd}$ is the freely parameterised bias term . The positivity-ensuring transform u enforces monotonicity. Bijectivity is further ensured by using bijective activation functions. Note that this particular form of affine transformation place restrictions on the structure of the neural network. For instance, the hidden dimensions must be a multiple of the input dimension d. Similarly to NAFs, B-NAFs also have a universality result.

For each synthetic test problem we used a single B-NAF with the tanh activation function and $u(x) = \exp(x)$. The B-NAF used had two hidden layers and was of the form

$$f_{BNAF} = L_3 \circ \tanh \circ L_2 \circ \tanh \circ L_1,$$

with lower triangular block affine transformations $L_1 : \mathbb{R}^2 \to \mathbb{R}^{16}$, $L_2 : \mathbb{R}^{16} \to \mathbb{R}^{16}$ and $L_3 : \mathbb{R}^{16} \to \mathbb{R}^2$. The B-NAF was initialised using the same default random initialisation in both the KSD and KLD experiments. $10,000$ iterations of Adam were used, with learning rate 0.001.

Polynomial (Cubic): Polynomials were first put forward as possible parametric transport maps in measure transport (Marzouk et al., 2016; Parno and Marzouk, 2018). In Marzouk et al. (2016), each component of a polynomial transport map $T : \mathbb{R}^d \to \mathbb{R}^d$, was parameterised as a linear basis expansion of multivariate polynomials $\phi_j : \mathbb{R}^d \to \mathbb{R}$. Each ϕ_j is further parameterised with respect to a vector of polynomial degrees $j = (j_1, \dots, j_d) \in \mathbb{N}^d$ as a product of d univariate polynomials of the form

$$\phi_j(x) = \prod_{k=1}^{d} \psi_{j_k}(x_i),$$

where each $\psi_{j_k}(x_i)$ is a univariate degree j_k polynomial. These ψ_{j_k} can come from orthogonal families of polynomials or simply be monomials. For instance, we could take the ψ_{j_k} to be orthogonal with respect to the reference measure of the transport map. The ith component of T can thus be written as

$$T_i(x) = \sum_{j \in \mathcal{J}_i} \lambda_{j,i} \phi_j(x),$$

where each $\lambda_{j,i} \in \mathbb{R}$. This is a flexible parameterisation that can enforce triangularity through the choices of the $\mathcal{J}_i$. For example, a natural choice to enforce triangularity would be to take $\mathcal{J}_i$ to consist of vectors

$j = (j_1, \dots, j_i, 0, \dots, 0)$ such that $\sum_k j_k \leq p$. The first constraint enforces triangularity and the second restraint ensures that the total degree of the resulting polynomials would be no greater than a given $p \in \mathbb{N}$. In higher dimensions, this may not be practical since the number of parameters grows quickly as the dimension increases. Thus other constraints on $\mathcal{J}_i$ were put forward, such as removing mixed terms in the basis.

An issue with this approach is that the resulting maps are not monotonic for all values of the coefficients $\lambda_{j,i}$. In Marzouk et al. (2016) and Parno and Marzouk (2018), monotonicity was constrained locally at a given set of samples $\{u_i\}_{i=1}^n$ from the reference distribution. Due to the triangular nature of the transport map, this effectively results in a finite set of linear constraints of the form $\partial_{x_i} T_i(u_k) > 0$ for $i = 1, \dots d$ and $k = 1, \dots, n$. In our implementation of KLD-based polynomial transport for the synthetic test bed, we found that this approach was not necessary since, in each case, the resulting Jacobian always had a positive determinant.

In our synthetic experiments, we used the the natural choice of the $\mathcal{J}_i$ that enforces triangularity that we previously discussed with $p = 3$ and took the ψ_i as simple monomials. The overall transport map was thus a multivariate cubic polynomial of the form

$$\begin{pmatrix} T_1(x_1) \\ T_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^3 c_i x_1^i \\ \sum_{i=0}^3 \sum_{j=0}^{3-i} c_{ij} x_1^i x_2^j \end{pmatrix}.$$

The polynomial transport map was initialised to the identity in all synthetic experiments. $10,000$ iterations of Adam were used, with learning rate 0.001.

IAF mixture: A mixture of transport maps is a distribution of the form

$$\sum_{i=1}^n w_i T_\#^{(i)} Q_i,$$

where the $T^{(i)}$ are each a transport map of a given form, the Q_i are possibly distinct reference distributions and the mixing weights $w_i \geq 0$ satisfy $\sum_i w_i = 1$. This is a very flexible extension to using just a single transport map. Furthermore, in principle, both the number of mixing components n and the mixing weights w_i could be learnt. For example, the weights w_i could be the output of a neural network with softmax applied to the output layer[4], as was done in Pires and Figueiredo (2020).

For simplicity, in our synthetic experiments, we *a priori* set $n = 4$ and further set each $w_i = 1/4$. We took each $T^{(i)}$ as a single IAF, where the dimensionality of the hidden units in the single hidden layer was 8. Refer to our discussion of an IAF in Appendix B.2 or Kingma et al. (2016) for full details of an IAF. The Q_i were initialised as Gaussians with means $(-2, 2), (-2, -2), (2, -2), (2, 2)$ respectively and each with identity covariance matrix. $30,000$ iterations of Adam were used, with learning rate 0.001.

ReLU network: Refer to Definition 4 for the definition of a deep ReLU network.

For our synthetic experiments, we implemented a deep ReLU network for each synthetic test problem. When using KSD, the transport map need not be a diffeomorphism and so, to illustrate this flexibility, the input dimension of the ReLU network for each experiment was taken as 4 (while the dimension of the target was 2). For each problem, the ReLU network $f_{ReLU} : \mathbb{R}^4 \to \mathbb{R}^2$ had two hidden layers and was of the form

$$f_{ReLU} = F_3 \circ \sigma \circ F_2 \circ \sigma \circ F_1,$$

where $F_1 : \mathbb{R}^4 \to \mathbb{R}^{20}$, $F_2 : \mathbb{R}^{20} \to \mathbb{R}^{20}$ and $F_3 : \mathbb{R}^{20} \to \mathbb{R}^2$ are affine transformations and σ is the ReLU non-linearity, defined in Appendix A.4. Using the default random initialisation of these ReLU networks nearly always resulted in bad output. So, for each synthetic experiment, the ReLU network was pretrained for $10,000$ iterations of KSD-based measure transport using Adam with learning rate 0.001, in order to approximate the reference distribution $\mathcal{N}((0,0), I_2)$. That is, we pretrained the ReLU network in order to initialise it close to $T_\# Q \approx \mathcal{N}((0,0), I_2)$. After pretraining, $50,000$ further iterations of Adam were used for each synthetic test problem, with learning rate 0.001.

[4] The ith component of the softmax function is of the form $\text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$.

B.3 Details of the Biochemical Oxygen Model Experiment

Derivation of the Posterior: Following on from Section 4.2, recall that the two-dimensional biochemical oxygen demand model is of the form

$$B(t) = \alpha_1(1 - \exp(-\alpha_2 t)).$$

Due to the positivity constraints on α_1 and α_2, we perform inference on the log of the parameters and thus consider the model

$$B(t; \theta_1, \theta_2) = e^{\theta_1}(1 - \exp(-e^{\theta_2} t)).$$

Synthetic data $y = (y_i)_{i=1}^6$ were generated at times $t = 0, 1, 2, 3, 4, 5$ with the parameter values $\theta_1 = \log(1)$ and $\theta_2 = \log(0.1)$, with observations corrupted by independent mean 0 Gaussian errors with variance $\sigma^2 = 0.05^2$. See Figure 6 for a plot of $B(t; \theta_1, \theta_2)$ with these given parameter values alongside our generated synthetic data.

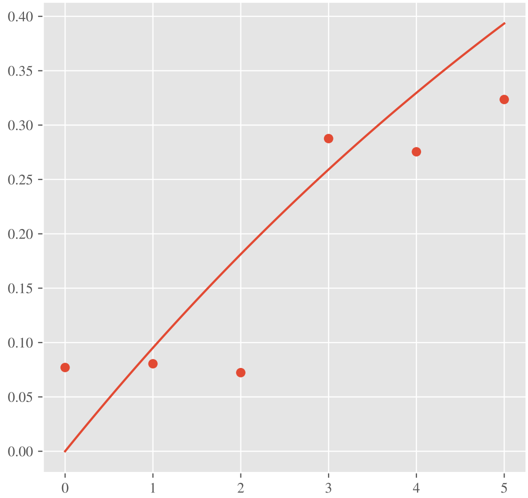


Figure 6: Plot of $B(t; \log(1), \log(0.1))$ with the corresponding synthetic data at $t = 0, 1, 2, 3, 4, 5$.

The likelihood is thus of the form

$$p(y \,|\, \theta_1, \theta_2) = \prod_{i=1}^{6} \mathcal{N}(y_i; B(t_i; \theta_1, \theta_2), \sigma^2).$$

The prior specified for $\theta = (\theta_1, \theta_2)$ was $\theta \sim \mathcal{N}((0, 0), I_2)$. The resulting posterior density is thus of the form

$$p(\theta_1, \theta_2 \,|\, y) \propto \mathcal{N}(\theta_1, \theta_2; (0, 0), I_2) \prod_{i=1}^{6} \mathcal{N}(y_i; B(t_i; \theta_1, \theta_2), \sigma^2).$$

Methodology: Our choice of parametric transport map was a Block Neural Autoregressive Flow (B-NAF) of Cao et al. (2019). We used the same B-NAF as the one used in Section 4.1, a B-NAF with two hidden layers of the form

$$f_{BNAF} = L_3 \circ \tanh \circ L_2 \circ \tanh \circ L_1,$$

with lower triangular block affine transformations $L_1 : \mathbb{R}^2 \to \mathbb{R}^{16}$, $L_2 : \mathbb{R}^{16} \to \mathbb{R}^{16}$ and $L_3 : \mathbb{R}^{16} \to \mathbb{R}^2$. Refer to Appendix B.2 or to Cao et al. (2019) for a full description of a B-NAF. The lengthscale used for KSD was $\ell = 0.1$. We again used the Adam optimiser, with default learning rate 0.001 with 30,000 iterations for each method.

Results: See Figure 2 for samples obtained from each resulting transport map. The KSD-based method obtained a Wasserstein-1 distance of 0.069 and the KLD-based method obtained a Wasserstein-1 distance of 0.015. Refer to Appendix B.1 for details on how this was calculated. Figure 7 plots $B(t; \theta_1, \theta_2)$ using samples from the prior and the approximate posterior using KSD-based measure transport.

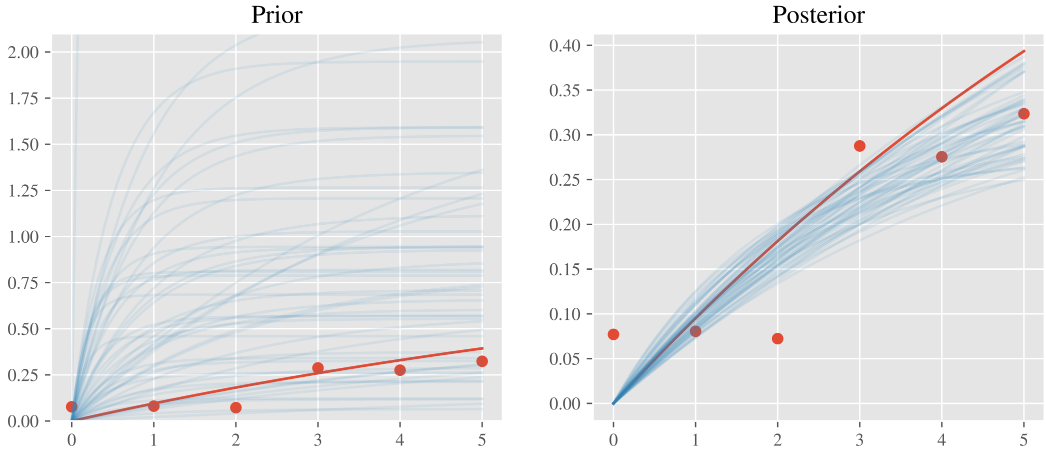


Figure 7: Plot of $B(t;\theta_1,\theta_2)$ using 50 samples for θ from (left) the prior and (right) the posterior, as approximated using KSD-based measure transport. The red line is $B(t;\log(1),\log(0.1))$.

B.4 Details of the Generalised Lotka–Volterra Model Experiment

Prior Specification: Recall that, from Section 4.3, the generalised Lotka–Volterra model we considered was of the form

$$\frac{\mathrm{d}p}{\mathrm{d}t}(t) = rp(t)\left(1-\frac{p(t)}{k}\right) - s\frac{p(t)q(t)}{a+p(t)}$$
$$\frac{\mathrm{d}q}{\mathrm{d}t}(t) = u\frac{p(t)q(t)}{a+p(t)} - vq(t),$$

with parameters $p_0, q_0, r, k, s, a, u, v > 0$. These parameters are physical quantities, see Rockwood (2015) for their full meaning. Due to the positivity constraints on these parameter values, similar to our biochemical oxygen demand experiment Section 4.2, we again perform inference on the log of the parameters. We thus consider the model

$$\frac{\mathrm{d}p}{\mathrm{d}t}(t) = e^{r'}p(t)\left(1-\frac{p(t)}{e^{k'}}\right) - e^{s'}\frac{p(t)q(t)}{e^{a'}+p(t)}$$
$$\frac{\mathrm{d}q}{\mathrm{d}t}(t) = e^{u'}\frac{p(t)q(t)}{e^{a'}+p(t)} - e^{v'}q(t),$$

where we perform inference on the parameter $\theta = (p_0', q_0', r', k', s', a', u', v') \in \mathbb{R}^8$.

After an investigation of the sensitivities of the solutions of the ODE with respect to the parameter values, we specified the following independent prior

$$p_0' \sim \mathcal{N}(\log 45, 0.2^2), q_0' \ \sim \mathcal{N}(\log 7, 0.3^2)$$
$$r' \sim \mathcal{N}(\log 0.5, 0.3^2), k' \sim \mathcal{N}(\log 80, 0.15^2)$$
$$s' \sim \mathcal{N}(\log 1.3, 0.2^2), a' \sim \mathcal{N}(\log 30, 0.1^2)$$
$$u' \sim \mathcal{N}(\log 0.6, 0.1^2), v' \sim \mathcal{N}(\log 0.28, 0.07^2).$$

Letting $\mu \in \mathbb{R}^8$ be the vector of these given mean values and K the diagonal matrix with these given variances on the diagonal, we have $\theta \sim \mathcal{N}(\mu, K)$. This specification results in log-normal priors on the exponentiated parameters.

Synthetic data $y = (p_i, q_i)_{i=1}^6$ were generated at times $t = 0, 10, 20, 30, 40, 50$ with parameter values $(p_0', q_0', r', k', s', a', u', v') = (\log 50, \log 5, \log 0.6, \log 90, \log 1.2, \log 25, \log 0.5, \log 0.3)$; these data were perturbed by independent mean 0 Gaussian errors with variance $\sigma^2 = 40$. See Figure 8 for a plot of the solution of the generalised Lotka-Volterra model with these given parameters alongside our generated synthetic data.

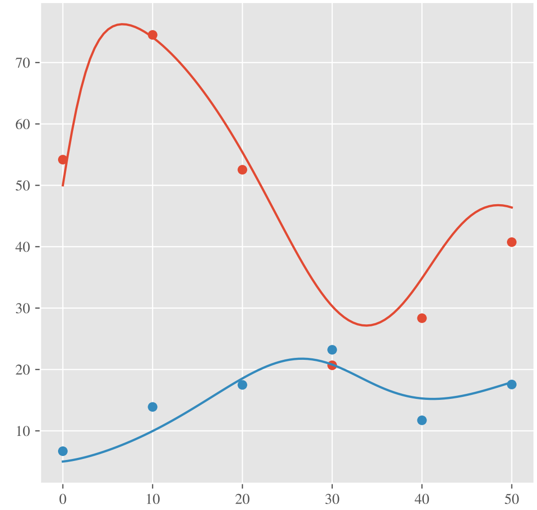


Figure 8: Solution of the generalised Lotka–Volterra model and the data that were provided at $t = 0, 10, 20, 30, 40, 50$.

The likelihood is thus of the form

$$p(y \mid \theta) = \prod_{i=1}^{6} \mathcal{N}(p_i, q_i; (p_\theta(t_i), q_\theta(t_i)), \sigma^2 I_2),$$

where $p_\theta(t), q_\theta(t)$ are the solutions of generalised Lotka-Volterra ODE (12) with given parameter θ. The resulting posterior density is thus of the form

$$p(\theta \mid y) \propto \mathcal{N}(\theta; \mu, K) \prod_{i=1}^{6} \mathcal{N}(p_i, q_i; (p_\theta(t_i), q_\theta(t_i)), \sigma^2 I_2).$$

Methodology: We used the `torchdiffeq` Python library (Chen et al., 2018) in order to numerically solve the Lotka–Volterra model and further utilised Pytorch's automatic differentiation capabilities to propagate gradients through the solver. In our implementation, we used the default Dormand-Prince Runge-Kutta method. The reference measure used was the prior. Our choice of parametric transport map was a B-NAF with $u(x) = \exp(x)$, of the form

$$f_{BNAF} = L_3 \circ \tanh \circ L_2 \circ \tanh \circ L_1,$$

with lower triangular block affine transformations $L_1 : \mathbb{R}^8 \to \mathbb{R}^{64}$, $L_2 : \mathbb{R}^{64} \to \mathbb{R}^{64}$ and $L_3 : \mathbb{R}^{64} \to \mathbb{R}^8$. Refer to Appendix B.2 or to Cao et al. (2019) for a full description of a B-NAF.

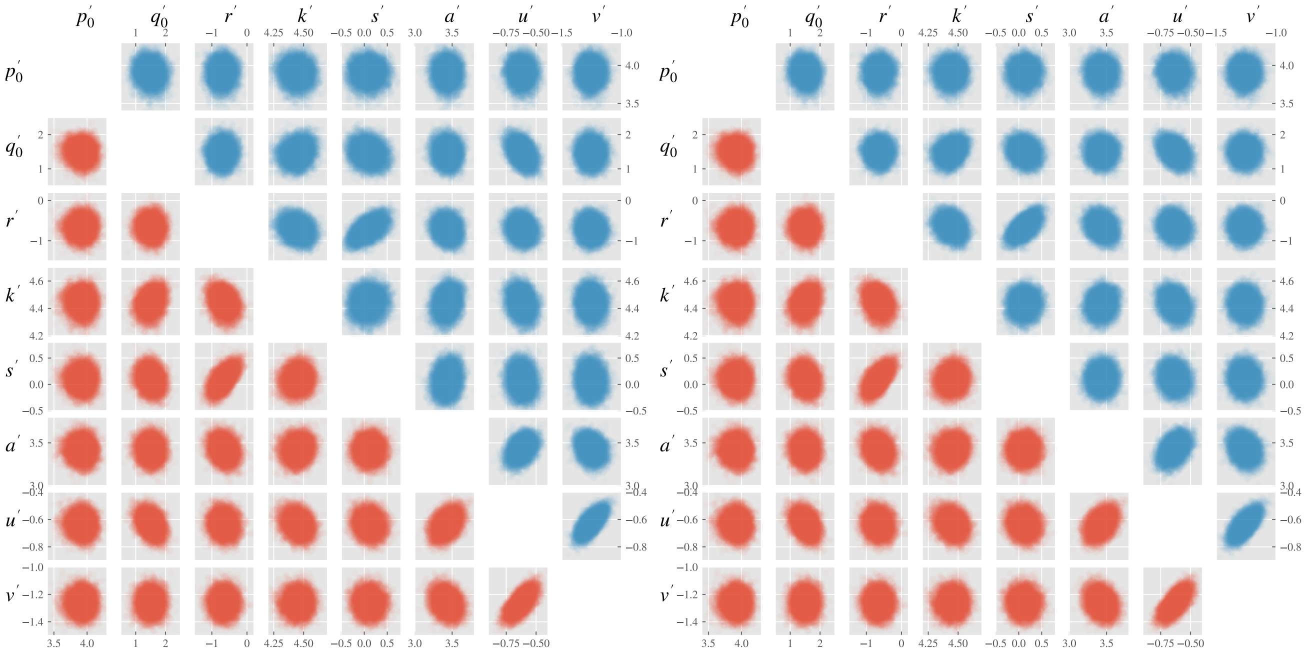
For both the KSD and KLD experiments, we used the same random initialisation of the B-NAF and pretrained on $10,000$ iterations of KLD-based measure transport on the prior (the reference measure), to ensure that the initial pushforward of samples through the B-NAF resulted in non-degenerate solutions of the Lotka–Volterra model.

The lengthscale used for KSD was $\ell = 0.1$ and we again used the Adam optimiser, with default learning rate 0.001 with $50,000$ iterations for each method.

Results: The KSD-based method obtained a Wasserstein-1 distance of 0.130, whereas the KLD-based method acheived a Wasserstein-1 distance of 0.110. Refer to Appendix B.1 for details on how this was calculated. The resulting approximating distributions for both the KSD and KLD methods are plotted in Figure 9.

C Further Investigations

Here we report a series of further investigations, that explore specific aspects of KSD-based measure transport in more detail.



(a) KSD

(b) KLD

Figure 9: Two-dimensional projections of samples from (a) the KSD-based method and (b) the KLD-based method. In both plots, the lower triangular subplots are the two-dimensional projections of samples from the gold-standard HMC and the upper triangular subplots are the two-dimensional projections of samples from the two methods.

C.1 Initialisation of Parameters in the Transport Map

Both KLD and KSD based measure transport can be sensitive to the initialisation of the parameters in a given transport map. This is, for instance, evidenced in Figure 16. In our experiments we generally used a random initialisation as specified by their implementations in Pyro (Bingham et al., 2018). In Appendix B, we specify for each experiment in the main paper what initialisation was used and whether we pretrained on the reference distribution Q.

A general remedy for poor initialisation is either to pretrain on the reference distribution. This was done in our applied examples in Section 4.2 and Section 4.3. Alternatively, in a Bayesian inference context one could pretrain on the prior distribution instead. The latter approach may be advantageous since we are guaranteed that the target's support is contained within the prior's support.

C.2 Investigating the Choice of Stochastic Optimisation Method

In all our experiments in Section 4, we used the Adam optimiser of Kingma and Ba (2015) with a fixed batch size of 100 and with a varying number of iterations. In this section we explore how the output of KSD-based measure transport, with a fixed number of iterations of stochastic optimisation, interacts with the batch size as well as the stochastic optimisation method used. We fixed the target distribution as the multimodal problem and considered only B-NAF as our transport map with standard Gaussian reference distribution. Results are shown in Table 2, where we report the Wasserstein-1 distance using 10^4 samples (see Appendix B.1 for more details). We pretrained the B-NAF on the reference distribution using KLD as our loss for 5000 iterations of Adam with learning rate $l = 0.001$, hence the discrepancy with the main results reported in Table 1.

From Table 2, the most consistent and best performing optimisation methods were Adam and RMSprop, where the smaller learning rate of $l = 0.001$ seemed to perform best. For stochastic gradient descent (SGD) and averaged stochastic gradient descent (ASGD), since the learning rate is non-adaptive, if the initial learning rate l is too large, the optimiser can fail to converge. This is evidenced by ASGD failing at a batch size of 25 with $l = 0.005$ in Table 2. For each of the optimisation methods detailed in Table 2, all parameters other than the

	Batch size			
Optimisation method	25	50	100	200
Adam ($l = 0.001$)	0.594	0.546	0.121	**0.0827**
Adam ($l = 0.01$)	0.663	0.768	0.726	**0.242**
Adagrad ($l = 0.001$)	0.630	0.620	0.617	**0.616**
Adagrad ($l = 0.01$)	0.612	0.608	0.606	0.602
RMSprop ($l = 0.001$)	0.590	0.504	0.144	**0.0856**
RMSprop ($l = 0.01$)	1.00	0.699	**0.0653**	0.125
SGD ($l = 0.001$)	0.902	**0.194**	0.402	0.433
SGD ($l = 0.005$)	1.24	0.265	**0.143**	0.241
ASGD ($l = 0.001$)	0.701	**0.34**	0.401	0.432
ASGD ($l = 0.005$)	N/A	**0.166**	0.478	0.391

Table 2: The W_1 distance to the multimodal target, varying the stochastic optimisation method. We used $10,000$ iterations for each experiment. The l value next to the name of each optimisation method was the learning rate used. Bold values indicate which of the batch sizes obtained the best Wasserstein-1 distance for each given optimisation method. N/A values indicate when the optimisation method failed.

learning rate were set to their default values as specified in Pytorch.

C.3 Investigating the Effect of Quasi-Monte Carlo Sampling

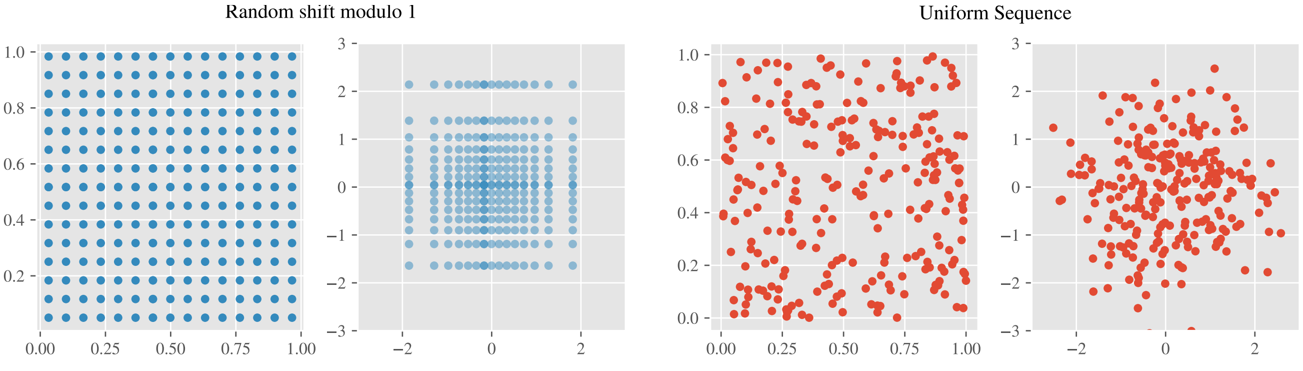
To reduce the variance of the Monte-Carlo based gradient estimators, it was put forward in Buchholz et al. (2018) and Wenzel et al. (2018), to instead use randomised Quasi-Monte Carlo (QMC) in constructing an unbiased estimator of the gradient. This is achieved by simply replacing the Monte-Carlo samples from the base distribution with samples from a (randomised) QMC sequence in a principled manner. This may be especially useful in our setting, where the variance of the U-statistic estimator of KSD is often quite large. In this section, we explore the replacement of the Monte-Carlo based U-statistic estimator in Proposition 1 with a QMC-based estimator empirically. We first, however, briefly outline the rudimentary idea. Refer to Buchholz et al. (2018) and Wenzel et al. (2018) for the full details.

A low-discrepancy sequence or a QMC sequence of a given length on $[0, 1]^d$, roughly speaking, allocates points such that the number of points in a given measurable subset of $[0, 1]^d$ is proportional to its volume. A prototypical example of a randomised QMC estimator is the *random shift modulo 1*, where the sequence is generated by first specifying a grid of values x_i over $[0, 1]^d$ and then sampling a $u \sim U([0, 1]^d)$, the resulting QMC sequence is the set of points $x_i + u \mod 1$. Using an appropriate measurable function $S : [0, 1]^d \to \mathbb{R}^d$, one can use QMC to integrate with respect to a given distribution P, as long as $S_{\#}U([0, 1]^d) = P$, by pushing forward the QMC sequence through S. Figure 10 plots a randomly shifted (modulo 1) grid in two-dimensions, along with its pushforward on to the standard Gaussian against a uniform sample.

Results are shown in Table 3, where we compare this prototypical QMC method with Monte Carlo; the convergence is shown in Figure 11. The transport map chosen was a NAF and of the same form as the NAF used in Section 4.1 and specified in Appendix B.2. We used 8000 iterations of Adam with learning rate 0.001. It appears that this QMC sequence generally performed worse than standard Monte-Carlo. These negative findings dissuaded us from further exploring QMC in this work. However, it remains to be seen whether more advanced randomised QMC sequences, such as the scrambled Sobol sequence that was used in Wenzel et al. (2018), provide performance gains relative to standard Monte Carlo.

C.4 Investigating the Choice of Reference Distribution

All the experiments in the main text used a Gaussian distribution as the reference distribution Q. However, different reference distributions could potentially offer some advantages, for instance in capturing thicker tails or multimodality (Izmailov et al., 2020). To investigate, we used the IAF as our transport map and compared



(a) (Prototypical) quasi Monte Carlo

(b) Monte Carlo

Figure 10: A size 256 quasi Monte Carlo point set (left) against a 256 length uniform sequence (right), each alongside their pushforwards to the standard Gaussian.

Sampling Method	Sinusoidal	Banana	Multimodal
Random shifted (modulo 1) grid	0.59	0.53	0.22
Monte Carlo	**0.55**	**0.39**	**0.16**

Table 3: The resulting W_1 metrics from sampling using either quasi Monte Carlo or standard Monte Carlo. The transport map used was a NAF with 8000 iterations of Adam. Bold values indicate which sampling rule performed best on each target.

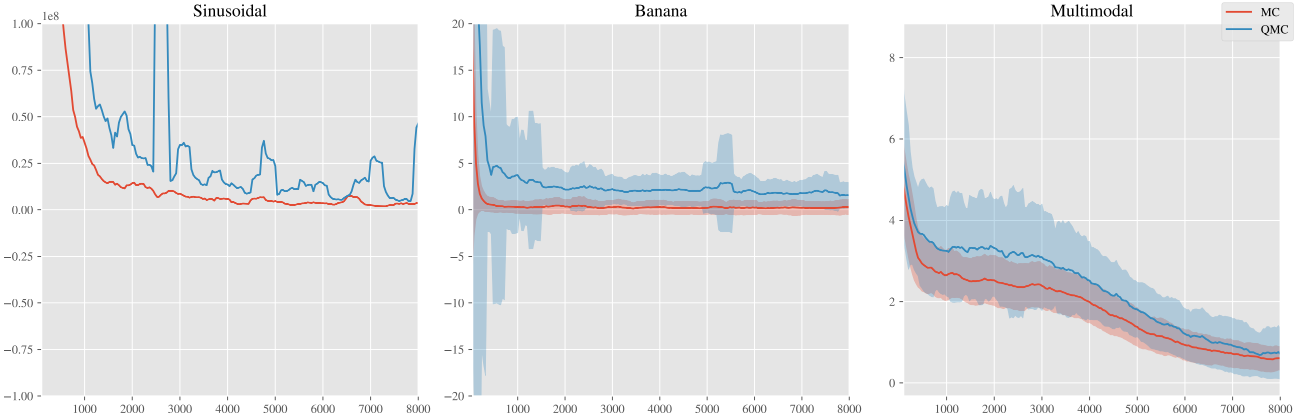


Figure 11: Loss function against number of iterations of Adam for standard Monte Carlo (MC) and quasi Monte Carlo (QMC).

the following reference distributions: a mixture of two Gaussians, a symmetric multivariate Laplace distribution, and the standard Gaussian used in Section 4. The mixture of two Gaussians reference distribution was of the form $\frac{1}{2}\mathcal{N}((0,-3),I_2)+\frac{1}{2}\mathcal{N}((0,3),I_2)$. The multivariate Laplace reference distribution was of the form Laplace$((0,0),I_2)$. The IAF we employed was the same one used in Section 4.1 and fully specified in Appendix B.2. For each experiment we used the Adam optimiser with learning 0.001 with 10,000 iterations of Adam. The target distributions were the synthetic distributions used in Section 4.1 and fully specified in Appendix B.2. Results are shown in Table 4 and notable output is shown in Figure 12.

Looking at Figure 12, the heavier tails of the Laplace reference distribution resulted in heavier tailed output. Furthermore, the Gaussian mixture reference allowed the IAF two capture two modes, however it was unable to capture all four modes of the multimodal target. Results in Table 4 suggest it may be useful to consider the choice of Q as part of the optimisation problem to be solved, although we did not attempt to do so in this work.

Reference Distribution Q	Sinusoidal	Banana	Multimodal
Laplace	0.45	0.25	1.4
Gaussian Mixture	0.23	0.25	0.46
Gaussian	0.38	0.20	0.67

Table 4: The resulting W_1 metrics from using different reference distributions Q in KSD-based measure transport for the targets in Section 4.1. The transport map used for each experiment was an IAF and was optimised with 10,000 iterations of Adam.

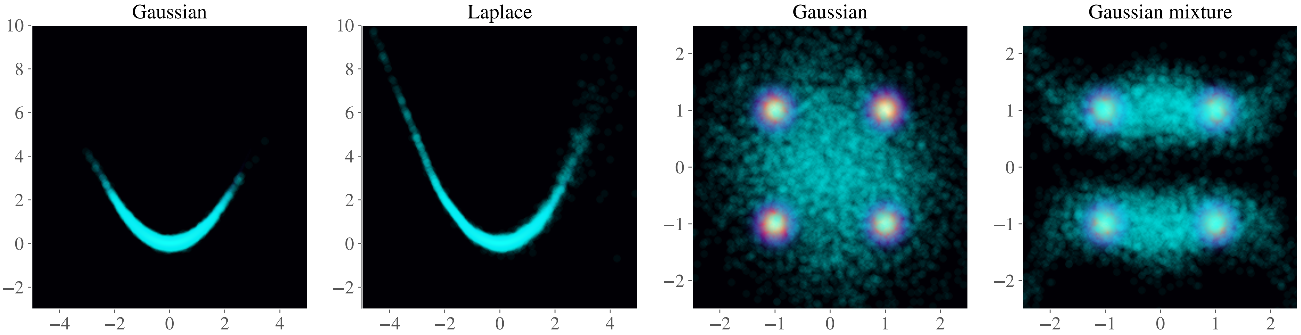


Figure 12: Output of KSD-based measure transport using different reference distributions on the banana and multimodal targets. The reference distribution used in each experiment is indicated in the titles of the corresponding subplots.

C.5 Investigating the Choice of Lengthscale

In this section we investigate how the choice of lengthscale ℓ of the inverse multi-quadric kernel (see Theorem 1) can affect the output of KSD-based measure transport. We will see that, relative to the target distribution, if ℓ is too small the resulting output can be too focused if the target distribution has a relatively large dispersion, and on the other hand, if ℓ is too large, the resulting output can exhibit pathologies. To demonstrate, we will focus on the NAF transport map and consider simple Gaussian targets with covariance matrices $10I_2$ and $0.1I_2$. We will demonstrate the pathologies also occur in a more complex example of the multimodal problem encountered in Section 4.1. Output is shown in Figure 13.

The NAF used is the same one used in the experiments in Section 4.1 and specified in Appendix B.2. We used the Adam optimiser for 10,000 iterations with learning rate 0.001.

C.6 Investigating High Dimensional Targets

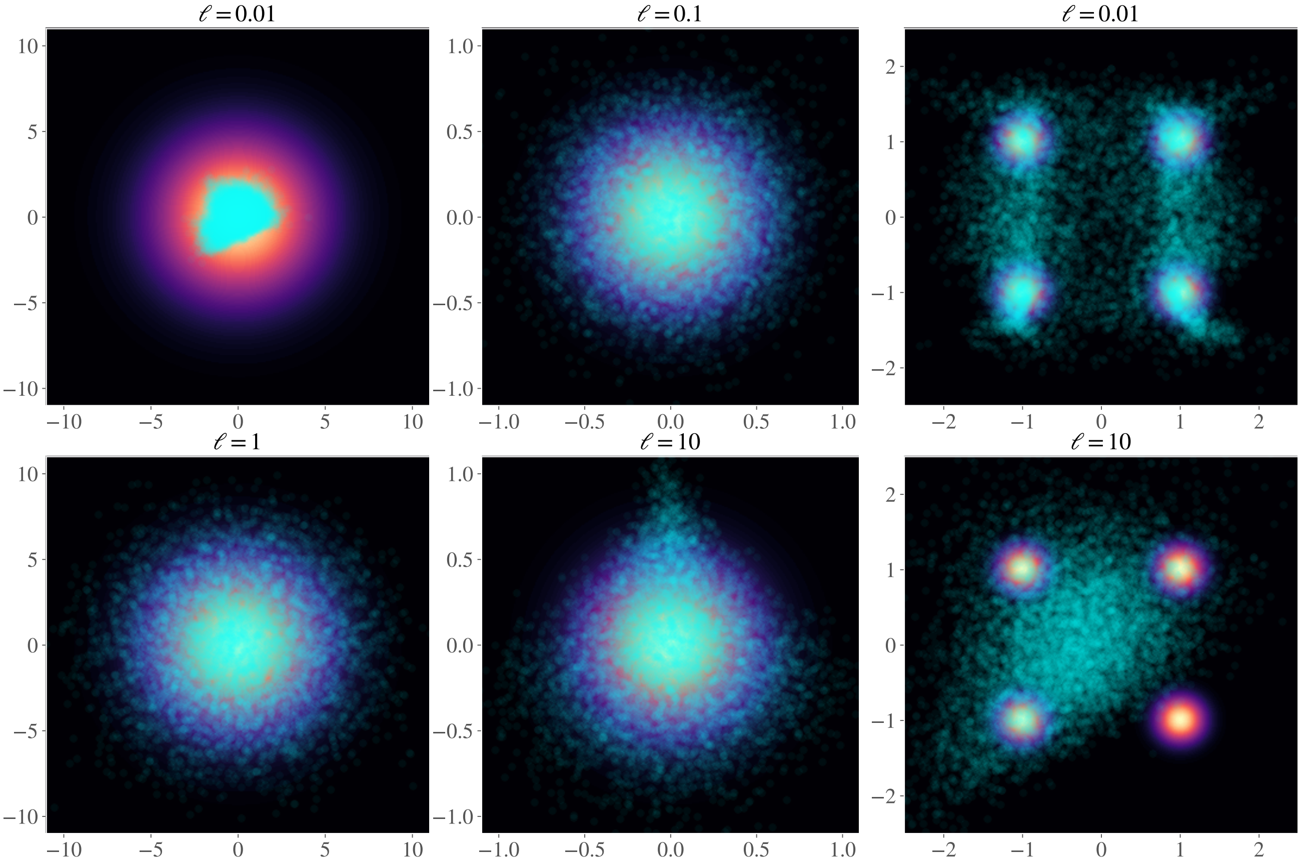
In this section we investigate how the dimensionality of the target distribution affects the performance of KSD-based measure transport against KLD-based measure transport. Our target densities are a natural higher dimension generalisation of the sinusoidal and banana experiments investigated in Section 4.1. The d-dimensional targets we consider are of the form

$$p_1^d(x) := \mathcal{N}\left(x_d; \sin\left(\frac{a}{d-1}\sum_{i=1}^{d-1} x_i\right), \eta_d^2\right)\prod_{i=1}^{d-1}\mathcal{N}(x_i; 0, \eta_i^2),$$

$$p_2^d(x) := \mathcal{N}\left(x_d; b\left(\frac{1}{d-1}\sum_{i=1}^{d-1} x_i\right)^2, \sigma_d^2\right)\prod_{i=1}^{d-1}\mathcal{N}(x_i; 0, \sigma_i^2).$$

Note that, when $d = 2$, p_1^d and p_2^d coincide with the sinusoidal density p_1 and the banana density p_2 specified in Appendix B.2.

The parameter choices for each experiment mirror those specified in Appendix B.2. For p_1^d, we took $a = 1.2, \eta_d^2 = 0.001^2$ and $\eta_i^2 = 1.3^2$ for $i \in \{1, \ldots, d-1\}$. For p_2^d, we took $b = 0.5$, $\sigma_d^2 = 0.1^2$ and $\sigma_i^2 = 1$ for



(a) Gaussian $10I_2$ (b) Gaussian $0.1I_2$ (c) Multimodal

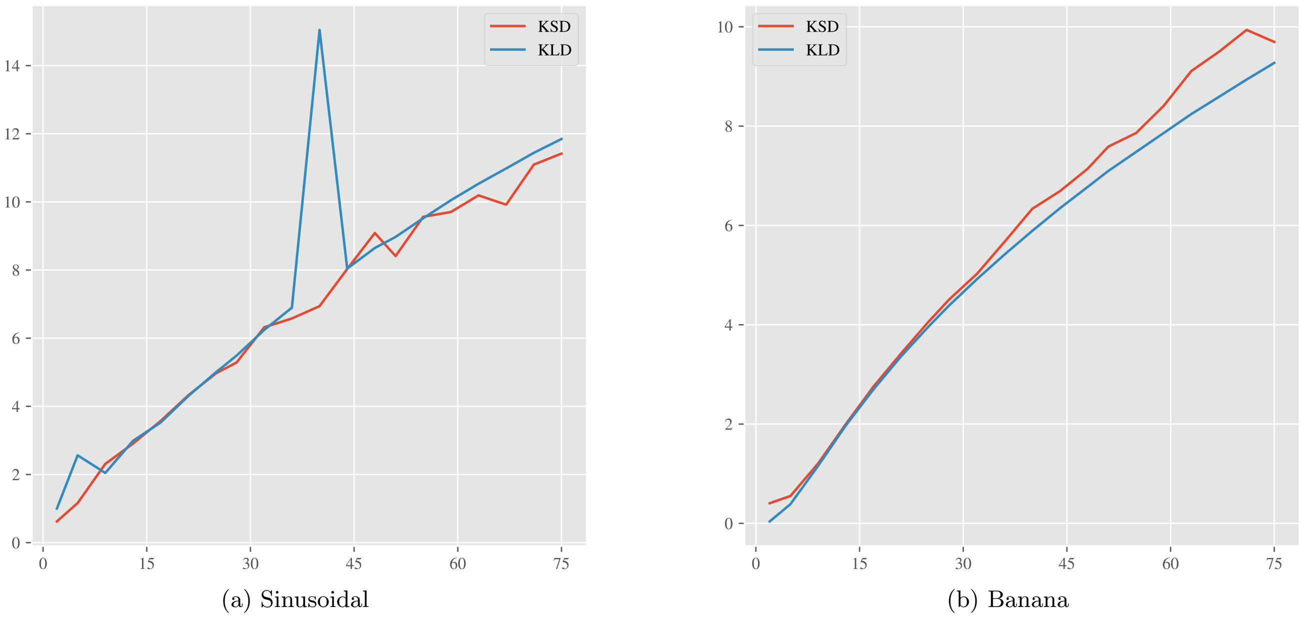
Figure 13: KSD-based measure transport with varying choices of lengthscale ℓ for (a) a mean zero Gaussian target with covariance matrix $10I_2$, (b) a mean zero Gaussian target with covariance matrix $0.1I_2$ and (c) the multimodal problem encountered in Section 4.1. The lengthscale ℓ used in each experiment is indicated in the titles of the corresponding subplots.

$i \in \{1, \ldots, d-1\}$. We only considered one class of transport maps, the Inverse Autogressive Flows. Refer to Kingma et al. (2016) or Appendix B.2 for full details. For each experiment in d-dimensions, we used a single IAF where the dimensionality of the hidden units in the single hidden layer of the underlying autoregressive neural network was $d + 20$. For each experiment we used $30,000$ iterations of Adam, with a learning rate of 0.001. The lengthscale parameter for the sinusoidal experiment of dimension d was taken as $\ell = 0.1 + \frac{20-0.1}{73}(d-2)$ and for the banana experiment of dimension d, we took $\ell = 0.1 + \frac{15-0.1}{73}(d-2)$. Note that is a linear scale where, for $d = 75$, the lengthscale parameters are 20 and 15 respectively for the sinusoidal and banana experiment, and at $d = 2$ the lengthscale parameters are 0.1 in both the sinusoidal and banana experiment. The specific choices of ℓ were taken by investigating the output of the KSD-based measure transport at $d = 75$ and performing a grid search optimisation on the hyperparameter ℓ. We then used linear interpolation between these optimal ℓ values and the lenghtscale parameter used in Section 4.1 for $d = 2$. We do note that improved output can be obtained if one is willing to tune ℓ for the intermediary dimensions. Output for $2 \leq d \leq 75$ is shown in Figure 14.

For the sinusoidal experiment (Figure 14a), KSD-based measure transport consistently either performs equally or outperforms KLD-based measure transport. For the banana experiment (Figure 14b), there is some evidence KSD-based measure transport's performance deteriorates relative to KLD-based measure transport for $d > 35$.

C.7 Investigating the Effect of Input Dimension in the ReLU Network Transport Map

In this section we investigate how changing input dimension of the ReLU network transport map can effect output. In order to isolate the input dimension as our variable of investigation, we fix the topology of the ReLU



(a) Sinusoidal

(b) Banana

Figure 14: Output of KSD-based measure transport vs. KLD-based measure transport on the sinusoidal and banana targets. In both plots, the resulting Wasserstein score is plotted against the dimension of the target.

and consider ReLU networks of the form

$$f_{ReLU} = F_3 \circ \sigma \circ F_2 \circ \sigma \circ F_1,$$

where $F_1 : \mathbb{R}^p \to \mathbb{R}^{20}$, $F_2 : \mathbb{R}^{20} \to \mathbb{R}^{20}$ and $F_3 : \mathbb{R}^{20} \to \mathbb{R}^2$ are affine transformations and σ is the ReLU non-linearity, defined in Appendix A.4. We consider input dimensions $p = 1, 2, 3, 4, 5$. The target we considered is the multimodal target of Section 4.1. For each experiment, we pretrained each ReLU network on the reference distribution $\mathcal{N}((0,0), I_2)$ for $10,000$ iterations of Adam with learning rate 0.001. We then trained on the multimodal target for $20,000$ iterations of Adam, again with learning rate 0.001. Due to the random effects of initialisation (see Figure 16), we ran the experiments for 10 different initialisations and report the best ones, see Table 5. The resulting transport maps are shown in Figure 15. As we can see, the transport map struggles with the multiple modes when the input dimension was $p = 1$ or $p = 2$.

Input dimension	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
W_1	0.88	0.39	0.19	0.27	0.31

Table 5: The W_1 distance to the target P, as a function of the dimension p of the reference distribution Q, using the ReLU neural network transport map.

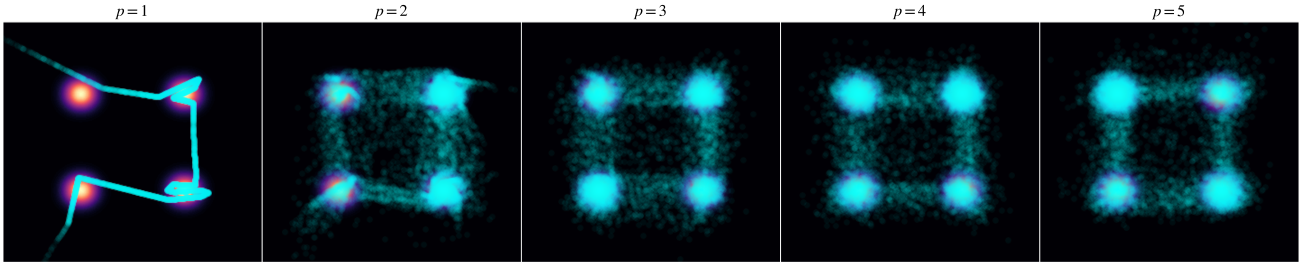


Figure 15: Output of KSD-based measure transport using the ReLU network transport with varying input dimension. The input dimension used in each experiment is indicated in the titles of the corresponding subplots.

		Sinusoidal		Banana		Multimodal	
Transport Map	N	KSD-U	KSD-V	KSD-U	KSD-V	KSD-U	KSD-V
IAF	10^4	**0.38**	0.39	**0.20**	0.25	0.67	**0.61**
IAF (stable)	10^4	**0.35**	0.36	**0.16**	0.19	**0.61**	**0.61**
NAF	10^4	**0.55**	0.58	**0.39**	0.43	**0.095**	0.12
SAF	10^4	**0.23**	0.27	**0.20**	0.48	**0.30**	1.2
B-NAF	10^4	**0.78**	0.85	**0.70**	**0.70**	**1.0**	**1.0**
Polynomial (cubic)	10^4	**0.40**	0.61	**0.25**	0.41	0.51	**0.40**
IAF mixture	3×10^4	**1.3**	**1.3**	**0.19**	0.39	**0.037**	0.040
ReLU network	5×10^4	**0.71**	0.96	**0.43**	0.53	**0.22**	1.2

Table 6: Results from the synthetic test-bed using either the U-statistic (KSD-U) or V-statistic (KSD-V) form of KSD as our objective. The first column indicates with parametric class of transport maps was used; full details for each class can be found in Appendix B.2. A map-dependent number of iterations of stochastic optimisation, N, are reported - this is to ensure all the optimisers approximately converged. The table reports the Wasserstein-1 metric between the approximation $T_{\#}Q$ and the target P. Bold values indicate which of KSD-U or KSD-V performed best for each transport map.

C.8 Investigating the Effect of the U-statistic estimator vs. the V-statistic estimator

Recall from Equation (5), that the square of KSD is of the form

$$\mathcal{D}_S(P,P')^2 = \mathbb{E}_{Y,Y'\sim P'}\left[u_p(Y,Y')\right],$$

where p is the density of P. For a given I.I.D. sample $\{y_i\}_{i=1}^n$ from P', there are two natural estimators of $\mathcal{D}_S(P,P')^2$. The first is the *V-statistic* (KSD-V),

$$\hat{\mathcal{D}}_S^V(P,P')^2 = \frac{1}{n^2}\sum_{i=1}^n\sum_{j=1}^n u_p(y_i,y_j),$$

and the second is the *U-statistic* (KSD-U),

$$\hat{\mathcal{D}}_S^U(P,P')^2 = \frac{1}{n(n-1)}\sum_{1\leq i\neq j\leq n} u_p(y_i,y_j),$$

which is simply the V-statistic with the diagonal $i=j$ elements removed. The advantage of U-statistic is that it is unbiased and, for any given sample, provides the minimum-variance unbiased estimator (MVUE) (Liu et al., 2016). On the other hand, the V-statistic provides a non-negative estimator, due to the positive-definiteness of u_p.

To explore the differences between KSD-U and KSD-V as the objective, we re-ran the synthetic test bed experiments along with the majority of the transport maps in Section 4.1. Results are reported in Table 6. It appears that, although KSD-U and KSD-V often have very close outcomes, KSD-U seems to be strictly better than KSD-V. The transport maps and their initialisation were the same as was used in Section 4.1 and fully specified in Appendix B.2. For each experiment we used $10,000$ iterations of Adam with learning rate 0.001.

C.9 Pathologies of KSD for Measure Transport

Since KSD is a score-based method, it may exhibit similar pathologies to other score-based methods; see e.g. Wenliang (2020). In this section, we detail certain pathologies of KSD-based measure transport that we found experimentally and, if available, offer potential mitigation strategies.

Point Convergence: For small batch sizes (e.g. 25) in the sinusoidal synthetic experiment of Section 4.1, it was observed (albeit rarely) that, when using the ReLU transport map, the transport map converged to a limit in which all inputs were mapped to the origin. The origin is the mode of the sinusoidal synthetic density.

This only occurred when using a degenerate initialisation and for small batch sizes. Due to the tightness of the sinusoidal target and the resulting large score values at points x diverging away slightly from the support of the target, the KSD value of output for all the transport maps used in Section 4.1 was generally of the order 10^8. However, for the approximating transport map maps everything to the origin, the resulting KSD score was 200 with $\ell = 0.1$. Thus, from the perspective of KSD, a transport map that maps everything to the origin and thus having poor Wasserstein distance, was considered better than transport maps that obtain smaller Wasserstein distances.

This problem was mitigated by using better initialisations, larger batch sizes and pretraining on the reference distribution.

Multimodal Failure: It was found that, particularly with ReLU neural network transport map, the inferred transport map for the multimodal synthetic distribution in Section 4.1 could fail to find all four high density regions in the target. The outcome was highly dependent on the initialisation and even persisted when pretraining on the reference distribution. For example, in Figure 16 we plot three random initialisations of the ReLU transport map used in Section 4.1. Each transport map was pretrained on their reference distribution for $10,000$ iterations of Adam. The KSD estimates with $\ell = 0.1$ using $10,000$ samples from the approximating distributions were 0.194, 0.205 and 0.147 from left to right respectively. The three outputs achieved broadly similar KSD scores, indicating that KSD was not able to differentiate between these outcomes. This is problem is further exacerbated as the number of modes of the target increases. For discussion of remedies, see Wenliang (2020).

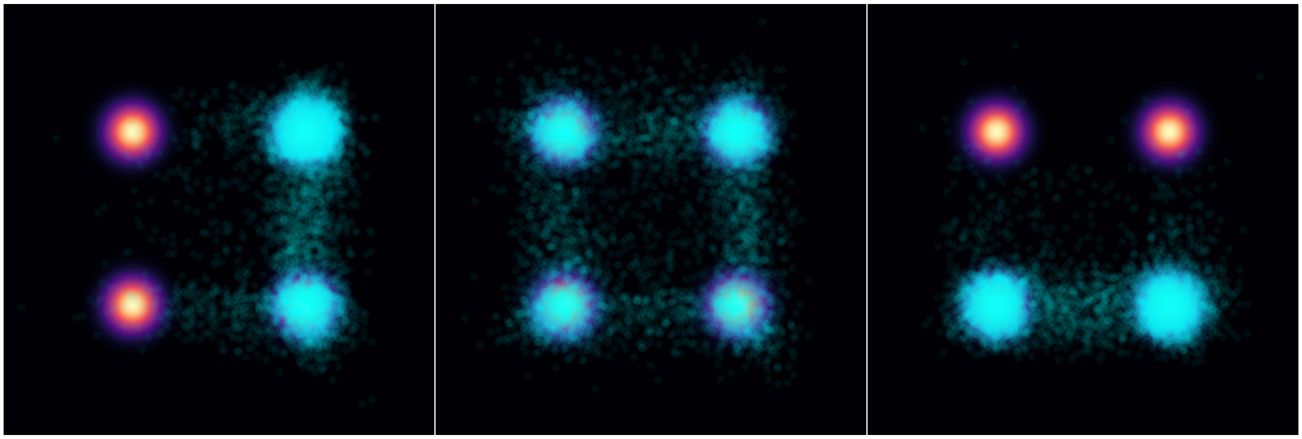


Figure 16: Output of KSD-based measure transport with three random initialisations of the ReLU transport map. Each transport map was pretrained on their reference distribution for $10,000$ iterations of Adam. The KSD estimates with $\ell = 0.1$ using $10,000$ samples from the approximating distributions were 0.194, 0.205 and 0.147 from left to right respectively.

Poor Choice of Lengthscale: Finally, as we have seen in Appendix C.5, if the choice of ℓ is poor, the resulting output of KSD-based measure transport can exhibit pathologies. This is demonstrated, for instance, in Figure 13. This issue is remedied by, for example, employing the median heuristic (Garreau et al., 2018) to set the length-scale parameter in the kernel.