SLICED-WASSERSTEIN AUTO-ENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper we use the geometric properties of the optimal transport (OT) problem and the Wasserstein distances to define a prior distribution for the latent space of an auto-encoder. We introduce Sliced-Wasserstein Auto-Encoders (SWAE), that enable one to shape the distribution of the latent space into any samplable probability distribution without the need for training an adversarial network or having a likelihood function specified. In short, we regularize the auto-encoder loss with the sliced-Wasserstein distance between the distribution of the encoded training samples and a samplable prior distribution. We show that the proposed formulation has an efficient numerical solution that provides similar capabilities to Wasserstein Auto-Encoders (WAE) and Variational Auto-Encoders (VAE), while benefiting from an embarrassingly simple implementation. We provide extensive error analysis for our algorithm, and show its merits on three benchmark datasets.

Scalable generative models that capture the rich and often nonlinear distribution of high-dimensional data, (i.e., image, video, and audio), play a central role in various applications of machine learning, including transfer learning Isola et al. (2017); Murez et al. (2018), super-resolution Ledig et al. (2016); Kolouri & Rohde (2015), image inpainting and completion Yeh et al. (2017), and image retrieval Creswell & Bharath (2016), among many others. The recent parametric generative models, including Generative Adversarial Networks (GANs) Goodfellow et al. (2014); Radford et al. (2015); Arjovsky et al. (2017); Berthelot et al. (2017) and Variational Autoencoders (VAE) Kingma & Welling (2013); Mescheder et al. (2017); Bousquet et al. (2017) enable an unsupervised and end-to-end modeling of the high-dimensional distribution of the training data.

Learning such generative models boils down to minimizing a dissimilarity measure between the data distribution and the output distribution of the generative model. To this end, and following the work of Arjovsky et al. (2017) and Bousquet et al. (2017), we approach the problem of generative modeling from the optimal transport point of view. The optimal transport problem Villani (2008); Kolouri et al. (2017) provides a way to measure the distances between probability distributions by transporting (i.e., morphing) one distribution into another. Moreover, and as opposed to the common information theoretic dissimilarity measures (e.g., f-divergences), the p-Wasserstein dissimilarity measures that arise from the optimal transport problem: 1) are true distances, and 2) metrize a weak convergence of probability measures (at least on compact spaces). Wasserstein distances have recently attracted a lot of interest in the learning community Frogner et al. (2015); Gulrajani et al. (2017); Bousquet et al. (2017); Arjovsky et al. (2017); Kolouri et al. (2017) due to their exquisite geometric characteristics Santambrogio (2015). See the supplementary material for an intuitive example showing the benefit of the Wasserstein distance over commonly used f-divergences.

In this paper, we introduce a new type of autoencoders for generative modeling (Algorithm 1), which we call Sliced-Wasserstein Autoencoders (SWAE), that minimize the sliced-Wasserstein distance between the distribution of the encoded samples and a samplable prior distribution. Our work is most closely related to the recent work by Bousquet et al. (2017) and more specifically the follow-up work by Tolstikhin et al. (2017). However, our approach avoids the need to perform adversarial training in the encoding space and is not restricted to closed-form distributions, while still benefiting from a Wasserstein-like distance measure in the latent space. Calculating the Wasserstein distance can be computationally expensive, but our approach permits a simple numerical solution to the problem. Finally, we note that there has been several concurrent papers, including the work by Deshpande et al. (2018) and Şimşekli et al. (2018), that also looked into the application of sliced-Wasserstein distance in generative modeling. Regardless of the concurrent nature of these papers, our work remains novel and is distinguished from these methods. Deshpande et al. (2018) use the sliced-Wasserstein distance to match the distributions of high-dimensional reconstructed images, which require large number of

slices, $\mathcal{O}(10^4)$, while in our method and due to the distribution matching in the latent space we only need $\mathcal{O}(10)$ slices. We also note that Deshpande et al. (2018) proposed to learn discriminative slices to mitigate the need for a very large number of random projections that is in essence similar to the adversarial training used in GANs, which contradicts with our goal of not using adversarial training. Şimşekli et al. (2018), on the other hand, take an interesting but different approach of parameter-free generative modeling via sliced-Wasserstein flows.

1 Notation and Preliminaries

Let X denote the compact domain of a manifold in Euclidean space and let $x_n \in X$ denote an individual input data point. Furthermore, let ρ_X be a Borel probability measure defined on X. We define the probability density function $p_X(x)$ for input data x to be:

$$d\rho_X(x) = p_X(x)dx$$

Let $\phi : X \to Z$ denote a deterministic parametric mapping from the input space to a latent space Z (e.g., a neural network encoder). Utilizing a technique often used in the theoretical physics community (See Gillespie (1983)), known as Random Variable Transformation (RVT), the probability density function of the encoded samples z can be expressed in terms of ϕ and p_X by:

$$p_Z(z) = \int_X p_X(x)\delta(z - \phi(x))dx, \tag{1}$$

where δ denotes the Dirac distribution function. Similar to variational Auto-Encoders (VAEs) Kingma & Welling (2013) and the Wasserstein Auto-Encoders (WAE) Tolstikhin et al. (2017), our main objective is to encode the input data points $x \in X$ into latent codes $z \in Z$ such that: 1) x can be recovered/approximated from z, and 2) the probability density function of the encoded samples, p_Z, follows a prior distribution q_Z. Let $\psi : Z \to X$ be the decoder that maps the latent codes back to the original space such that

$$p_Y(y) = \int_X p_X(x)\delta(y - \psi(\phi(x)))dx, \tag{2}$$

where y denotes the decoded samples. It is straightforward to see that when $\psi = \phi^{-1}$ (i.e. $\psi(\phi(\cdot)) = id(\cdot)$), the distribution of the decoder p_Y and the input distribution p_X are identical. Hence, in its most general form, the objective of such auto-encoders simplifies to learning ϕ and ψ, so that they minimize a dissimilarity measure between p_Y and p_X, and between p_Z and q_Z. In what follows, we briefly review the existing dissimilarity measures for these distributions.

1.1 Minimizing Dissimilarity between p_X and p_Y

We first emphasize that the VAE often assumes stochastic encoders and decoders Kingma & Welling (2013), while we consider the case of only deterministic mappings. Although, we note that, similar to WAE, SWAE can also be formulated with stochastic encoders. Different measures have been used previously to compute the dissimilarity between p_X and p_Y. Most notably, Nowozin et al. (2016) showed that for the general family of f-divergences, $D_f(p_X, p_Y)$, (including the KL-divergence, Jensen-Shannon, etc.), using the Fenchel conjugate of the convex function f and minimizing $D_f(p_X, p_Y)$ leads to a min-max problem that is equivalent to the *adversarial training* widely used in the generative modeling literature Goodfellow et al. (2014); Makhzani et al. (2015); Mescheder et al. (2017).

Others have utilized the rich mathematical foundation of the OT problem and Wasserstein distances Arjovsky et al. (2017); Gulrajani et al. (2017); Bousquet et al. (2017); Tolstikhin et al. (2017) to define a distance between p_X and p_Y. In Wasserstein-GAN, Arjovsky et al. (2017) utilized the Kantorovich-Rubinstein duality for the 1-Wasserstein distance, $W_1(p_X, p_Y)$, and reformulated the problem as a min-max optimization that is solved through an adversarial training scheme. In a different approach, Bousquet et al. (2017) utilized the autoencoding nature of the problem and showed that minimizing $W_c(p_X, p_Y)$ could be simplified as:

$$inf_{\phi,\psi}\mathbb{E}_{p_X}(c(x, \psi(\phi(x)))) = inf_{\phi,\psi}\int_X p_X(x)c(x, \psi(\phi(x)))dx \tag{3}$$

Note that equation 3 is equivalent to Theorem 1 in Bousquet et al. (2017) for deterministic encoder-decoder pairs, and also note that ϕ and ψ are parametric differentiable models (e.g. neural networks).

Furthermore, equation 3 supports a simple implementation where for i.i.d samples of the input distribution $\{x_n\}_{n=1}^N$ the minimization can be written as:

$$inf_{\phi,\psi} \frac{1}{N} \sum_{n=1}^{N} c(x_n, \psi(\phi(x_n))) \tag{4}$$

equation 3 and equation 4 take advantage of the existence of pairs x_n and $y_n = \psi(\phi(x_n))$, hence calculating the transport distance coincides with summing the transportation costs between all pairs (x_n, y_n). For example, the total transport distance may be defined as the sum of Euclidean distances between all pairs of points. In this paper, we also minimize $W_c(p_X, p_Y)$ following equation 4 to minimize the discrepancy between p_X and p_Y. Next, we review the discrepancy measures between p_Z and q_Z.

1.2 Minimizing Dissimilarity between p_Z and q_Z

If q_Z is a known distribution with an explicit formulation (e.g. Normal distribution) the most straightforward approach for measuring the (dis)similarity between p_Z and q_Z is the log-likelihood of $z = \phi(x)$ with respect to q_Z, formally:

$$sup_{\phi} \int_X p_X(x) log(q_Z(\phi(x)))dx \tag{5}$$

maximizing the log-likelihood is equivalent to minimizing the KL-divergence between p_Z and q_Z, $D_{KL}(p_Z, q_Z)$ (see supplementary material for more details and derivation of Equation equation 5). This approach has two major limitations: 1) The KL-Divergence and in general f-divergences do not provide meaningful dissimilarity measures for distributions supported on non-overlapping low-dimensional manifolds Arjovsky et al. (2017); Kolouri et al. (2018) (see supplementary material), which is common in hidden layers of neural networks, and therefore they do not provide informative gradients for training ϕ, and 2) we are limited to distributions q_Z that have known explicit formulations, which is restrictive as it eliminates the ability to use the much broader class of samplable distributions.

Various alternatives exist in the literature to address the above-mentioned limitations. These methods often sample $\tilde{\mathcal{Z}} = \{\tilde{z}_j\}_{j=1}^N$ from q_Z and $\mathcal{Z} = \{z_n = \phi(x_n)\}_{n=1}^N$ from p_X and measure the discrepancy between these sets (i.e. point clouds). Note that there are no one-to-one correspondences between $\tilde{z}_j$s and z_ns. In their influential WAE paper, Tolstikhin et al. (2017) proposed two different approaches for measuring the discrepancy between $\tilde{\mathcal{Z}}$ and $\mathcal{Z}$, namely the GAN-based and the *maximum mean discrepancy* (MMD)-based approaches. The GAN-based approach proposed in Tolstikhin et al. (2017) defines a discriminator network, $D_Z(p_Z, q_Z)$, to classify $\tilde{z}_j$s and z_ns as coming from 'true' and 'fake' distributions correspondingly, and proposes a min-max adversarial optimization for learning ϕ and D_Z. The MMD-based approach, utilizes a positive-definite reproducing kernel $k : Z \times Z \to \mathbb{R}$ to measure the discrepancy between $\tilde{\mathcal{Z}}$ and $\mathcal{Z}$. The choice of the kernel, however, remains a data-dependent design parameter.

An interesting alternative approach is to use the Wasserstein distance between p_Z and q_Z. Following the work of Arjovsky et al. (2017), this can be accomplished utilizing the Kantorovich-Rubinstein duality and through introducing a min-max problem, which leads to yet another adversarial training scheme similar the GAN-based method in Tolstikhin et al. (2017). Note that, since elements of $\tilde{\mathcal{Z}}$ and $\mathcal{Z}$ are not paired, an approach similar to equation 4 could not be used to minimize the Wasserstein distance. In this paper, we propose to use the sliced-Wasserstein metric, Rabin & Peyré (2011); Rabin et al. (2011); Bonneel et al. (2015); Kolouri et al. (2016b); Carriere et al. (2017); Kolouri et al. (2018), to measure the discrepancy between p_Z and q_Z. We show that using the sliced-Wasserstein distance ameliorates the need for training an adversary network, and provides an efficient, stable, and simple numerical implementation.

Before explaining our proposed approach, it is worthwhile to point out the major difference between learning autoencoders as generative models and GANs. In GANs, one needs to minimize a distance between $\{\psi(\tilde{z}_j)|\tilde{z}_j \sim q_Z\}_{j=1}^M$ and $\{x_n\}_{n=1}^M$, which are high-dimensional point clouds for which there are no correspondences between $\psi(\tilde{z}_j)$s and x_ns. For the autoencoders, on the other hand, there exists correspondences between the high-dimensional point clouds $\{x_n\}_{n=1}^M$ and $\{y_n = \psi(\phi(x_n))\}_{n=1}^M$, and the problem simplifies to matching the lower-dimensional point clouds $\{\phi(x_n)\}_{n=1}^M$ and $\{\tilde{z}_j \sim q_Z\}_{j=1}^M$. In other words, the encoder performs a nonlinear dimensionality reduction, that enables us to solve a simpler problem compared to GANs. Next we introduce the details of our approach.

2 PROPOSED METHOD

In what follows we first provide a brief review of the necessary equations to understand the Wasserstein and sliced-Wasserstein distances and then present our Sliced Wasserstein Autoencoder (SWAE).

2.1 WASSERSTEIN DISTANCES

The Wasserstein distance between probability measures ρ_X and ρ_Y, with corresponding densities $d\rho_X = p_X(x)dx$ and $d\rho_Y = p_Y(y)dy$ is defined as:

$$W_c(p_X, p_Y) = inf_{\gamma \in \Gamma(\rho_X, \rho_Y)} \int_{X \times Y} c(x, y) d\gamma(x, y) \tag{6}$$

where $\Gamma(\rho_X, \rho_Y)$ is the set of all transportation plans (i.e. joint measures) with marginal densities p_X and p_Y, and $c : X \times Y \to \mathbb{R}^+$ is the transportation cost. equation 6 is known as the Kantorovich formulation of the optimal mass transportation problem, which seeks the optimal transportation plan between p_X and p_Y. If there exist diffeomorphic mappings, $f : X \to Y$ (i.e. transport maps) such that $y = f(x)$ and consequently,

$$p_Y(y) = \int_X p_X(x)\delta(y - f(x))dx \xrightarrow[\text{a diffeomorphism}]{\text{When } f \text{ is}} p_Y(y) = det(Df^{-1}(y))p_X(f^{-1}(y)) \tag{7}$$

where $det(D\cdot)$ is the determinant of the Jacobian, then the Wasserstein distance could be defined based on the Monge formulation of the problem (see Villani (2008) and Kolouri et al. (2017)) as:

$$W_c(p_X, p_Y) = min_{f \in MP} \int_X c(x, f(x)) d\rho_X(x) \tag{8}$$

where MP is the set of all diffeomorphisms that satisfy equation 7. As can be seen from equation 6 and equation 8, obtaining the Wasserstein distance requires solving an optimization problem. We note that various efficient optimization techniques have been proposed in the past (e.g. Cuturi (2013); Solomon et al. (2015); Oberman & Ruan (2015)) to solve this optimization. For one-dimensional probability densities, p_X and p_Y, however, the Wasserstein distance has a closed-form solution. Let P_X and P_Y be the cumulative distributions of one-dimensional probability distributions p_X and p_Y, correspondingly. The Wassertein distance can then be calculated as below (see Kolouri et al. (2017) for more details):

$$W_c(p_X, p_Y) = \int_0^1 c(P_X^{-1}(\tau), P_Y^{-1}(\tau)) d\tau, \tag{9}$$

This closed-form solution motivates the definition of sliced-Wasserstein distances.

2.2 SLICED-WASSERSTEIN DISTANCES

Sliced-Wasserstein distance has similar qualitative properties to the Wasserstein distance, but it is much easier to compute. The sliced-Wasserstein distance was used in Rabin & Peyré (2011); Rabin et al. (2011) to calculate barycenter of distributions and point clouds. Bonneel et al. (2015) provided a nice theoretical overview of barycenteric calculations using the sliced-Wasserstein distance. Kolouri et al. (2016b) used it to define positive definite kernels for distributions and Carriere et al. (2017) to define a kernel for persistence diagrams. Sliced-Wasserstein was recently used for learning Gaussian mixture models in Kolouri et al. (2018), and it was also used as a measure of goodness of fit for GANs in Karras et al. (2017).

The main idea behind the sliced-Wasserstein distance is to slice (i.e., project) higher-dimensional probability densities into sets of one-dimensional marginal distributions and compare these marginal distributions via the Wasserstein distance. The slicing/projection process is related to the field of Integral Geometry and specifically the Radon transform Helgason (2011). The relevant result to our discussion is that a d-dimensional probability density p_X can be uniquely represented as the set of its one-dimensional marginal distributions following the Radon transform and the Fourier slice theorem Helgason (2011). These one dimensional marginal distributions of p_X are defined as:

$$\mathcal{R}p_X(t; \theta) = \int_X p_X(x)\delta(t - \theta \cdot x)dx, \;\; \forall \theta \in \mathbb{S}^{d-1}, \; \forall t \in \mathbb{R} \tag{10}$$

where $\mathbb{S}^{d-1}$ is the d-dimensional unit sphere. Note that for any fixed $\theta \in \mathbb{S}^{d-1}$, $\mathcal{R}p_X(\cdot;\theta)$ is a one-dimensional slice of distribution p_X. In other words, $\mathcal{R}p_X(\cdot;\theta)$ is a marginal distribution of p_X that is obtained from integrating p_X over the hyperplane orthogonal to θ.Utilizing these marginal distributions in equation 10, the sliced Wasserstein distance could be defined as:

$$SW_c(p_X, p_Y) = \int_{\mathbb{S}^{d-1}} W_c(\mathcal{R}p_X(\cdot;\theta), \mathcal{R}p_Y(\cdot;\theta))d\theta \tag{11}$$

Given that $\mathcal{R}p_X(\cdot;\theta)$ and $\mathcal{R}p_Y(\cdot;\theta)$ are one-dimensional, the Wasserstein distance in the integrand has a closed-form solution (see equation 9). Moreover, it can be shown that SW_c is a true metric (Bonnotte (2013) and Kolouri et al. (2016a)), and it induces the same topology as W_c, at least on compact sets Santambrogio (2015). A natural transportation cost that has extensively studied in the past is the ℓ_2^2, $c(x, y) = \|x - y\|_2^2$, for which there are theoretical guarantees on existence and uniqueness of transportation plans and maps (see Santambrogio (2015) and Villani (2008)). When $c(x, y) = \|x - y\|_p^p$ for $p \geq 2$, the following upper bound hold for the SW distance:

$$SW_p^p(p_X, p_Y) \leq \alpha_{d,p} W_p^p(p_X, p_Y) \tag{12}$$

where,

$$\alpha_{d,p} = \frac{1}{d} \int_{\mathbb{S}^{d-1}} \|\theta\|_p^p d\theta \leq 1.$$

Chapter 5 in Bonnotte (2013) proves this inequality. In our paper, we are interested in $p = 2$, for which $\alpha_{p,d} = \frac{1}{d}$, and we have:

$$SW_2(p_X, p_Y) \leq \frac{1}{\sqrt{d}} W_2(p_X, p_Y) \tag{13}$$

In the Numerical Implementation Section, we provide a numerical experiment to compare W_2 and SW_2, that confirms the above equation.

2.3 Sliced-Wasserstein Auto-Encoder (SWAE)

Our proposed formulation for the SWAE is as follows:

$$\text{argmin}_{\phi,\psi}\, W_c(p_X, p_Y) + \lambda SW_c(p_Z, q_Z) \tag{14}$$

where ϕ is the encoder, ψ is the decoder, p_X is the data distribution, p_Y is the data distribution after encoding and decoding (equation 2), p_Z is the distribution of the encoded data (equation 1), q_Z is a predefined samplable distribution, and λ indicates the relative importance of the loss functions. To further clarify why we use the Wasserstein distance to measure the difference between p_X and p_Y, but the *sliced-Wasserstein* distance to measure the difference between p_Z and q_Z, we reiterate that the Wasserstein distance for the first term can be solved via equation 4 due to the existence of correspondences between y_n and x_n (i.e., we desire $x_n = y_n$), however, for p_Z and q_Z, analogous correspondences between the $\tilde{z}_i$s and z_js do not exist and therefore calculation of the Wasserstein distance requires an additional optimization step (e.g., in the form of an adversarial network). To avoid this additional optimization, while maintaining the favorable characteristics of the Wasserstein distance, we use the sliced-Wasserstein distance to measure the discrepancy between p_Z and q_Z. Finally, we should note that our formulation of the problem in equation 14 falls under the family of Wasserstein Auto-Encoders (WAE) as introduced by Tolstikhin et al. (2017).

3 Numerical implementation

We now describe the details of our computationally efficient approach for estimating the Wasserstein distance between high-dimensional densities, which is based on the sliced-Wasserstein distance. We provide empirical and theoretical error analysis for different numerical approximations we use.

3.1 Numerical implementation of the Wasserstein distance in 1D

The Wasserstein distance between two one-dimensional probability densities p_X and p_Y is obtained from equation 9. The integral in equation 9 can be numerically estimated using the midpoint Riemann

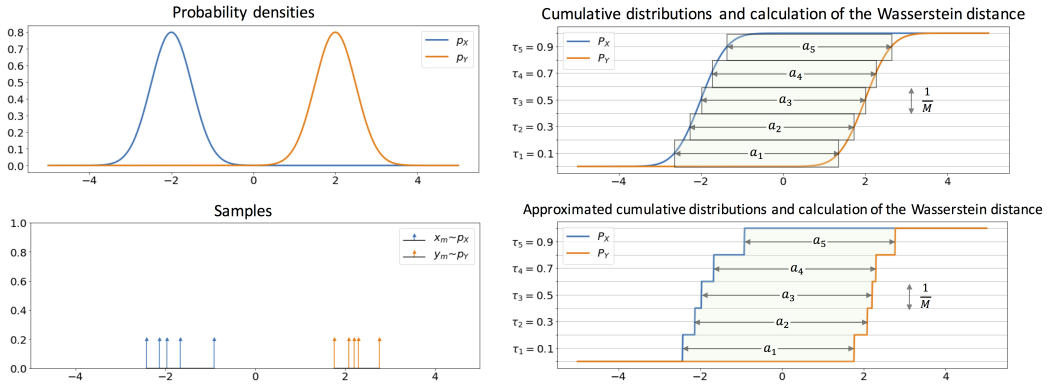

Figure 1: The Wasserstein distance for one-dimensional probability distributions p_X and p_Y (top left) is calculated based on equation 9. For a numerical implementation, the integral in equation 9 is substituted with $\frac{1}{M}\sum_{m=1}^{M} a_m$ where, $a_m = c(P_X^{-1}(\tau_m), P_Y^{-1}(\tau_m))$ (top right). When only samples from the distributions are available $x_n \sim p_X$ and $y_n \sim Y$ (bottom left), the Wasserstein distance is approximated by sorting x_ms and y_ms and letting $a_m = c(x_{i[m]}, y_{j[m]})$, where $i[m]$ and $j[m]$ are the sorted indices (bottom right).

sum, $\frac{1}{M}\sum_{m=1}^{M} a_m$, where $a_m = c(P_X^{-1}(\tau_m), P_Y^{-1}(\tau_m))$ and $\tau_m = \frac{2m-1}{2M}$ (see Fig. 1). In scenarios where only samples from the distributions are available, $x_m \sim p_X$ and $y_m \sim p_Y$, the empirical densities can be estimated as $p_X \approx p_{X,M} = \frac{1}{M}\sum_{m=1}^{M} \delta_{x_m}$ and $p_Y \approx p_{Y,M} = \frac{1}{M}\sum_{m=1}^{M} \delta_{y_m}$, where δ_{x_m} is the Dirac delta function centered at x_m. Therefore the corresponding empirical distribution function of p_X is $P_X(t) \approx P_{X,M}(t) = \frac{1}{M}\sum_{m=1}^{M} u(t - x_m)$ where $u(.)$ is the step function ($P_{Y,M}(t)$ is defined similarly). From Glivenko-Cantelli Theorem we have that $\sup_t |P_{X,M}(t) - P_X(t)| \xrightarrow{a.s.} 0$, where the convergence behavior is achieved via Dvoretzky–Kiefer–Wolfowitz inequality bound: $Prob(\sup_t |P_{X,M}(t) - P_X(t)| > \epsilon) \leq 2\exp(-2M\epsilon^2)$. Calculating the Wasserstein distance with the empirical distribution function is computationally attractive. Sorting x_ms in an ascending order, such that $x_{i[m]} \leq x_{i[m+1]}$ and where $i[m]$ is the index of the sorted x_ms, it is straightforward to see that $P_{X,M}^{-1}(\tau_m) = x_{i[m]}$ (see Fig. 1 for a visualization). The Wasserstein distance can be approximated by first sorting x_ms and y_ms and then calculating:

$$W_c(p_X, p_Y) \approx W_c(p_{X,M}, p_{Y,M}) = \frac{1}{M}\sum_{m=1}^{M} c(x_{i[m]}, y_{j[m]}) \tag{15}$$

The problem of calculating the Wasserstein distance between samples from one-dimensional densities simplifies to solving two sorting problems (solved in $\mathcal{O}(M)$/$\mathcal{O}(Mlog(M))$ best/worst case).

We need to address one final question here. How well does equation 15 approximate the Wasserstein distance, $W_c(p_X, p_Y)$? We first note that the rates of convergence of empirical distributions, for the p-Wasserstein metric (i.e., $c(x, y) = |x - y|^p$) of order $p \geq 1$, have been extensively studied in the mathematics and statistics communities (see for instance Bobkov & Ledoux (2014) and Dedecker et al. (2015)). A detailed description of these rates is, however, beyond the scope of this paper, especially since these rates are dependent on the choice of p. In short, for $p = 1$ it can be shown that $\mathbb{E}(W_1(p_{X,M}, p_X) \leq \frac{C}{\sqrt{M}}$ where C is an absolute constant. Similar results are achieved for $\mathbb{E}(W_p(p_{X,M}, p_X))$ and $(\mathbb{E}(W_p^p(p_{X,M}, p_X)))^{\frac{1}{p}}$, although under more strict assumptions on p_X (i.e., slightly stronger assumptions than having a finite second moment). Using the triangle inequality together with the convergence rates of empirical distributions with respect to the p-Wasserstein distance, see Bobkov & Ledoux (2014), for $W_1(p_{X,M}, p_X)$ (or more generally $W_p(p_{X,M}, p_X)$) we can show that (see supplementary material):

$$\mathbb{E}(W_1(p_X, p_Y) - W_1(p_{X,M}, p_{Y,M})) \leq \frac{C}{\sqrt{M}} \tag{16}$$

for some absolute constant, C. We reiterate that similar bounds could be found for W_p although with slightly more strict assumptions on p_X and p_Y.

3.2 SLICING EMPIRICAL DISTRIBUTIONS

In scenarios where only samples from the d-dimensional distribution, p_X, are available, $x_m \sim p_X$, the empirical density can be estimated as $p_{X,M} = \frac{1}{M}\sum_{m=1}^{M}\delta_{x_m}$. Following equation 10 it is straightforward to show that the marginal densities (i.e. slices) are obtained from:

$$\mathcal{R}p_X(t,\theta) \approx \mathcal{R}p_{X,M}(t,\theta) = \frac{1}{M}\sum_{m=1}^{M}\delta(t - x_m \cdot \theta),\ \forall\theta \in \mathbb{S}^{d-1}, \text{and } \forall t \in \mathbb{R} \tag{17}$$

see the supplementary material for a proof. The Dvoretzky–Kiefer–Wolfowitz upper bound holds for $\mathcal{R}p_X(t,\theta)$ and $\mathcal{R}p_{X,M}(t,\theta)$.

3.3 MINIMIZING SLICED-WASSERSTEIN VIA RANDOM SLICING

Minimizing the sliced-Wasserstein distance (i.e., as in the second term of 14) requires an integration over the unit sphere in $\mathbb{R}^d$, i.e., $\mathbb{S}^{d-1}$. In practice, this integration is approximated by using a simple Monte Carlo scheme that draws uniform samples from $\mathbb{S}^{d-1}$ and replaces the integral with a finite-sample average,

$$\min_\phi SW_c(p_Z, q_Z) \approx \min_\phi \frac{1}{|\Theta|}\sum_{\theta_l\in\Theta} W_c(\mathcal{R}p_Z(\cdot;\theta_l), \mathcal{R}q_Z(\cdot;\theta_l))$$

Such Monte Carlo estimation was used in Rabin & Peyré (2011), and later used in Bonneel et al. (2015); Kolouri et al. (2018); Şimşekli et al. (2018); Deshpande et al. (2018). Moreover, the global minimum for $SW_c(p_Z, q_Z)$ is also a global minimum for each $W_c(\mathcal{R}p_Z(\cdot;\theta_l), \mathcal{R}q_Z(\cdot;\theta_l))$. Note that $SW_c(p_Z, q_Z) = \mathbb{E}_{\mathbb{S}^{(d-1)}}(W_c(\mathcal{R}p_Z(\cdot;\theta), \mathcal{R}q_Z(\cdot;\theta)))$. A fine sampling of $\mathbb{S}^{d-1}$, however, is required for a good approximation of $SW_c(p_Z, q_Z)$. Intuitively, if p_Z and q_Z are similar, then their projections with respect to any finite subset of $\mathbb{S}^{d-1}$ would also be similar. This leads to a stochastic gradient descent scheme where in addition to the random sampling of the input data, we also random sample the projection angles from $\mathbb{S}^{d-1}$.

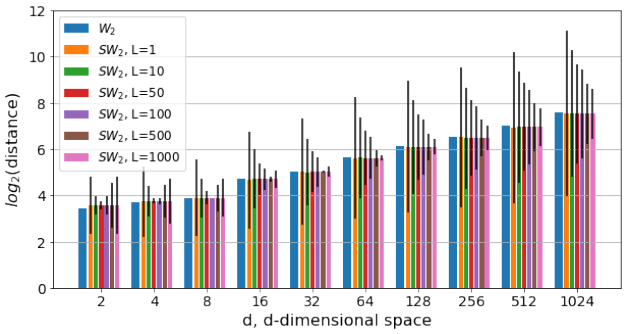

Figure 2: SW approximations (scaled by $1.22\sqrt{d}$) of the W-distance in different dimensions, $d \in \{2^n\}_{n=1}^{10}$, and different number of random slices, L.

A natural question arises on the effect of the number of random slices, $L = |\Theta|$, on the approximation of the SW distance. Here, we devised a simple experiment that demonstrates the effect of L on approximating the SW distance. We generated two random multi-variate Gaussian distributions in a d-dimensional space, where $d \in \{2^n\}_{n=1}^{10}$, to serve as $p_X = \mathcal{N}(\mu_X, \Sigma_X)$ and $p_X = \mathcal{N}(\mu_Y, \Sigma_Y)$. The Wasserstein distance for the two Gaussian distributions has a closed form solution,

$$W_2^2(p_X, p_Y) = \|\mu_X - \mu_Y\|_2^2 + trace(\Sigma_X + \Sigma_Y - 2(\Sigma_X^{\frac{1}{2}}\Sigma_Y\Sigma_X^{\frac{1}{2}})^{\frac{1}{2}}),$$

which served as the ground-truth distance between the distributions. We then measured the SW distance between $M = 1000$ samples generated from the two Gaussian distributions using $L \in \{1, 10, 50, 100, 500, 1000\}$ random slices. We repeated the experiment for each L and d, a thousand times and report the means and standard deviations in Figure 2. Following equation 13 we scaled the SW distance by $\sqrt{d}$. Moreover we found out empirically that $1.22\sqrt{d}\,\mathbb{E}(SW_2(p_{X,M}, p_{Y,M})) \approx W_2(p_X, p_Y)$. It can be seen from Figure 2 that the expected value of the scaled SW-distance closely

Algorithm 1 Sliced-Wasserstein Auto-Encoder (SWAE)

Require: Regularization coefficient λ, and number of random projections, L.
Initialize the parameters of the encoder, ϕ, and decoder, ψ
while ϕ and ψ have not converged **do**
Sample $\{x_1, ..., x_M\}$ from training set (i.e. p_X)
Sample $\{\tilde{z}_1, ..., \tilde{z}_M\}$ from q_Z
Sample $\{\theta_1, ..., \theta_L\}$ from $\mathbb{S}^{K-1}$
Sort $\theta_l \cdot \tilde{z}_M$ such that $\theta_l \cdot \tilde{z}_{i[m]} \leq \theta_l \cdot \tilde{z}_{i[m+1]}$
Sort $\theta_l \cdot \phi(x_m)$ such that $\theta_l \cdot \phi(x_{j[m]}) \leq \theta_l \cdot \phi(x_{j[m+1]})$
Update ϕ and ψ by descending: $\sum_{m=1}^{M} c(x_m, \psi(\phi(x_m))) + \lambda \sum_{l=1}^{L} \sum_{m=1}^{M} c(\theta_l \cdot \tilde{z}_{i[m]}, \theta_l \cdot \phi(x_{j[m]}))$
end while

follows the true Wasserstein distance. A more interesting observation is that the variance of estimation increases for higher dimensions d and decreases as the number of random projections, L, increases. Hence, calculating the SW distance in the image space, as in Deshpande et al. (2018), requires a very large number of projections L to get a less variant approximation of the distance.

3.4 Putting it all together

To optimize the proposed SWAE objective function in equation 14 we use a stochastic gradient descent scheme as described here. In each iteration, let $\{x_m \sim p_X\}_{m=1}^{M}$ and $\{\tilde{z}_m \sim q_Z\}_{m=1}^{M}$ be i.i.d random samples from the input data and the predefined distribution, q_Z, correspondingly. Let $\{\theta_l\}_{l=1}^{L}$ be randomly sampled from a uniform distribution on $\mathbb{S}^{d-1}$. Then using the numerical approximations described in this section, the loss function in equation 14 can be rewritten as:

$$\mathcal{L}(\phi, \psi) = \frac{1}{M} \sum_{m=1}^{M} c(x_m, \psi(\phi(x_m))) + \frac{\lambda}{LM} \sum_{l=1}^{L} \sum_{m=1}^{M} c(\theta_l \cdot \tilde{z}_{i[m]}, \theta_l \cdot \phi(x_{j[m]})) \tag{18}$$

where $i[m]$ and $j[m]$ are the indices of sorted $\theta_l \cdot \tilde{z}_m$s and $\theta_l \cdot \phi(x_m)$ with respect to m, correspondingly. The steps of our proposed method are presented in Algorithm 1. It is worth pointing out that sorting is by itself an optimization problem (which can be solved very efficiently), and therefore the sorting followed by the gradient descent update on ϕ and ψ is in essence a min-max problem, which is being solved in an alternating fashion. Finally, we point out that each iteration of SWAE costs $\mathcal{O}(LMlog(M))$ operations.

4 Experiments

In our experiments we used three image datasets, namely the MNIST dataset by LeCun (1998), the CelebFaces Attributes Dataset (CelebA) by Liu et al. (2015), and the LSUN Bedroom Dataset by Yu et al. (2015). For the MNIST dataset we used a simple autoencoder with mirrored classic deep convolutional neural networks with 2D average poolings, leaky rectified linear units (Leaky-ReLu) as the activation functions, and upsampling layers in the decoder. For the CelebA and LSUN datasets we used the DCGAN Radford et al. (2015) architecture similar to Tolstikhin et al. (2017).

To test the capability of our proposed algorithm in shaping the latent space of the encoder, we started with the MNIST dataset and trained SWAE to encode this dataset to a two-dimensional latent space (for the sake of visualization) while enforcing a match between p_X and p_Y and p_Z and q_Z. We chose four different samplable distributions as shown in Figure 3. It can bee seen that SWAE can successfully embed the dataset into the latent space while enforcing p_Z to closely follow q_Z. In addition, we sample the two-dimensional latent spaces on a 25×25 grid in $[-1, 1]^2$ and decode these points to visualize their corresponding images in the digit/image space.

To get a sense of the convergence behavior of SWAE, and similar to the work of Karras et al. (2017), we calculate the Sliced Wasserstein distance between p_Z and q_Z as well as p_X and p_Y at each batch iteration where we used p-LDA Wang et al. (2011) to calculate projections (See supplementary material). We compared the convergence behavior of SWAE with the closest related work, WAE Tolstikhin et al. (2017) (specifically WAE-GAN) where an adversarial training is used to match p_Z to q_Z, while the loss function for p_X and p_Y remains exactly the same between the two methods. We

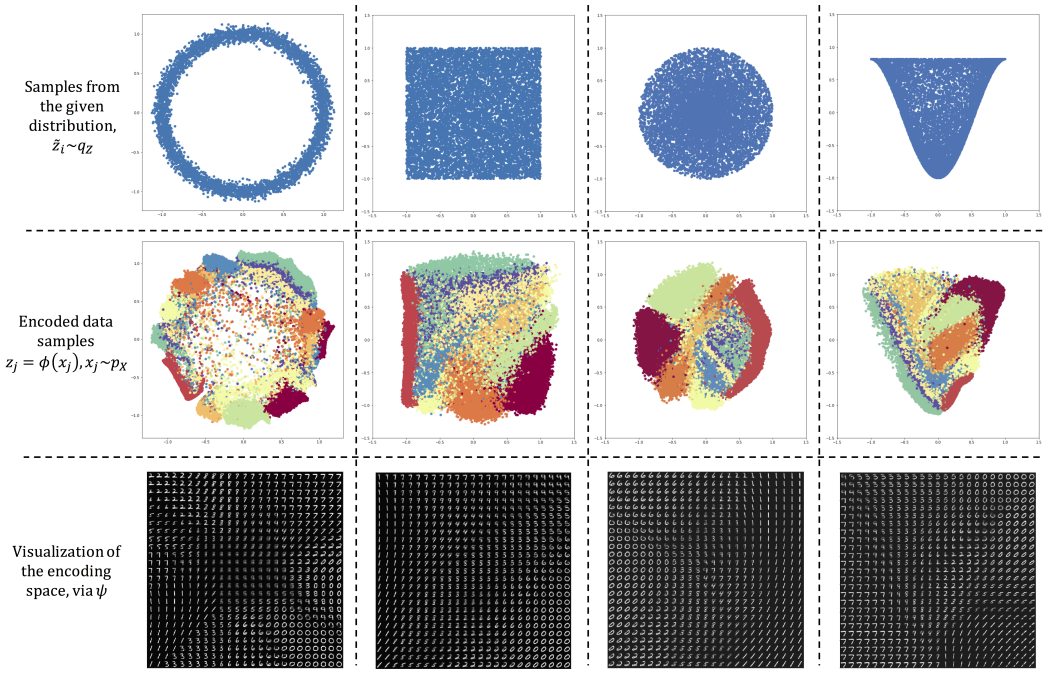

Figure 3: The results of SWAE on the MNIST dataset with a two-dimensional embedding space for four different distributions as ,q_Z, namely the ring distribution (top left), the uniform distribution (bottom left), the uniform polar distribution (top right), and a custom polar distribution (bottom right). Note that the far right visualization demonstrates the decoding of a 25×25 grid in $[-1, 1]^2$.

repeated the experiments 100 times and report the summary of results in Figure 4. We mention that the exact same models and optimizers were used for both methods in this experiment. An interesting observation, here is that while WAE-GAN provides good or even slightly better generated random samples for MNIST (lower sliced-Wasserstein distance between p_X and p_Y), it fails to provide a good match between p_Z and q_Z for the choice of the prior distribution reported in Figure 4. This phenomenon seems to be related to the mode-collapse problem of GANs, where the adversary fails to sense that the distribution is not fully covered. Finally, in our experiments we did not notice a significant difference between the computational time for SWAE and WAE-GAN. For the MNIST experiment and on a single NVIDIA Tesla $P100$ GPU, each batch iteration (batchsize=500) of WAE-GAN took 0.2571 ± 0.0435(sec) while SWAE (with $L = 50$ projections) took 0.2437 ± 0.0391(sec).

Dataset	**Iteration**	**log SW(p_z, q_z)**		**log SW(p_x, p_y)**		**NLL($Z \mid q_z$)**	
		SWAE	**WAEGAN**	**SWAE**	**WAEGAN**	**SWAE**	**WAEGAN**
CelebA	10000	$\mathbf{-0.80 \pm 0.07}$	-0.76 ± 0.05	$\mathbf{-2.19 \pm 0.03}$	-1.96 ± 0.04	$\mathbf{31374 \pm 594}$	33402 ± 366
CelebA	50000	$\mathbf{-1.74 \pm 0.13}$	-1.13 ± 0.03	$\mathbf{-2.64 \pm 0.01}$	-2.28 ± 0.07	$\mathbf{32341 \pm 161}$	34186 ± 630
LSUN	10000	-0.98 ± 0.17	$\mathbf{-1.18 \pm 0.16}$	-1.88 ± 0.06	$\mathbf{-1.90 \pm 0.07}$	$\mathbf{31226 \pm 727}$	33085 ± 161
LSUN	50000	-1.94 ± 0.12	$\mathbf{-2.34 \pm 0.04}$	$\mathbf{-2.34 \pm 0.04}$	-2.30 ± 0.04	$\mathbf{32192 \pm 172}$	33965 ± 802

Table 1: Quantitative comparison of the SWAE and WAE-GAN using the sliced-Wasserstein distance with discriminant slices in the latent space, $SW(p_Z, q_Z)$, and the output space, $SW(p_X, p_Y)$. The distribution in the 64-dimensional latent space, q_Z, was set to Normal. We also report the negative log-likelihood of $\{z_i = \phi(x_i)\}$ with repect to q_Z for 1000 testing samples for both datasets.

The CelebA face and the LSUN bedroom datasets contain higher degrees of variations compared to the MNIST dataset and therefore a two-dimensional latent-space does not suffice to capture the variations in these datasets (See supplementary material for more details on the dimensionality of the latent space). We used a $K = 64$ dimensional latent spaces for both the CelebA and the LSUN Bedroom datasets, and also used a larger auto-encoder (i.e., DCGAN, following the work of Tolstikhin et al. (2017)). For these datasets SWAE was trained with q_Z being the Normal distribution to enable the calculation of the negative log likelihood (NLL). Table 1 shows the comparison between SWAE and WAE for these two datasets. We note that all experimental parameters were kept the

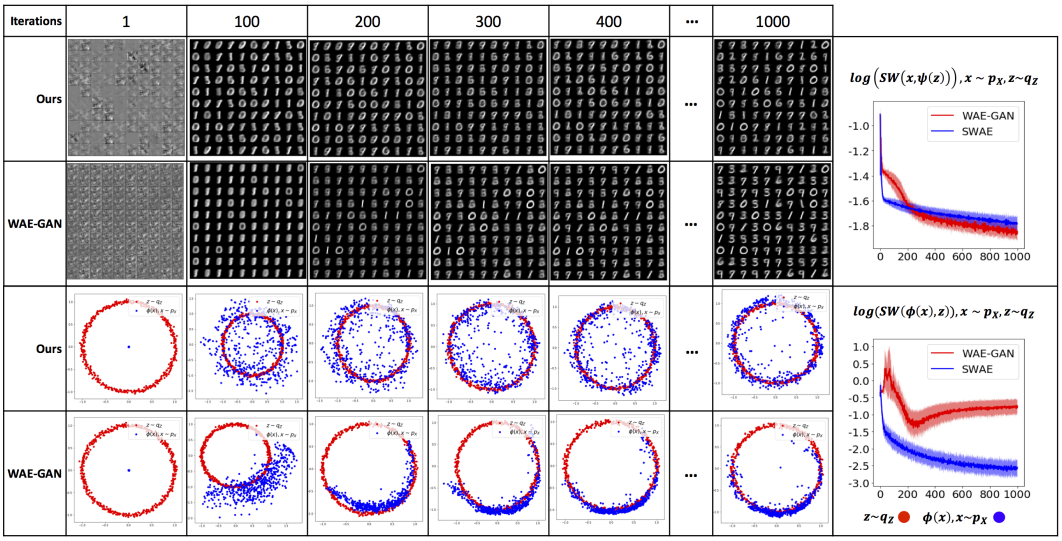

Figure 4: Sample convergence behavior for our method compared to the WAE-GAN, where q_Z is set to a ring distribution (Figure 3, top left). The columns represent batch iterations (batchsize$= 500$). The top half of the table shows results of $\psi(z)$ for $z \sim q_Z$, and the bottom half shows $z \sim q_Z$ and $\phi(x)$ for $x \sim p_X$. It can be seen that the adversarial loss in the latent space does not provide a full coverage of the distribution, which is a similar problem to the well-known 'mode collapse' problem in the GANs. It can be seen that SWAE provides a superior match between p_Z and q_Z while it does not require adversarial training.

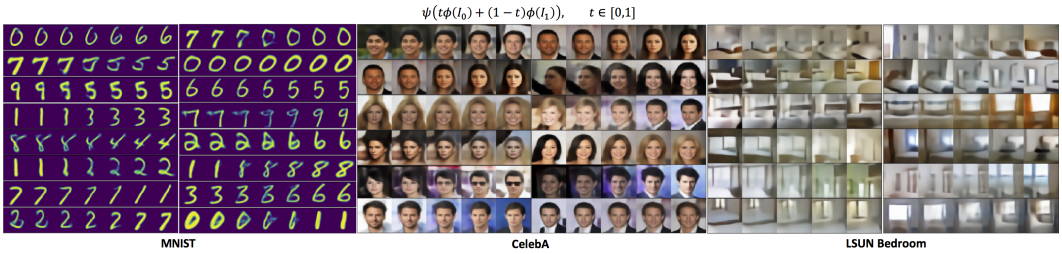

Figure 5: Interpolation in the latent space, $\psi(t\phi(I_0) + (1-t)\phi(I_1))$ for $t \in [0, 1]$.

same to enable an apples to apples comparison. Finally, Figure 5 demonstrates the interpolation between two sample points in the latent space, i.e. $\psi(t\phi(I_0) + (1-t)\phi(I_1))$ for $t \in [0, 1]$, for all three datasets.

5 Conclusions

We introduced Sliced Wasserstein Autoencoders (SWAE), which enable one to shape the distribution of the encoded samples to any samplable distribution without the need for adversarial training or having a likelihood function specified. In addition, we provided a simple and efficient numerical scheme for this problem, which only relies on few inner products and sorting operations in each SGD iteration. We further demonstrated the capability of our method on three image datasets, namely the MNIST, the CelebA face, and the LSUN Bedroom datasets, and showed competitive performance, in the sense of matching distributions p_Z and q_Z, to the techniques that rely on additional adversarial trainings. Finally, we envision SWAE could be effectively used in transfer learning and domain adaptation algorithms where q_Z comes from a source domain and the task is to encode the target domain p_X in a latent space such that the distribution follows the distribution of the target domain.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

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

Sergey Bobkov and Michel Ledoux. One-dimensional empirical measures, order statistics and kantorovich transport distances. *preprint*, 2014.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

Nicolas Bonnotte. *Unidimensional and evolution methods for optimal transportation*. PhD thesis, Paris 11, 2013.

Olivier Bousquet, Sylvain Gelly, Ilya Tolstikhin, Carl-Johann Simon-Gabriel, and Bernhard Schoelkopf. From optimal transport to generative modeling: the VEGAN cookbook. *arXiv preprint arXiv:1705.07642*, 2017.

Mathieu Carriere, Marco Cuturi, and Steve Oudot. Sliced wasserstein kernel for persistence diagrams. *arXiv preprint arXiv:1706.03358*, 2017.

Antonia Creswell and Anil Anthony Bharath. Adversarial training for sketch retrieval. In *European Conference on Computer Vision*, pp. 798–809. Springer, 2016.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pp. 2292–2300, 2013.

Jérôme Dedecker, Aurélie Fischer, Bertrand Michel, et al. Improved rates for wasserstein deconvolution with ordinary smooth error in dimension one. *Electronic journal of statistics*, 9(1):234–265, 2015.

Ishan Deshpande, Ziyu Zhang, and Alexander Schwing. Generative modeling using the sliced wasserstein distance. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. In *Advances in Neural Information Processing Systems*, pp. 2053–2061, 2015.

Daniel T Gillespie. A theorem for physicists in the theory of random variables. *American Journal of Physics*, 51(6):520–533, 1983.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pp. 5769–5779, 2017.

Sigurdur Helgason. The radon transform on rn. In *Integral Geometry and Radon Transforms*, pp. 1–62. Springer, 2011.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Soheil Kolouri and Gustavo K Rohde. Transport-based single frame super resolution of very low resolution face images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4876–4884, 2015.

Soheil Kolouri, Se Rim Park, and Gustavo K Rohde. The radon cumulative distribution transform and its application to image classification. *IEEE transactions on image processing*, 25(2):920–934, 2016a.

Soheil Kolouri, Yang Zou, and Gustavo K Rohde. Sliced Wasserstein kernels for probability distributions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5258–5267, 2016b.

Soheil Kolouri, Se Rim Park, Matthew Thorpe, Dejan Slepcev, and Gustavo K Rohde. Optimal mass transport: Signal processing and machine-learning applications. *IEEE Signal Processing Magazine*, 34(4):43–59, 2017.

Soheil Kolouri, Gustavo K Rohde, and Heiko Hoffman. Sliced Wasserstein distance for learning Gaussian mixture models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3427–, 2018.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint*, 2016.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. *arXiv preprint arXiv:1701.04722*, 2017.

Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4500–4509, 2018.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pp. 271–279, 2016.

Adam M Oberman and Yuanlong Ruan. An efficient linear programming method for optimal transportation. *arXiv preprint arXiv:1509.03668*, 2015.

Julien Rabin and Gabriel Peyré. Wasserstein regularization of imaging problem. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 1541–1544. IEEE, 2011.

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 435–446. Springer, 2011.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, pp. 99–102, 2015.

Umut Şimşekli, Antoine Liutkus, Szymon Majewski, and Alain Durmus. Sliced-wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. *arXiv preprint arXiv:1806.08141*, 2018.

Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)*, 34(4):66, 2015.

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

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

Wei Wang, Yilin Mo, John A Ozolek, and Gustavo K Rohde. Penalized fisher discriminant analysis and its application to image-based morphometry. *Pattern recognition letters*, 32(15):2128–2135, 2011.

Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5485–5493, 2017.

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

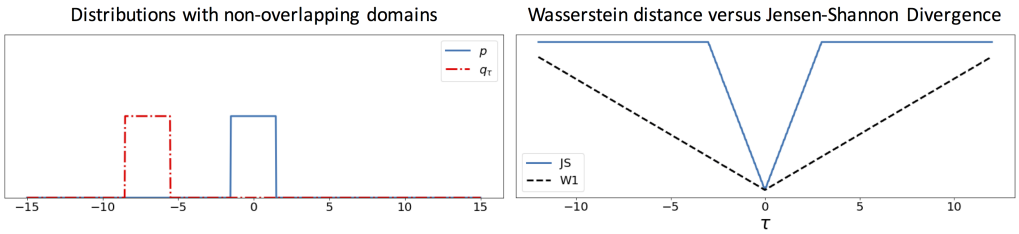

Figure 6: These plots show $W_1(p, q_\tau)$ and $JS(p, q_\tau)$ where p is a uniform distribution around zero and $q_\tau(x) = p(x - \tau)$. It is clear that JS divergence does not provide a usable gradient when distributions are supported on non-overlapping domains.

SUPPLEMENTARY MATERIAL

COMPARISON OF DIFFERENT DISTANCES

Following the example by Arjovsky et al. (2017) and later Kolouri et al. (2018) here we show a simple example comparing the Jensen-Shannon divergence with the Wasserstein distance. First note that the Jensen-Shannon divergence is defined as,

$$JS(p, q) = KL(p, \frac{p+q}{2}) + KL(q, \frac{p+q}{2})$$

where $KL(p, q) = \int_X p(x) log(\frac{p(x)}{q(x)}) dx$ is the Kullback-Leibler divergence. Now consider the following densities, $p(x)$ be a uniform distribution around zero and let $q_\tau(x) = p(x - \tau)$ be a shifted version of the p. Figure 6 show $W_1(p, q_\tau)$ and $JS(p, q_\tau)$ as a function of τ. As can be seen the JS divergence fails to provide a useful gradient when the distributions are supported on non-overlapping domains.

LOG-LIKELIHOOD

To maximize (minimize) the similarity (dissimilarity) between p_Z and q_Z, we can write :

$$\begin{aligned} \text{argmax}_\phi \int_Z p_Z(z) log(q_Z(z)) dz &= \int_Z \int_X p_X(x) \delta(z - \phi(x)) log(q_Z(z)) dx dz \\ &= \int_X p_X(x) log(q_Z(\phi(x))) dx \end{aligned}$$

where we replaced p_Z with equation 1. Furthermore, it is straightforward to show:

$$\begin{aligned} \text{argmax}_\phi \int_Z p_Z(z) log(q_Z(z)) dz &= \text{argmax}_\phi \int_Z p_Z(z) log(\frac{q_Z(z)}{p_Z(z)}) dz \\ &= \text{argmin}_\phi D_{KL}(p_Z, q_Z) \end{aligned}$$

SLICING EMPIRICAL DISTRIBUTIONS

Following equation 10 a distribution can be sliced via:

$$\mathcal{R}p_X(t, \theta) = \int_X p_X(x) \delta(t - \theta \cdot x) dx$$

Figure 7 visualizes two sample slices for an example distribution p_X. Here we calculate a Radon slice of the empirical distribution $p_X(x) = \frac{1}{M} \sum_{m=1}^{M} \delta(x - x_m)$ with respect to $\theta \in \mathbb{S}^{d-1}$. Using the definition of the Radon transform in equation 10 and RVT in equation 1 we have:

$$\begin{aligned} \mathcal{R}p_X(t, \theta) &= \frac{1}{M} \sum_{m=1}^{M} \int_X \delta(x - x_m) \delta(t - \theta \cdot x) dx \\ &= \frac{1}{M} \sum_{m=1}^{M} \delta(t - \theta \cdot x_m) \end{aligned}$$

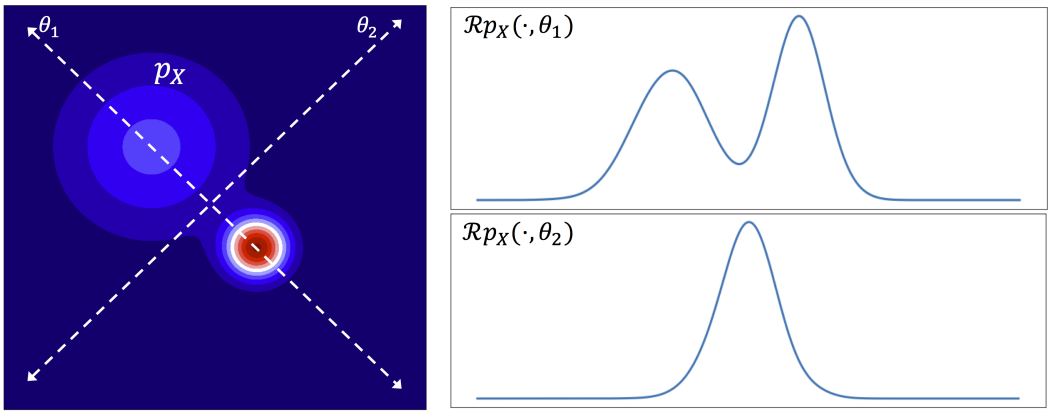

Figure 7: Visualization of the slicing process defined in equation 10

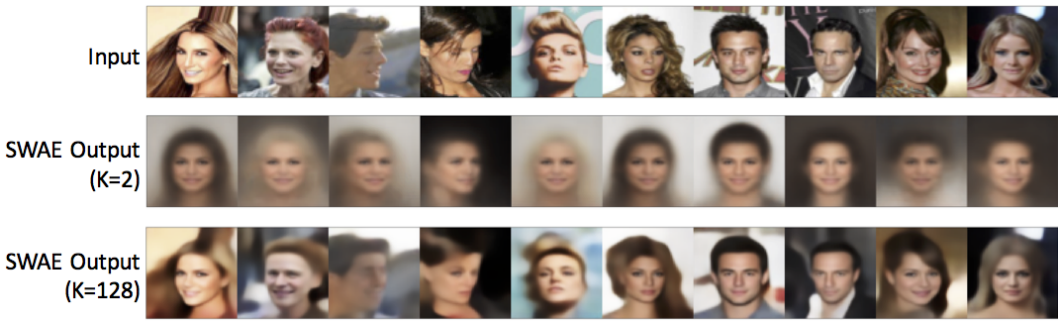

Figure 8: Trained SWAE outputs for sample input images with different embedding spaces of size $K = 2$ and $K = 128$.

Dimensionality of the latent space

Figure 8 demonstrates the outputs of trained SWAEs with $K = 2$ and $K = 128$ for sample input images. The input images were resized to 64×64 and then fed to our autoencoder structure. This effect can also be seen for the MNIST dataset as shown in Figure 9. When the dimensionality of the latent-space (i.e. information bottleneck) is too low the latent space will not contain enough information to reconstruct crisp images. Increasing the dimensionality of the latent space leads to crisper images.

Calculating the sliced Wasserstein distance as a measure of goodness of fit

In this paper we also used the sliced Wasserstein distance as a measure of goodness of fit (for convergence analysis). To provide a fair comparison between different methods, we avoided random projections for this comparison. Instead, we calculated a discriminant subspace to separate $\psi(z)$ from $\psi(\phi(x))$ for $z \sim q_Z$ and $x \sim p_X$, and set the projection parameters θs to the calculated discriminant components. This will lead to only slices that contain discriminant information. We point out that the linear discriminant analysis (LDA) is not a good choice for this task as it only leads to one discriminant component (because we only have two classes). We used the penalized linear discriminant analysis (p-LDA) that utilizes a combination of LDA and PCA. In short, p-LDA solves the following objective function:

$$\begin{aligned} \text{argmax}_\theta \quad & \frac{\theta^T S_T \theta}{\theta^T (S_W + \alpha I)\theta} \\ s.t. \quad & \|\theta\| = 1 \end{aligned}$$

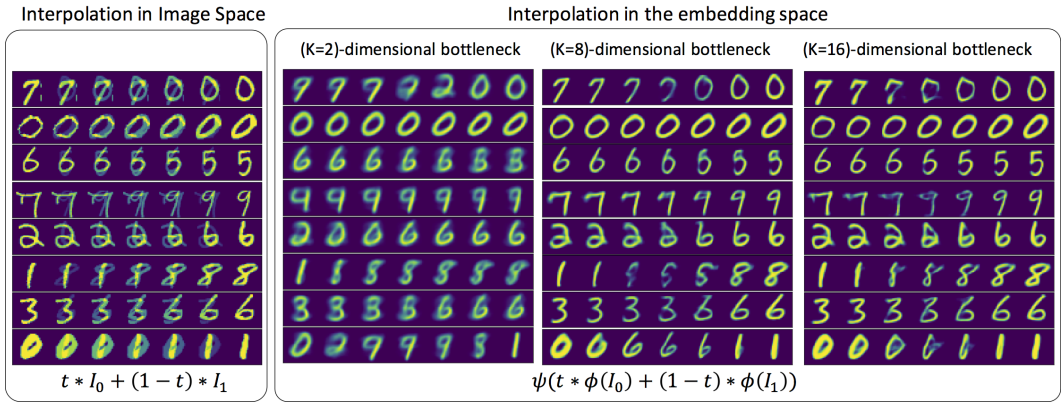

Figure 9: Interpolation results for on the MNIST dataset with various dimensions of the latent space. The parameter $t \in [0, 1]$ indicates the interpolation parameter.

where S_W is the within class covariance matrix, S_T is the data covariance matrix, I is the identity matrix, and α identifies the interpolation between PCA and LDA (i.e. $\alpha = 0$ leads to LDA and $\alpha \to \infty$ leads to PCA).

Error analysis of Wasserstein distance

For $p \geq 1$ we can use the triangle inequality and write

$$\begin{aligned} W_p(p_X, p_Y) &\leq W_p(p_X, p_{X,M}) + W_p(p_Y, p_{X,M}) \\ &\leq W_p(p_X, p_{X,M}) + W_p(p_Y, p_{Y,M}) + W_p(p_{X,M}, p_{Y,M}) \end{aligned}$$

which leads to

$$W_p(p_X, p_Y) - W_p(p_{X,M}, p_{Y,M}) \leq W_p(p_X, p_{X,M}) + W_p(p_Y, p_{Y,M})$$

Taking the expectation of both sides of the inequality and using the empirical convergence bounds of W_p (in this case W_1) we have,

$$\begin{aligned} \mathbb{E}(W_1(p_X, p_Y) - W_1(p_{X,M}, p_{Y,M})) &\leq \mathbb{E}(W_1(p_X, p_{X,M})) + \mathbb{E}(W_1(p_Y, p_{Y,M})) \\ &\leq \frac{C}{\sqrt{M}} \end{aligned}$$

for some absolute constant C, where the last line comes from the empirical convergence bounds of distributions with respect to the Wasserstein distance, see Bobkov & Ledoux (2014).

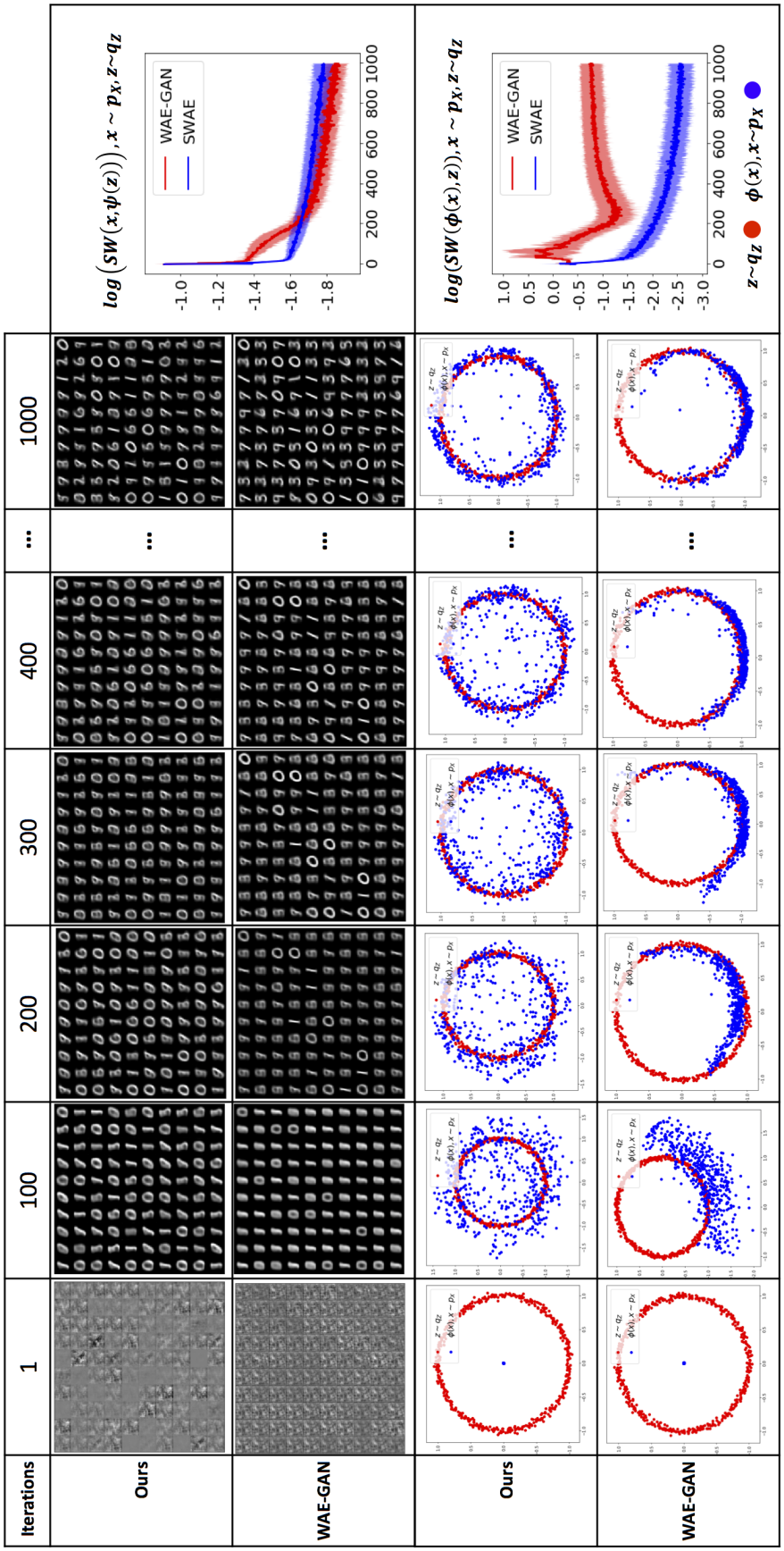

Figure 10: Sample convergence behavior for our method compared to the WAE-GAN Tolstikhin et al. (2017), where q_Z is set to the ring distribution. The columns represent batch iterations (batchsize$=500$). The top half of the table shows results of $\psi(z)$ for $z \sim q_Z$, and the bottom half shows $z \sim q_Z$ and $\phi(x)$ for $x \sim p_X$. It can be seen that SWAE provides a superior match between $\phi(x)$ (i.e. p_Z) and q_Z while being less computationally expensive and not requiring adversarial training.

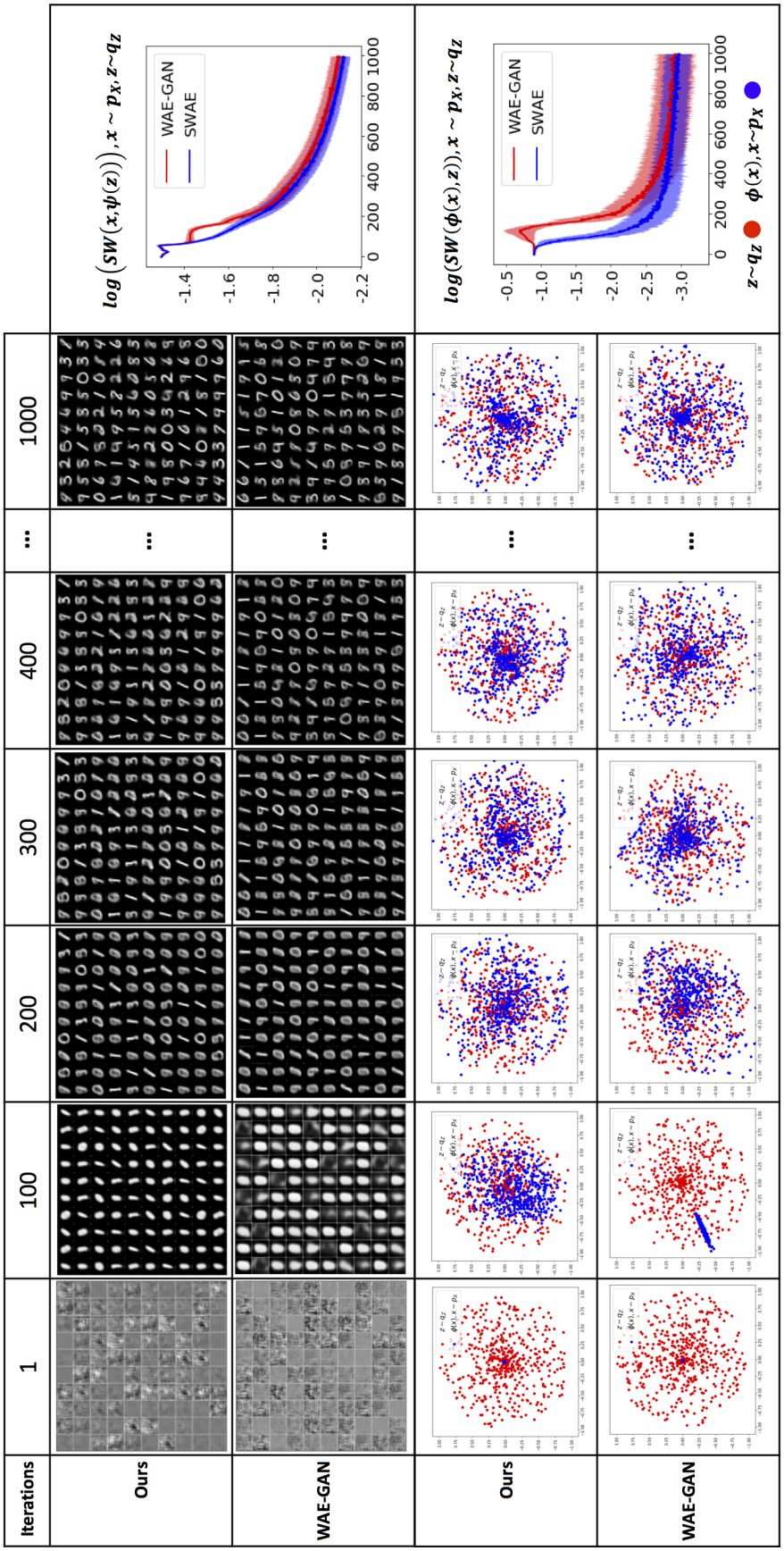

Figure 11: Sample convergence behavior for our method compared to the WAE-GAN Tolstikhin et al. (2017), where q_Z is set to the uniform polar distribution. The columns represent batch iterations (batchsize$=500$). The top half of the table shows results of $\psi(z)$ for $z \sim q_Z$, and the bottom half shows $z \sim q_Z$ and $\phi(x)$ for $x \sim p_X$.It can be seen that SWAE provides a superior match between $\phi(x)$ (i.e. p_Z) and q_Z while being less computationally expensive and not requiring adversarial training.