Journal of Machine Learning Research 21 (2020) 1-28 Submitted 7/19; Revised 4/20; Published 8/20

Cramer-Wold Auto-Encoder

Szymon Knop SZYMONKNOP@GMAIL.COM
Przemysław Spurek PRZEMYSLAW.SPUREK@UJ.EDU.PL
Jacek Tabor JACEK.TABOR@UJ.EDU.PL
Igor Podolak IGOR.PODOLAK@UJ.EDU.PL
Marcin Mazur MARCIN.MAZUR@UJ.EDU.PL
Faculty of Mathematics and Computer Science
Jagiellonian University, Kraków, Poland

Stanisław Jastrzębski STASZEK.JASTRZEBSKI@GMAIL.COM
Center of Data Science / Department of Radiology
New York University, New York, United States

Editor: John Cunningham

Abstract

The computation of the distance to the true distribution is a key component of most state-of-the-art generative models. Inspired by prior works on the Sliced-Wasserstein Auto-Encoders (SWAE) and the Wasserstein Auto-Encoders with MMD-based penalty (WAE-MMD), we propose a new generative model – a Cramer-Wold Auto-Encoder (CWAE). A fundamental component of CWAE is the characteristic kernel, the construction of which is one of the goals of this paper, from here on referred to as the Cramer-Wold kernel. Its main distinguishing feature is that it has a closed-form of the kernel product of radial Gaussians. Consequently, CWAE model has a closed-form for the distance between the posterior and the normal prior, which simplifies the optimization procedure by removing the need to sample in order to compute the loss function. At the same time, CWAE performance often improves upon WAE-MMD and SWAE on standard benchmarks.

Keywords: Auto-Encoder, Generative model, Wasserstein Auto-Encoder, Cramer-Wold Theorem, Deep neural network

1. Introduction

One of the crucial aspects in the construction of generative models is devising efficient methods for computing and minimizing the distance to the true data distribution. In Variational Auto-Encoder (VAE), the distance to the true distribution is measured using KL divergence under the latent variable model and minimized using variational inference. An improvement was brought by the introduction of the Wasserstein metric (Tolstikhin et al., 2017) in the construction of WAE-GAN and WAE-MMD models, which relaxed the need for variational methods. WAE-GAN requires a separate optimization problem to be solved to approximate the used divergence measure, while in WAE-MMD the discriminator has the closed-form obtained from a characteristic kernel[1].

Most recently Kolouri et al. (2018) introduced the Sliced-Wasserstein Auto-Encoder (SWAE), which simplifies distance computation even further. The main innovation of SWAE

1. Kernel is characteristic if it is injective on distributions, see e.g. Muandet et al. (2017).

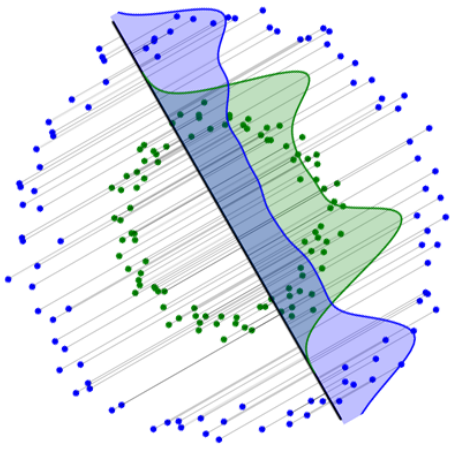

Figure 1: Cramer-Wold distance between two sets is obtained as the mean squared L_2 distance of their smoothed projections on all one-dimensional lines. Figure shows an exemplary (one of many) projection.

was the introduction of the Sliced-Wasserstein distance, a fast to estimate metric for comparing two distributions, based on the mean Wasserstein distance of one-dimensional projections. However, even in SWAE there is no closed-form analytic formula that would enable computing the distance of the sample from the standard normal distribution. Consequently, in SWAE two types of sampling are needed: (i) sampling from the prior distribution and (ii) sampling from one-dimensional projections.

The main contribution of this paper is the introduction of the Cramer-Wold distance between distributions, which is based on MMD distance and a new Cramer-Wold kernel. Cramer-Wold kernel is characteristic, i.e. the embedding is injective, and admits a closed-form in a certain case (see Eq. (12)). Thanks to the closed-form formula, it can be efficiently computed. We use it to construct the Cramer-World Auto-Encoder (CWAE) model, in which the cost function has a closed analytic formula. We demonstrate on standard benchmarks that CWAE is faster to optimise, more stable (no sampling is needed during the learning process) and retains, or even improves, performance compared to both WAE-MMD and SWAE.

The Cramer-Wold kernel can be used as a measure between a sample and a mixture of radial Gaussian distributions. Śmieja et al. (2019) present a semi-supervised generative model SeGMA, which is able to learn a joint probability distribution of data and their classes. It is implemented in a typical auto-encoder framework but uses a mixture of Gaussians as a target distribution in the latent space. In such a situation, the classical Wasserstein kernel is difficult to use since it requires sampling from both (target and real) distributions. SeGMA works efficiently due to the use of Cramer-Wold distance as a maximum mean discrepancy penalty, which yields a closed-form expression for a mixture of spherical Gaussian components, and thus, eliminates the need for sampling.

This paper is arranged as follows. In sections 3 and 4 we introduce and theoretically analyze the Cramer-Wold distance, with the formal definition of a Cramer-Wold kernel in

Section 5. Readers interested mainly in the construction of CWAE may proceed directly to Section 6. Section 7 contains experiments. Finally, we conclude in Section 9.

2. Motivation and related work

One of the ways to look at modern generative models (see, e.g. Tolstikhin et al. (2017)) is to note that each of them tends to minimise a certain divergence measure between the true, but unknown, data distribution P_X and the model distribution $P_{\mathcal{D}}$ that is defined as a possibly random transportation via the given map $\mathcal{D}$ of a fixed distribution $P_{\mathcal{Z}}$, acting on the latent space $\mathcal{Z}$, into X. The most well known are the Kullback-Leibler (KL) and Jensen-Shannon (JS) divergences, which refer to the Variational Auto-Encoder VAE (Kingma et al., 2014) and the Generative Adversarial Network GAN (Goodfellow et al., 2014) models, respectively (although in GAN a saddle-point objective occurs and hence adversarial training is required). However, these measures are often hard to use in a learning process due to some computational problems, including complexity, vanishing gradient, etc.

In recent years new approaches, involving optimal transport (OT) setting (Villani, 2008), appeared in generative modeling. They were based on the use of the Wasserstein or, generally, optimal transport, distance as a measure of divergence between distributions. Beside the classical Wasserstein GAN (Arjovsky et al., 2017) model, we can mention here the Wasserstein Auto-Encoder WAE (Tolstikhin et al., 2017) as well as the Sliced-Wasserstein Auto-Encoder SWAE (Kolouri et al., 2018) as models, which were the inspiration and reference points for our work. In the following two paragraphs we briefly recall the main concepts behind these ideas.

Wasserstein Auto-Encoder (WAE) Tolstikhin et al. (2017) introduce an auto-encoder based generative model with deterministic decoder $\mathcal{D}$ and a, possibly random, encoder $\mathcal{E}$, which is based on minimizing the Wasserstein distance $d_W(P_X, P_{\mathcal{D}})$ between the data and the model distributions. Recall that $d_W(\mu, \nu)$ is given by the following formula

$$d_W^2(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \int \|x_1 - x_2\|_2^2 \, d\gamma(x_1, x_2),$$

where $\Gamma(\mu, \nu)$ is the set of joint probability measures having μ and ν as marginals. By Theorem 1 of Tolstikhin et al. (2017) this leads to the WAE objective function expressed by a sum consisting of two terms: (i) an expected cost of a difference between the data distribution P_X and another distribution on the data space that is obtained by a self-transportation of P_X via, appropriately understood, superposition of $\mathcal{E}$ and $\mathcal{D}$, and (ii) a tuned divergence $d_{\mathcal{Z}}$ between a prior distribution $P_{\mathcal{Z}}$ and another distribution on $\mathcal{Z}$ that is obtained by a transportation of P_X into Z via $\mathcal{E}$. In consequence, assuming a deterministic encoder, the authors introduce two generative models, depending on a specific divergence measure used: WAE-GAN, involving the JS-divergence as $d_{\mathcal{Z}}$ (learned by the adversarial training), and WAE-MMD, using as $d_{\mathcal{Z}}$ the maximum mean discrepancy MMD_k with a suitably established characteristic kernel function k.

Sliced-Wasserstein Auto-Encoder (SWAE) Another contribution that involves optimal transport setting in generative modeling is the work of Kolouri et al. (2018). It differs from WAE in the choice of the divergence measure $d_{\mathcal{Z}}$. Based on a slicing method and

the fact that the Wasserstein distance between one-dimensional distributions can be easily expressed as

$$d_W^2(\mu,\nu) = \int_0^1 (P_\mu^{-1}(t) - P_\nu^{-1}(t))^2\, dt,$$

where P_μ^{-1} and P_ν^{-1} denote the quantile functions of μ and ν, respectively, the authors use the mean value of d_W^2 as $d_{\mathcal{Z}}$, taken over all one-dimensional projections of appropriate distributions on the latent space $\mathcal{Z}$ (see the next section for more details). This idea directly motivated our Cramer-Wold distance.

3. Cramer-Wold distance

Motivated by the prevalent use of normal distribution as the prior in modern generative models, we investigate whether it is possible to simplify and speed up the optimization of such models. As the first step towards this, we introduce Cramer-Wold distance, which has a simple analytical formula for computing the normality of high-dimensional samples. On a high level, our proposed approach uses the traditional L_2 distance of kernel-based density estimation, computed across multiple single-dimensional projections of the true data and the output distribution of the model. We base our construction on the following two popular tricks of the trade: sliced based decomposition and smoothing distributions.

Sliced-based decomposition of a distribution Following Kolouri et al. (2018); Deshpande et al. (2018), the initial concept is to leverage the Cramer–Wold Theorem (Cramér and Wold, 1936) and the Radon Transform (Deans, 1983) to reduce computing distance between two distributions to one-dimensional calculations. For v in the unit sphere $S_D \subset \mathbb{R}^D$, the projection of a set $X \subset \mathbb{R}^D$ onto the space spanned by v is given by $v^T X$, whereas the projection of $N(m, \alpha I)$ is $N(v^T m, \alpha)$. The Cramer-Wold theorem states that two multivariate distributions can be uniquely identified by all their one-dimensional projections. Hence, to obtain the key component of SWAE model, i.e. the sliced-Wasserstein distance between two samples $X, Y \in \mathbb{R}^D$, we compute the mean Wasserstein distance between all one-dimensional projections[2]:

$$d_{\mathrm{sw}}^2(X,Y) = \int_{S_D} d_W^2(v^T X, v^T Y)\, d\sigma_D(v), \tag{1}$$

where S_D denotes the unit sphere in $\mathbb{R}^D$ and σ_D is the normalised surface measure on S_D. This approach is effective since the one-dimensional Wasserstein distance between samples has the closed-form; and therefore, to estimate Eq. (1), one has to sample only from all one-dimensional projections.

Smoothing distributions Using the slice-based decomposition requires defining a distance measure between two sets of samples in a one-dimensional space. To this end, we use an approach commonly applied in statistics to compare samples or distributions, which is to first smoothen (sample) distribution with a Gaussian kernel. For a sample $R = (r_i)_{i=1..n} \subset \mathbb{R}$

2. Observe that in space H with the scalar product $\langle \cdot, \cdot \rangle$, each one-dimensional projection is given by a scalar product $x \to \langle x, v \rangle$, for some $v \in H$. Consequently, this projection is proportional to $x \to \langle x, \frac{v}{\|v\|} \rangle$, which is a 1D-projection with respect to the element from the unit sphere.

by its smoothing with Gaussian kernel $N(0,\gamma)$ we understand

$$\mathrm{sm}_\gamma(R) = \frac{1}{n}\sum_i N(r_i,\gamma),$$

where by $N(m,S)$ we denote the one-dimensional normal density with mean m and variance S. This produces a distribution with regular density and is commonly used in kernel density estimation.

If R comes from the normal distribution with standard deviation close to one, the asymptotically optimal choice of $\gamma = (4/3n)^{2/5}$ is given by the Silverman's rule of thumb (see Silverman (1986)). Theoretically, one can choose an arbitrary fixed γ. However, we use an approach similar to the Bowman-Foster normality test (Bowman and Foster, 1993)[3]. For a continuous density f, its smoothing $\mathrm{sm}_\gamma(f)$ is given by the convolution with $N(0,\gamma)$, and in the special case of Gaussians we have $\mathrm{sm}_\gamma(N(m,S)) = N(m,S+\gamma)$.

Cramer-Wold distance We are now ready to introduce the Cramer-Wold distance. For the convenience of the reader, we formulate the distance between samples first, and then between sample and a distribution. For a formal definition of distance between distribution see paragraph "Generalised Cramer-Wold kernel" in Section 6. In a nutshell, we propose to compute the squared distance between two samples by considering the mean squared L_2 distance between their smoothed projections over all one-dimensional subspaces. By the squared L_2 distance between functions $f,g:\mathbb{R}\to\mathbb{R}$ we refer to $\|f-g\|_2^2 = \int |f(x)-g(x)|^2 dx$. A key feature of this distance is that it permits a closed-form in the case of the normal distribution.

The following algorithm fully defines the Cramer-Wold distance between two samples $X=(x_i)_{i=1..n}, Y=(y_j)_{j=1..k}\subset\mathbb{R}^D$ (for illustration of Steps 1 and 2 see Figure 1):

1. given v in the unit sphere $S(0,1)\subset\mathbb{R}^D$, consider all the projections $v^TX=(v^Tx_i)_{i=1..n}$ and $v^TY=(v^Ty_j)_{j=1..k}$,
2. compute the squared L_2 distance of the densities $\mathrm{sm}_\gamma(v^TX)$ and $\mathrm{sm}_\gamma(v^TX)$:
$$\|\mathrm{sm}_\gamma(v^TX)-\mathrm{sm}_\gamma(v^TY)\|_2^2,$$
3. to obtain squared Cramer-Wold distance average, integrate the above formula over all possible $v\in S_D$.

The key theoretical outcome of this paper is that the computation of the Cramer-Wold distance can be simplified to a closed-form solution. Consequently, to compute the distance of two samples there is no need to find the optimal transport like in WAE, nor is it necessary to sample over the projections as in SWAE.

Theorem 1 *Let $X=(x_i)_{i=1..n}$, $Y=(y_j)_{j=1..n}\subset\mathbb{R}^D$ be given[4]. We formally define the squared Cramer-Wold distance with formula*

$$d^2_{\mathrm{cw}}(X,Y) := \int_{S_D} \|\mathrm{sm}_\gamma(v^TX)-\mathrm{sm}_\gamma(v^TY)\|_2^2\, d\sigma_D(v). \tag{2}$$

3. The choice of the optimal value of γ parameter is still a challenging problem. In our paper we use Silverman's rule of thumb since it works very well in practical applications. There are other equivalent rules although other rules also possible.
4. For clarity of presentation we provide here the formula for the case of samples of equal size.

Then

$$d^2_{\mathrm{cw}}(X,Y) = \frac{1}{2n^2\sqrt{\pi\gamma}}\left(\sum_{ii'}\phi_D\big(\tfrac{\|x_i-x_{i'}\|^2}{4\gamma}\big)+\sum_{jj'}\phi_D\big(\tfrac{\|y_j-y_{j'}\|^2}{4\gamma}\big)-2\sum_{ij}\phi_D\big(\tfrac{\|x_i-y_j\|^2}{4\gamma}\big)\right), \quad (3)$$

where $\phi_D(s) = {}_1F_1(\frac{1}{2};\frac{D}{2};-s)$ *and* ${}_1F_1$ *is the Kummer's confluent hypergeometric function (see, e.g., Barnard et al. (1998)). Moreover,* $\phi_D(s)$ *has the following asymptotic formula valid for* $D \geq 20$*:*

$$\phi_D(s) \approx (1+\tfrac{4s}{2D-3})^{-1/2}. \quad (4)$$

To prove Theorem 1 we need the following crucial technical proposition.

Proposition 2 *Let* $z \in \mathbb{R}^D$ *and* $\gamma > 0$ *be given. Then*

$$\int_{S_D} N(v^T z,\gamma)(0)\, d\sigma_D(v) = \frac{1}{\sqrt{2\pi\gamma}}\phi_D\left(\frac{\|z\|^2}{2\gamma}\right). \quad (5)$$

Proof By applying an orthonormal change of coordinates, without loss of generality, we may assume that $z = (z_1, 0, \ldots, 0)$, and then $v^T z = z_1 v_1$ for $v = (v_1, \ldots, v_D)$. Consequently, we get

$$\int_{S_D} N(v^T z,\gamma)(0)\, d\sigma_D(v) = \int_{S_D} N(z_1 v_1,\gamma)(0)\, d\sigma_D(v).$$

Making use of the formula for slice integration of functions on spheres (Axler et al., 1992, Corollary A.6) we get:

$$\int_{S_D} f\, d\sigma_D = \frac{V_{D-1}}{V_D}\int_{-1}^{1}(1-x^2)^{(D-3)/2}\cdot\int_{S_{D-1}} f(x,\sqrt{1-x^2}\,\zeta)\, d\sigma_{D-1}(\zeta)\, dx,$$

where V_K denotes the surface volume of sphere $S_K \subset \mathbb{R}^K$. Applying the above equality to function $f(v_1,\ldots,v_D) = N(z_1 v_1,\gamma)(0)$ and $s = z_1^2/(2\gamma) = \|z\|^2/(2\gamma)$, we consequently get that the LHS of (5) simplifies to

$$\frac{V_{D-1}}{V_D}\frac{1}{\sqrt{2\pi\gamma}}\int_{-1}^{1}(1-x^2)^{(D-3)/2}\exp(-sx^2)\, dx,$$

which completes the proof since $V_K = \frac{2\cdot\pi^{\frac{K}{2}}}{\Gamma(\frac{K}{2})}$ and

$$\int_{-1}^{1}\exp(-sx^2)(1-x^2)^{(D-3)/2}\, dx = \sqrt{\pi}\frac{\Gamma(\frac{D-1}{2})}{\Gamma(\frac{D}{2})}\,{}_1F_1\left(\tfrac{1}{2};\tfrac{D}{2};-s\right)$$

■

Proof [Proof of Theorem 1] Directly from the definition of smoothing we obtain that

$$d^2_{\mathrm{cw}}(X,Y) = \int_{S_D}\left\|\tfrac{1}{n}\textstyle\sum_i N(v^T x_i,\gamma) - \tfrac{1}{n}\sum_j N(v^T y_j,\gamma)\right\|_2^2 d\sigma_D(v). \quad (6)$$

Applying now the one-dimensional formula for the L_2 scalar product of two Gaussians:

$$\langle N(r_1,\gamma_1), N(r_2,\gamma_2)\rangle_2 = N(r_1-r_2,\gamma_1+\gamma_2)(0)$$

and the equality $\|f-g\|_2^2 = \langle f,f\rangle_2 + \langle g,g\rangle_2 - 2\langle f,g\rangle_2$ (where $\langle f,g\rangle_2 = \int f(x)g(x)dx$), we simplify the squared L_2 norm in the integral of RHS of (6) to

$$\|\tfrac{1}{n}\sum_i N(v^T x_i,\gamma) - \tfrac{1}{n}\sum_i N(v^T y_j,\gamma)\|_2^2 = \tfrac{1}{n^2}\langle\sum_i N(v^T x_i,\gamma), \sum_i N(v^T x_i,\gamma)\rangle_2 +$$
$$\tfrac{1}{n^2}\langle\sum_j N(v^T y_j,\gamma), \sum_j N(v^T y_j,\gamma)\rangle_2 - \tfrac{2}{n^2}\langle\sum_i N(v^T x_i,\gamma), \sum_j N(v^T y_j,\gamma)\rangle_2 =$$
$$\tfrac{1}{n^2}\sum_{ii'} N(v^T(x_i - x_{i'}),2\gamma)(0) + \tfrac{1}{n^2}\sum_{jj'} N(v^T(y_j - y_{j'}),2\gamma)(0) - \tfrac{2}{n^2}\sum_{ij} N(v^T(x_i - y_j),2\gamma)(0).$$

Applying proposition 2 directly, we obtain formula (3). Proof of the formula for the asymptotics of function ϕ_D is provided in the next section. ■

Therefore, to estimate the distance of a given sample X to some prior distribution f, one can follow the common approach and take the distance between X and a sample from f. As the main theoretical result of the paper we consider the following theorem, which states that in the case of standard Gaussian multivariate prior, we can completely reduce the need for sampling (we omit the proof since it is similar to that of Theorem 1).

Theorem 3 *Let $X = (x_i)_{i=1..n} \subset \mathbb{R}^D$ be a given sample. We formally define*

$$d_{\text{cw}}^2(X, N(0,I)) := \int_{S_D} \|\text{sm}_\gamma(v^T X) - \text{sm}_\gamma(N(0,1))\|_2^2 d\sigma_D(v).$$

Then

$$d_{\text{cw}}^2(X, N(0,I)) = \tfrac{1}{2n^2\sqrt{\pi}}\big(\tfrac{1}{\sqrt{\gamma}}\textstyle\sum_{i,j}\phi_D(\frac{\|x_i-x_j\|^2}{4\gamma}) + \frac{n^2}{\sqrt{1+\gamma}} - \frac{2n}{\sqrt{\gamma+\frac{1}{2}}}\sum_i \phi_D(\frac{\|x_i\|^2}{2+4\gamma})\big).$$

See Figure 2 for a comparison between Cramer-Wold, Wasserstein MMD, and the Sliced-Wasserstein distances with different data dimensions and sample sizes. In the experiment, we use two samples from Gaussian distribution $N([0,\dots,0]^T, I)$ and $N([\alpha,0,\dots,0]^T, I)$, where we change the parameter α in range $[0,6]$. Note that the Cramer-Wold distance is the lowest one irrespective of data dimension and sample size, and does not change much.

4. Computation of ϕ_D

As it was shown in the previous section, the key element of the Cramer-Wold distance is the function

$$\phi_D(s) = {}_1F_1(\tfrac{1}{2}; \tfrac{D}{2}; -s) \text{ for } s \geq 0.$$

Consequently, in this section we focus on the derivation of its basic properties. We provide its approximate asymptotic formula valid for dimensions $D \geq 20$, and then consider the special case of $D = 2$ (see Figure 3), where we provide the exact formula.

To do so, let us first recall (see Abramowitz and Stegun (1964, Chapter 13)) that the Kummer's confluent hypergeometric function ${}_1F_1$ (denoted also by M) has the following integral representation

$${}_1F_1(a,b,z) = \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)}\int_0^1 e^{zu}u^{a-1}(1-u)^{b-a-1}\,du,$$

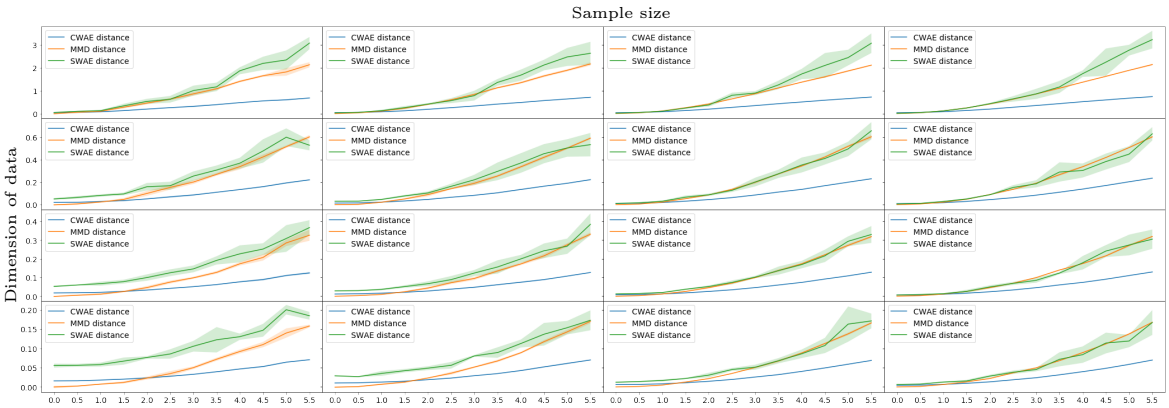

Figure 2: Comparison between Cramer-Wold, Wasserstein MMD, and Sliced-Wasserstein distances for different dimensions (from top to bottom for $10, 50, 100, 200$) and sample sizes (columns from left to right for $100, 200, 500, 1000$). In the experiment, we use two samples from Gaussians $N([0, \dots, 0]^T, I)$ and $N([\alpha, 0, \dots, 0]^T, I)$, where the parameter α of the mean shift is changed in range $[0, 6]$.

valid for $a, b > 0$ such that $b > a$. Since we consider that latent is at least of dimension $D \geq 2$, it follows that

$$\phi_D(s) = \frac{\Gamma(\frac{D}{2})}{\Gamma(\frac{1}{2})\Gamma(\frac{D}{2} - \frac{1}{2})} \int_0^1 e^{-su} u^{-1/2} (1-u)^{D/2-3/2} \, du.$$

By making a substitution $u = x^2$, $du = 2xdx$, we consequently get

$$\begin{aligned} \phi_D(s) &= \frac{2\Gamma(D/2)}{\Gamma(1/2)\Gamma(D/2-1/2)} \int_0^1 e^{-sx^2} (1-x^2)^{(D-3)/2} \, dx \\ &= \frac{\Gamma(D/2)}{\Gamma(1/2)\Gamma(D/2-1/2)} \int_{-1}^1 e^{-sx^2} (1-x^2)^{(D-3)/2} \, dx. \end{aligned} \tag{7}$$

Proposition 4 *For large*[5] D *we have*

$$\phi_D(s) \approx (1 + \tfrac{4s}{2D-3})^{-1/2} \text{ for all } s \geq 0. \tag{8}$$

Proof We have to estimate asymptotics of (7), i.e.

$$\phi_D(s) = \frac{\Gamma(\frac{D}{2})}{\Gamma(\frac{1}{2})\Gamma(\frac{D}{2} - \frac{1}{2})} \int_{-1}^1 e^{-sx^2} (1-x^2)^{(D-3)/2} \, dx.$$

Since for large D and all $x \in [-1, 1]$ we have

$$(1-x^2)^{(D-3)/2} e^{-sx^2} \approx (1-x^2)^{(D-3)/2} \cdot (1-x^2)^s = (1-x^2)^{s+(D-3)/2},$$

5. In practice we can take $D \geq 20$.

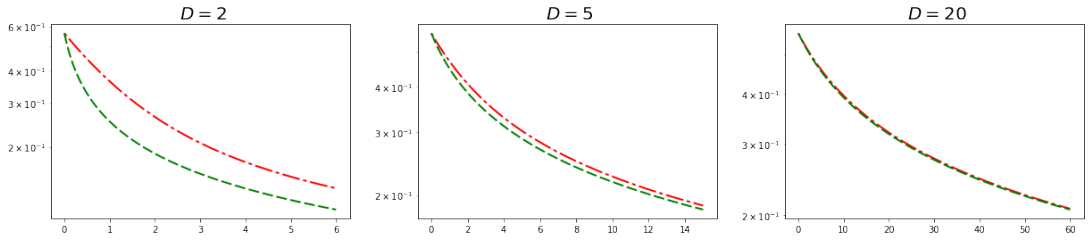

Figure 3: Comparison of ϕ_D value (red line) with the approximation given by (8) (green line) in the case of dimensions $D = 2, 5, 20$. Observe that for $D = 20$, the functions practically coincide.

we get

$$\phi_D(s) \approx \frac{\Gamma(\frac{D}{2})}{\Gamma(\frac{D-1}{2})\sqrt{\pi}} \cdot \int_{-1}^{1} (1-x^2)^{s+(D-3)/2}\, dx = \frac{\Gamma(\frac{D}{2})}{\Gamma(\frac{D-1}{2})\sqrt{\pi}} \cdot \sqrt{\pi}\frac{\Gamma(s+\frac{D}{2}-\frac{1}{2})}{\Gamma(s+\frac{D}{2})}.$$

To simplify the above, we apply the formula (1) from Tricomi and Erdélyi (1951):

$$\frac{\Gamma(z+\alpha)}{\Gamma(z+\beta)} = z^{\alpha-\beta}\left(1 + \frac{(\alpha-\beta)(\alpha+\beta-1)}{2z} + O(|z|^{-2})\right),$$

with α, β fixed so that $\alpha + \beta = 1$ (so only the error term of order $O(|z|^{-2})$ remains), and get the following

$$\begin{aligned} \Gamma(\tfrac{D}{2})/\Gamma(\tfrac{D-1}{2}) &= \frac{\Gamma((\frac{D}{2}-\frac{3}{4})+\frac{3}{4})}{\Gamma((\frac{D}{2}-\frac{3}{4})+\frac{1}{4})} \approx \left(\tfrac{D}{2}-\tfrac{3}{4}\right)^{\frac{1}{2}} \\ \Gamma(s+\tfrac{D}{2}-\tfrac{1}{2})/\Gamma(s+\tfrac{D}{2}) &\approx \left(s+\tfrac{D}{2}-\tfrac{3}{4}\right)^{-\frac{1}{2}}. \end{aligned} \tag{9}$$

Summarizing,

$$\phi_D(s) \approx \frac{(\frac{D}{2}-\frac{3}{4})^{1/2}}{(s+\frac{D}{2}-\frac{3}{4})^{1/2}} = (1+\tfrac{4s}{2D-3})^{-1/2}.$$

■

The above formula is valid for dimensions higher than 20. For lower dimensions we recommend using iterative direct formulas for ϕ_D function, which can be obtained using erf and modified Bessel functions of the first kind I_0 and I_1. To provide an example we consider here a special case of $D = 2$ since it is used in the paper for illustrative reasons in the latent for the MNIST data-set. As we have the equality (Gradshteyn and Ryzhik, 2015, (8.406.3) and (9.215.3))

$$\phi_2(s) = {}_1F_1(\tfrac{1}{2}, 1, -s) = e^{-\frac{s}{2}} I_0\left(\frac{s}{2}\right),$$

to practically implement ϕ_2 we apply the approximation of I_0 from Abramowitz and Stegun (1964, p. 378) given in the following remark.

Remark 5 *Let $s \geq 0$ be arbitrary and let $t = s/7.5$. Then*

$$\phi_2(s) \approx \begin{cases} e^{-\frac{s}{2}} \cdot (1+3.51562t^2+3.08994t^4+1.20675t^6+0.26597t^8+0.03608t^{10}+0.00458t^{12}) \\ \quad \text{for } s \in [0, 7.5], \\ \sqrt{\frac{2}{s}} \cdot (0.398942+0.013286t^{-1}+0.002253t^{-2}-0.001576t^{-3}+0.00916t^{-4}-0.020577t^{-5} \\ \quad +0.026355t^{-6}-0.016476t^{-7}+0.003924t^{-8}) \quad \text{for } s \geq 7.5. \end{cases}$$

5. Cramer-Wold kernel

In this section we first formally define the Cramer-Wold metric for arbitrary measures, and then show that it is given by a characteristic kernel which has a closed-form for spherical Gaussians. For more information on kernels, and kernel embedding of distributions, we refer the reader to Muandet et al. (2017).

Let us first introduce the general definition of the Cramer-Wold cw-metric. To do so we generalise the notion of smoothing for arbitrary measures μ with formula

$$\mathrm{sm}_\gamma(\mu) = \mu * N(0, \gamma I),$$

where $*$ denotes the convolution operator for two measures, and we identify the normal density $N(0, \gamma I)$ with the measure it introduces. It is well known that the resulting measure has the density given by

$$x \to \int N(x, \gamma I)(y) d\mu(y).$$

Clearly

$$\mathrm{sm}_\gamma(N(0, \alpha I)) = N(0, (\alpha + \gamma) I)).$$

Moreover, by applying the characteristic function one obtains that if the smoothing of two measures coincide, then the measures coincide too

$$\mathrm{sm}_\gamma(\mu) = \mathrm{sm}_\gamma(\mu) \implies \mu = \nu. \tag{10}$$

We also need to define the transport of the density by the projection $x \to v^T x$, where v is chosen from the unit sphere S_D. The definition is formulated so that if $\mathbf{X}$ is a random vector with density f, then f_v is the density of the random vector $\mathbf{X}_v := v^T \mathbf{X}$. Then

$$f_v(r) = \int_{y : y - rv \perp v} f(z) d_{D-1}(z),$$

where d_{D-1} denotes the $(D-1)$–dimensional Lebesgue measure. In general, if μ is a measure on $\mathbb{R}^D$, then μ_v is the measure defined on $\mathbb{R}$ by the formula

$$\mu_v(A) = \mu(\{x : v^T x \in A\}).$$

Since, if a random vector $\mathbf{X}$ has a density $N(a, \gamma I)$, and then the random variable $\mathbf{X}_v$ has the density $N(v^T a, \alpha)$, we may directly conclude that

$$N(a, \gamma I)_v = N(v^T a, \gamma).$$

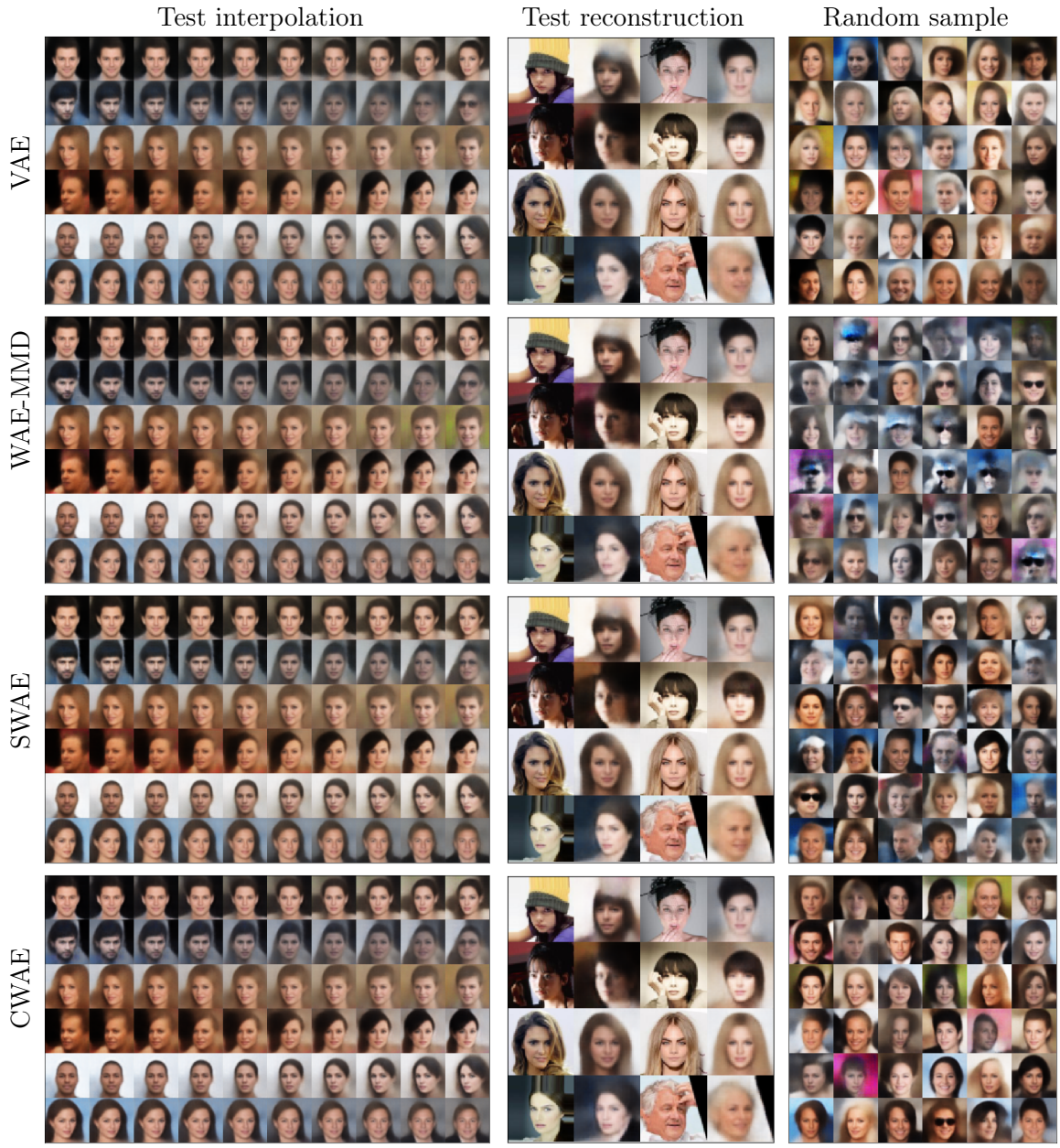

Figure 4: Results of VAE, WAE-MMD, SWAE, and CWAE models trained on CELEB A data-set using the WAE architecture from Tolstikhin et al. (2017). In "test reconstructions" odd rows correspond to the real test points.

It is also worth noting that due to the fact that the projection of a Gaussian is a Gaussian, the smoothing and projection operators commute, i.e.

$$\mathrm{sm}_\gamma(\mu_v) = (\mathrm{sm}_\gamma \mu)_v.$$

Given fixed $\gamma > 0$, the two above notions allow us to formally define the cw-distance of two measures μ and ν with formula

$$d^2_{\text{cw}}(\mu, \nu) = \int_{S_D} \|\text{sm}_\gamma(\mu_v) - \text{sm}_\gamma(\nu_v)\|^2_{L_2} d\sigma_D(v). \tag{11}$$

Observe that this implies that cw-distance is given by the kernel function

$$k(\mu, \nu) = \int_{S_D} \langle \text{sm}_\gamma(\mu_v), \text{sm}_\gamma(\nu_v) \rangle_{L_2} d\sigma_D(v).$$

Let us now prove that the function d_{cw}, defined by equation (11) is a metric, i.e. the kernel is characteristic).

Theorem 6 *Function d_{cw} is a metric.*

Proof Since d_{cw} comes from a scalar product, we only need to show that if the distance of two measures is zero, the measures coincide.

So let μ, ν be given measures such that $d_{\text{cw}}(\mu, \nu) = 0$. This implies that

$$\text{sm}_\gamma(\mu_v) = \text{sm}_\gamma(\nu_v).$$

By (10) this implies that $\mu_v = \nu_v$. Since this holds for all $v \in S_D$, by the Cramer-Wold Theorem we obtain that $\mu = \nu$. ■

We can summarize the above by saying that the Cramer-Wold kernel is a characteristic kernel which, by the definition and (5), has a closed-form of a scalar product of two radial Gaussians given by

$$\langle N(x, \alpha I), N(y, \beta I) \rangle_{\text{cw}} = \frac{1}{\sqrt{2\pi(\alpha+\beta+2\gamma)}} \phi_D \left(\frac{\|x-y\|^2}{2(\alpha+\beta+2\gamma)} \right). \tag{12}$$

Remark 7 *Observe, that except for the Gaussian kernel, it is the only kernel which has the closed–form for the spherical Gaussians. It is important since the RBF (Gaussian) kernels cannot be successfully applied in auto-encoder based generative models (we discuss it in the next section,). The reason is that the derivative of Gaussian vanishes quickly with distance; and therefore, it leads to difficulties in training as shown in (Tolstikhin et al., 2017, Section 4, WAE-GAN and WAE-MMD specifics).*

6. Cramer-Wold Auto-Encoder (CWAE)

In this section we construct an auto-encoder based on the Cramer-Wold distance. We start by introducing notation.

Auto-encoder. Let $X = (x_i)_{i=1..n} \subset \mathbb{R}^N$ be a given data-set. The basic aim of AE is to transport the data to a typically, but not necessarily, lower dimensional latent space $\mathcal{Z} = \mathbb{R}^D$ while minimizing the reconstruction error. Hence, we search for an encoder $\mathcal{E} : \mathbb{R}^n \to \mathcal{Z}$ and a decoder $\mathcal{D} : \mathcal{Z} \to \mathbb{R}^n$ functions that minimise the mean squared error $MSE(X; \mathcal{E}, \mathcal{D})$ on X and its reconstructions $\mathcal{D}(\mathcal{E}x_i)$.

Auto-encoder based generative model. CWAE, similarly to WAE, is a an auto-encoder model with modified cost function which forces the model to be generative, i.e. ensures that the data transported to the latent space comes from the prior distribution (typically Gaussian). This statement is formalized by the following important remark, see also Tolstikhin et al. (2017).

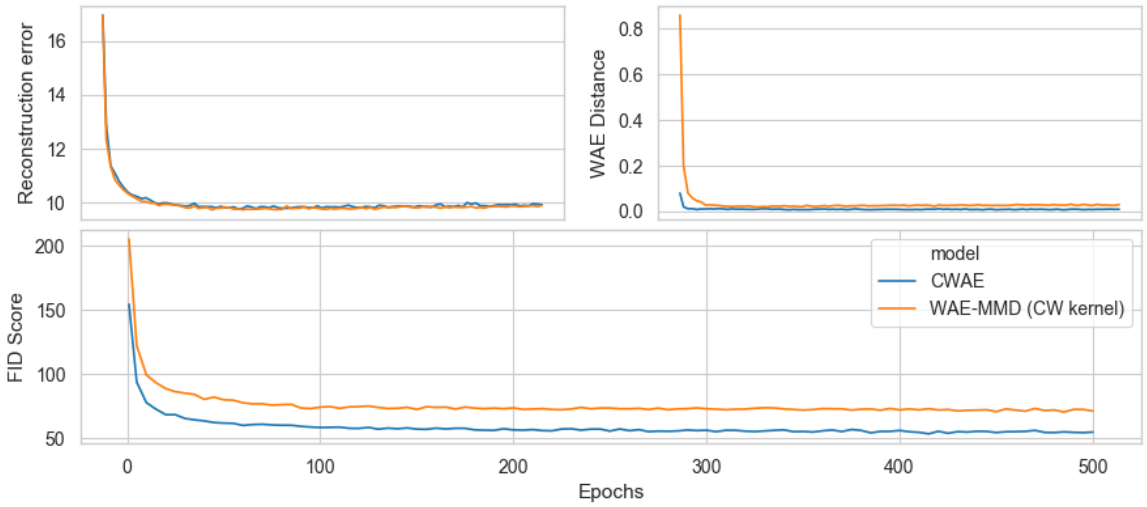

Figure 5: Comparison between CWAE and WAE-MMD with CW kernel on Fashion-MNIST data-set.

Remark 8 *Let* $\mathbf{X}$ *be an* N*-dimensional random vector, from which our data-set was drawn, and let* $\mathbf{Y}$ *be a random vector with distribution* f *on latent* $\mathcal{Z}$*.*

Suppose that we have constructed functions $\mathcal{E} : \mathbb{R}^N \to \mathcal{Z}$ *and* $\mathcal{D} : \mathcal{Z} \to \mathbb{R}^N$ *(representing the encoder and decoder pair) such that*[6]

1. $\mathcal{D}(\mathcal{E}x) = x$ *for* $x \in \text{image}(\mathbf{X})$*,*
2. *random vector* $\mathcal{E}\mathbf{X}$ *has the distribution* f*.*

Then by the point 1 we obtain that $\mathcal{D}(\mathcal{E}\mathbf{X}) = \mathbf{X}$*, therefore*

$$\mathcal{D}\mathbf{Y} \text{ has the same distribution as } \mathcal{D}(\mathcal{E}\mathbf{X}) = \mathbf{X}.$$

This means that to produce samples from $\mathbf{X}$ *we can instead produce samples from* $\mathbf{Y}$ *and map them by the decoder* $\mathcal{D}$*.*

Since an estimator of the image of the random vector $\mathbf{X}$ is given by its sample X, we conclude that a generative model is correct if it has a small reconstruction error and resembles the prior distribution in the latent. Thus, to construct a generative auto-encoder model (with Gaussian prior), we add to its cost function a measure of the distance of a given sample from a normal distribution.

6. We recall that for function (or in particular random vector) $\mathbf{X} : \Omega \to \mathbb{R}^D$, by $\text{image}(\mathbf{X})$ we denote the set consisting of all possible values $\mathbf{X}$ can attain, i.e. $\{\mathbf{X}(\omega) : \omega \in \Omega\}$.

CWAE cost function. Once the crucial ingredient of CWAE is ready, we can describe its cost function. To ensure that the data transported to latent $\mathcal{Z}$ are distributed according to standard normal density distribution, we can add the Cramer-Wold distance $d^2_{\text{cw}}(X, N(0, I))$ from standard multivariate normal density to the cost function:

$$\text{cost}(X; \mathcal{E}, D) = MSE(X; \mathcal{E}, \mathcal{D}) + \lambda d^2_{\text{cw}}(\mathcal{E}X, N(0, I))$$

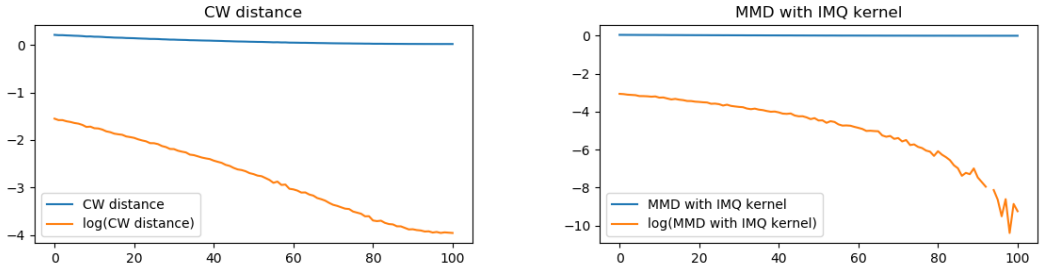

Figure 6: Synthetic data in the latent and the distance from prior cost: the CWAE model on the left, WAE-MMD on the right. On the horizontal axis there is the share of $z \sim N(0, 1)$ in uniform data. The blue curves represent a standard model (without logarithm), while the orange denote the one with a logarithm used.

Since the use of special functions involved in the formula for Cramer-Wold distance might be cumbersome, in all experiments (except the illustrative 2D case) we apply the asymptotic form (8) of function ϕ_D:

$$2\sqrt{\pi} d^2_{\text{cw}}(X) \approx \frac{1}{n^2} \sum_{ij} (\gamma_n + \frac{\|x_i - x_j\|^2}{2D-3})^{-1/2} + (1 + \gamma_n)^{-1/2} - \frac{2}{n} \sum_i (\gamma_n + \frac{1}{2} + \frac{\|x_i\|^2}{2D-3})^{-1/2},$$

where $\gamma_n = (\frac{4}{3n})^{2/5}$ is chosen using Silverman's rule of thumb (Silverman, 1986).

Such a solution can be understood as a use of WAE-MMD with a Cramer-Wold kernel. In CWAE model, we use a logarithm function:

$$\text{cost}(X; \mathcal{E}, D) = MSE(X; \mathcal{E}, \mathcal{D}) + \lambda \log d^2_{\text{cw}}(\mathcal{E}X, N(0, I)).$$

CWAE cost differs from the WAE model cost, by the utilisation of a logarithm function. We observed that using a logarithm to scale the second term increased training speed, as shown in Figure 5.

During the first few starting iterations it is typical for the errors' variation to be high. In case of CWAE, the D_{cw} cost is around 10 times larger than the d_k cost of WAE. The logarithm can tone it substantially, increasing the stability of learning, which is not needed in WAE. The network finds a smoother way to increase the normality of the latent space, thus speeding up training process.

At the same time it is probable that at the beginning of training, the distribution of example projections in the latent space is more uniform. Then, with training progression it tends to become closer to a normal distribution (assuming a normal prior). A synthetic data experiment showing this phenomenon is given in Figure 6. The logarithmic cost drops-off much quicker pulling the model towards quicker minimization.

On the other hand, a modification of WAE-MMD with cost $\cdots + d_k^2(\cdot,\cdot)$ (see Eq. (13)) to $\cdots + \log d_k^2(\cdot,\cdot)$ (in Eq. (14)) results in a steeper and more irregular descent. The WAE-MMD cost is closer to zero, and may sometimes be even negative as noted in Tolstikhin (2018, "...penalty used in WAE-MMD is not precisely the population MMD, but a sample based U-statistic...if the population MMD is zero, it necessarily needs to take negative values from time to time."). Therefore the log version is not suitable for the WAE-MMD version, which coincides with experiments.

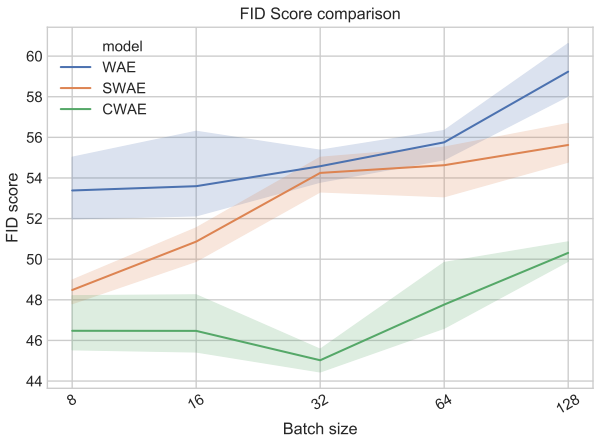

Figure 7: Comparison between WAE, SWAE and CWAE with respect to batch size. We repeated the experiment five times, confidence intervals represent the standard deviation.

The use of Cramer-Wold distance and a logarithm in cost function allows us to construct more stable models. More precisely, the cost function is less sensitive to the changes of training parameters like batch size and learning rate, see Figure 7. As a consequence, in practical applications the CWAE model is easier to train.

Comparison with WAE and SWAE models. Finally, let us briefly recapitulate the differences between the introduced CWAE model, WAE variants (Tolstikhin et al., 2017) and SWAE (Kolouri et al., 2018).

Firstly, from the theoretical point of view both SWAE and CWAE models use similar distances d_{sw} and d_{cw}, obtained by integration over all 1-dimensional projections (compare Eqs (1) and (2)). On the other hand, SWAE incorporates Wasserstein distances under the integral, while in CWAE, under the integral, the L_2 distances between regularizations are used. Additionally, the integral in the d_{sw} formula is estimated with a finite sum, while for d_{cw} we obtain analytically quite accurate approximate formula.

From a computational point of view, it is important that in contrast to WAE-MMD and SWAE, the CWAE model *does not* require sampling from normal distribution (as it is the case in WAE-MMD) or over slices (as in SWAE) to evaluate its cost function. In this sense, CWAE uses a closed formula cost function. In contrast to WAE-GAN, our objective does not require a separately trained neural network to approximate the optimal transport function, thus avoiding pitfalls of adversarial training.

Comparison with WAE-MMD models. We now compare the proposed CWAE model to WAE-MMD. In particular, we show that CWAE can be seen as a combination of the sliced-approach with the MMD-based models. The WAE-MMD model uses approximations, while CWAE uses a closed-form, which has an impact on training. It results in a more leveled drop of distance weight, with even negative values in case of a WAE-MMD estimator, see Figure 6.

Since both WAE and CWAE use kernels to discriminate between sample and normal density distribution to compare the models we first describe the WAE model. WAE cost function for a given characteristic kernel k and sample $X = (x_i)_{i=1..n} \subset \mathbb{R}^D$ (in the D-dimensional latent) is given by

$$\text{WAE cost} = MSE + \lambda \cdot d_k^2(X, Y), \tag{13}$$

where $Y = (y_i)_{i=1..n}$ is a sample from the standard normal density $N(0, I)$, and $d_k^2(X, Y)$ denotes the kernel-based distance between the probability distributions representing X and Y, that is $\frac{1}{n}\sum_i \delta_{x_i}$ and $\frac{1}{n}\sum_i \delta_{y_i}$, where δ_z denotes the atom Dirac measure at $z \in \mathbb{R}^D$. The inverse multi-quadratic kernel IMQ k is chosen as default

$$k(x, y) = \frac{C}{C + \|x - y\|_2^2},$$

where in experiments in Tolstikhin et al. (2017) a value $C = 2D\sigma^2$ was used, while σ is a hyper-parameter denoting the standard deviation of the normal density distribution. Therefore, the model has hyper-parameters λ and σ, which were chosen to be $\lambda = 10, \sigma^2 = 1$ in MNIST, $\lambda = 100, \sigma^2 = 2$ in CELEB A.

On the other hand, the CWAE cost function for a sample $X = (x_i)_{i=1..n} \subset \mathbb{R}^D$ (in the D-dimensional latent) is given by

$$\text{CWAE cost} = MSE + \lambda \log d_{\text{cw}}^2(X, N(0, I)), \tag{14}$$

where the distance between the sample and the standard normal distribution is taken with respect to the Cramer-Wold kernel with a regularizing hyper-parameter γ, given by the Silverman's rule of thumb (the motivation for such a choice of hyper-parameters is explained in Section 3).

We stress the following important differences

- Due to the properties of the Cramer-Wold kernel, we are able to substitute the sample estimation of $d_k^2(X, N(0, I))$ given in WAE-MMD by $d_{\text{cw}}^2(X, Y)$ by its exact formula.
- CWAE, as compared to WAE, is less sensitive to the choice of parameters:
    1. The choice of regularization hyper-parameter is given by the Silverman's rule of thumb and depends on the sample size (contrary to WAE-MMD, where the hyper-parameters are chosen by hand, and in general do not depend on the sample size).
    2. In our preliminary experiments, we have observed that frequently (like in the case of log-likelihood) taking the logarithm of the non-negative factors of the cost function, which we aim to minimise to zero, improves the learning. Motivated by the above and the CWAE cost function analysis, the CWAE cost uses logarithm of the Cramer-Wold distance to balance the MSE and the divergence terms. It turned out that in most cases it is enough to set in Eq. (14) the parameter $\lambda = 1$. Furthermore, we show (see Figure 7) that CWAE is less sensitive in respect to batch size. For every batch size and model we performed a grid search over

$\lambda \in \{1, 10, 100\}$ and learning rate values in $\{0.01, 0.001, 0.0001\}$. For every model, we selected a configuration with the lowest FID score and repeated the experiment five times. As we can see, CWAE seems to be insensitive to this parameter.

Summarizing, the CWAE model, contrary to WAE-MMD, is less sensitive to the choice of parameters. Moreover, since we do not have the noise in the learning process given by the random choice of the sample Y from $N(0, I)$, the learning should be more stable. As a consequence, CWAE generally learns faster than WAE-MMD, and has smaller standard deviation of the cost-function during the learning process. Detailed results of the experiments for CELEB A data-set are presented in Figure 8. Moreover, for better comparison, we verified how the learning process looks like in the case of original WAE-MMD architecture form (Tolstikhin et al., 2017), see Figure 8.

Generalised Cramer-Wold kernel. In this paragraph, we show that asymptotically, with respect to dimension D, Cauchy kernel used in WAE-MMD can in fact be seen as the sliced kernel. We use two-dimensional subspaces as slices. To do so we need the probability measure on d-dimensional linear subspaces of $\mathbb{R}^D$, see Mattila (1999). One can do it either directly with the definition of a Grassmanian, or describe it with the orthonormal basis for integration over orthonormal matrices (Aubert and Lam, 2003; Braun, 2006).

Now we define the d-dimensional sliced Cramer-Wold kernel by the formula

$$k_d(\mu, \nu) = \int_{G(d,D)} \langle \mathrm{sm}_\gamma(\mu_v), \mathrm{sm}_\gamma(\nu_v) \rangle_{L_2} d\gamma_{d,D}(v),$$

where $\gamma_{d,D}$ denotes the respective Radon probability measure on $G(d, D)$. Equivalently, we can integrate over orthonormal sequences in $\mathbb{R}^D$ of length d:

$$O_d(\mathbb{R}^D) = \{(v_1, \ldots, v_d) \in (\mathbb{R}^D)^d : \|v_i\| = 1, v_i \perp v_j\}.$$

The normalised, invariant with respect to orthonormal transformations, measure on O_d we denote with θ_d. Observe that for $d = 1$ we obtain normalised integration over the sphere.

Then we obtain that k_d can be equivalently defined as

$$k_d(\mu, \nu) = \int_{O_d} \langle \mathrm{sm}_\gamma(\mu_v), \mathrm{sm}_\gamma(\nu_v) \rangle_{L_2} d\theta_d(v).$$

Let us first observe that for Gaussian densities the formula for k_d can be slightly simplified

$$\begin{aligned} k_d(N(x, \alpha I), N(y, \beta I)) &= \textstyle\int_{O_d} N(v^T(x-y), (\alpha+\beta+2\gamma) I_d)(0) d\theta_d(v) \\ &= \textstyle\int_{O_d} \prod_{i=1}^{d} N(v_i^T(x-y), \alpha+\beta+2\gamma)(0) d\theta_d(v). \end{aligned}$$

Now if we define

$$\Phi_D^d(s, h) = \int_{O_d} N(v^T s e_1, h I_d)(0) d\theta_d(v),$$

where $e_1 \in \mathbb{R}^D$ is the first unit base vector, we obtain that the kernel-product reduces to computation of the scalar function Φ_D

$$k_d(N(x, \alpha I), N(y, \beta I)) = \Phi_D^d(\|x - y\|, \alpha + \beta + 2\gamma).$$

The crucial observation needed to proceed further is that the measure space $(O_d(\mathbb{R}^D), \theta_d)$ can be approximated by $(\mathbb{R}^D, N(0, I/D))^d$. This follows from the fact, that if $v_1, \ldots, v_d$ are drawn from the density $N(0, I/D)$, then for sufficiently large D we have $\|v_i\| \approx 1$ and $\langle v_i, v_j \rangle \approx 0$ for $i \neq j$.

Theorem 9 *We have*

$$\Phi_D^d(s,h) \to (2\pi)^{-d/2} \cdot (h + s^2/D)^{-d/2}.$$

Proof By the observation stated before the theorem, we have

$$\begin{aligned}\Phi_D^d(s,h) &= \int_{O_d} \prod_{i=1}^{d} N(v_i^T s e_1, h)(0) d\theta_d(v) \approx \int\limits_{(\mathbb{R}^D)^d} \prod_{i=1}^{d} N(v_i^T s e_1, h)(0) N(0, \tfrac{I}{D})(v_i) dv_1 \ldots dv_d \\ &= \prod_{i=1}^{d} \int\limits_{\mathbb{R}^D} N(v_i^T s e_1, h)(0) N(0, \tfrac{I}{D})(v_i) dv_i.\end{aligned}$$

It thus suffices to compute each component of the above formula. To do so, we denote by N_k the k-dimensional normal density, and get

$$\begin{aligned}\int_{\mathbb{R}^D} N_1(s\langle v, e_1\rangle, h)(0) \cdot N_D\left(0, \tfrac{1}{D} I\right)(v) dv = \\ &= \int_{-\infty}^{\infty} N_1(0,h)(st) \frac{N_D(0, \frac{1}{D} I)(te_1)}{N_{D-1}(0, \frac{1}{D} I)(0)} \int_{\mathbb{R}^{D-1}} N_{D-1}(0, \tfrac{1}{D} I)(w) dw\, dt \\ &= \int_{-\infty}^{\infty} N_1(0,h)(st) \frac{N_D(0, \frac{1}{D} I)(te_1)}{N_{D-1}(0, \frac{1}{D} I)} dt \\ &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi h}} \exp(-\tfrac{1}{2h}(st)^2) \cdot \frac{\sqrt{D}}{\sqrt{2\pi}} \exp(-\tfrac{1}{2} D t^2) dt \sqrt{2\pi} \sqrt{\frac{h}{s^2 + hD}} \\ &= \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{h + s^2/D}},\end{aligned}$$

which yields the assertion of the theorem. ■

As a direct consequence, we obtain the following asymptotic formula (with the dimension D large) of the generalised Cramer-Wold kernel of two spherical Gaussians:

$$k_d(N(x, \alpha I), N(y, \beta I)) \approx (2\pi)^{-d/2} \cdot (\alpha + \beta + 2\gamma + \|x - y\|^2/D)^{-d/2}.$$

Observe, that for $d = 2$ we obtain the standard inverse multiquadratic kernel.

7. Experiments

In this section we empirically validate the proposed CWAE[7] model on standard benchmarks for generative models CELEB A, CIFAR-10, MNIST, and FashionMNIST. We compare the proposed CWAE model with WAE-MMD (Tolstikhin et al., 2017) and SWAE (Kolouri et al., 2018). As we shall see, our results match, or even exceed, those of WAE-MMD and SWAE, while using a closed-form cost function (see previous sections for a more detailed discussion). The rest of this section is structured as follows. In Subsection 7.2 we report the

7. The code is available at `https://github.com/gmum/cwae`.

results of the standard qualitative tests, as well as visual investigations of the latent space. In Subsection 7.3 we will turn our attention to quantitative tests using Fréchet Inception Distance and other metrics (Heusel et al., 2017). In Subsection 8 we provide a proof of concept for an application of the Cramer-Wold distance in the framework introduced by Deshpande et al. (2018).

7.1. Experimentation setup

In the experiment we use two basic architecture types. Experiments on MNIST and Fashion-MNIST use a feed-forward network for both encoder and decoder, and an 8 neuron latent layer, all using ReLU activations. For CIFAR-10, and CELEB A data-sets we use convolution-deconvolution architectures. Please refer to Section 7.5 for full details.

Table 1: Comparison of different architectures on the MNIST, Fashion-MNIST, CIFAR-10 and CELEB A data-sets. All models outputs except AE are similarly close to the normal distribution. CWAE achieves the best value of FID score (lower is better). All hyper-parameters were found using a grid search (see section 7.5).

Data-set	Method	Learning rate	λ	Skewness	Kurtosis (normalised)	Rec. error	FID score
MNIST	AE	0.001	-	1197.24	878.07	11.19	52.74
	VAE	0.001	-	0.43	0.77	18.79	40.47
	SWAE	0.001	1.0	6.01	10.72	10.99	29.76
	WAE-MMD	0.0005	1.0	11.70	8.34	11.14	27.65
	CWAE	0.001	1.0	12.21	35.88	11.25	**23.63**
FASHION MNIST	AE	0.001	-	140.21	85.58	9.87	81.98
	VAE	0.001	-	0.20	4.86	15.41	64.98
	SWAE	0.001	100.0	1.15	18.14	10.56	54.48
	WAE-MMD	0.001	100.0	2.82	4.33	10.01	58.79
	CWAE	0.001	10.0	5.11	65.96	10.36	**49.95**
CIFAR10	AE	0.001	-	2.5×10^5	1.7×10^4	24.67	269.09
	VAE	0.001	-	35.81	3.67	63.77	172.39
	SWAE	0.001	1.0	517.32	121.17	25.42	141.91
	WAE-MMD	0.001	1.0	1105.73	2097.14	25.04	129.37
	CWAE	0.001	1.0	176.60	1796.66	25.93	**120.02**
CELEB A	AE	0.001	-	4.6×10^9	2.6×10^8	86.41	353.50
	VAE	0.001	-	43.72	171.66	110.87	60.85
	SWAE	0.0001	100.0	141.17	222.02	85.97	53.85
	WAE-MMD	0.0005	100.0	162.67	604.09	86.38	51.51
	CWAE	0.0005	5.0	130.08	542.42	86.89	**49.69**

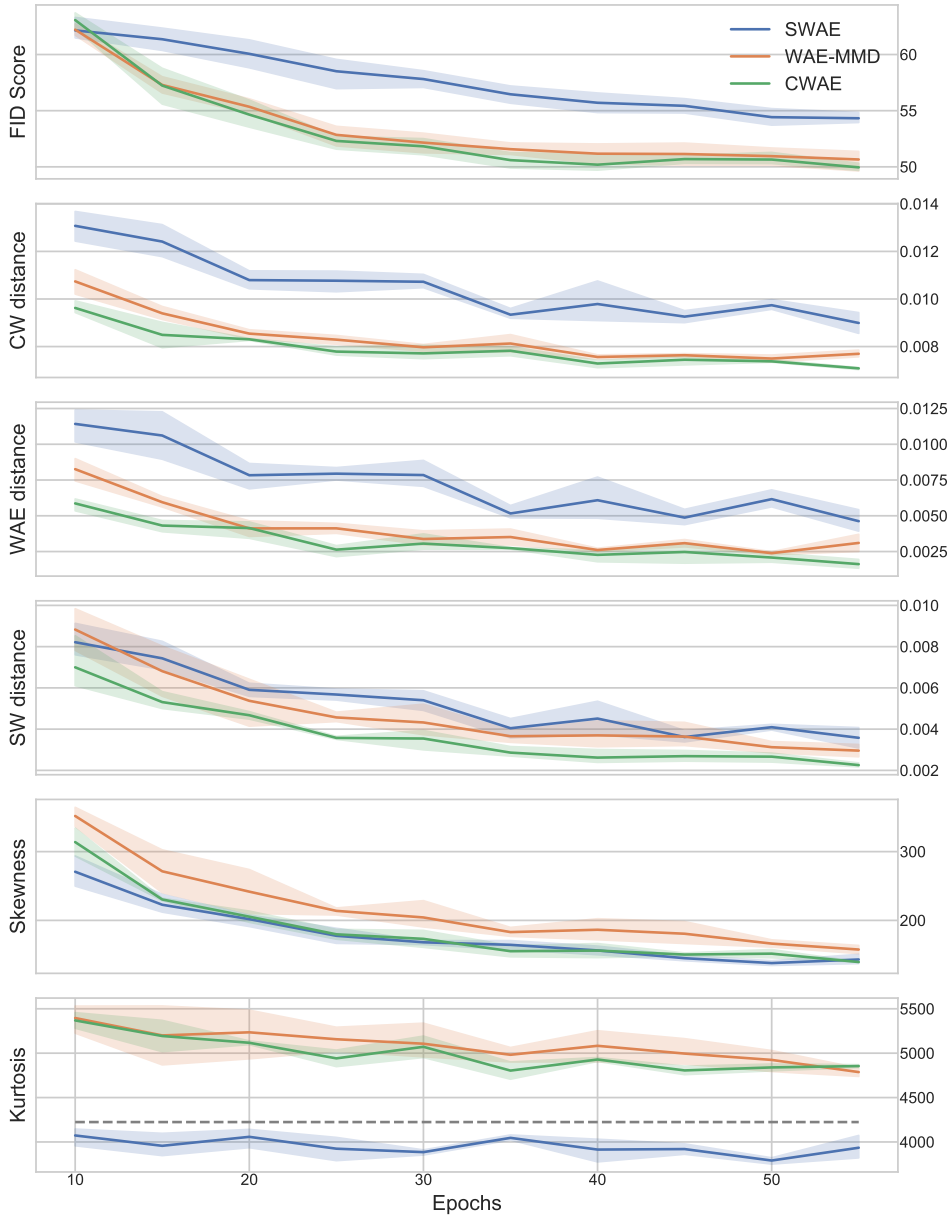

Figure 8: CELEB A trained CWAE, WAE, and SWAE models with FID score, kurtosis and skewness, as well as CW-, WAE-, and SWAE-distances on the original WAE-MMD architecture from Tolstikhin et al. (2017). All values are the averages from 5 models trained for each architecture. Confidence intervals represent the standard deviation. Optimum kurtosis is marked with a dashed line.

7.2. Qualitative tests

The quality of a generative model is typically evaluated by examining generated samples or by interpolating between samples in the latent space. We present such a comparison between CWAE with WAE-MMD in Figure 4. We follow the same procedure as in Tolstikhin et al. (2017). In particular, we use the same base neural architecture for both CWAE and WAE-MMD. For each model we consider (i) interpolation between two random examples from the test set (leftmost in Figure 4), (ii) reconstruction of a random example from the test set (middle column in Figure 4), and finally a sample reconstructed from a random point sampled from the prior distribution (right column in Figure 4). The experiment shows that there are no perceptual differences between CWAE and WAE-MMD generative distribution.

In the next experiment we qualitatively assess the normality of the latent space. This will allow us to ensure that CWAE does not compromise on the normality of its latent distribution, which is a part of the cost function for all the models except AE. We compare CWAE[8] with VAE, WAE and SWAE on the MNIST data using 2–dimensional latent space and a two-dimensional Gaussian prior distribution. Results are reported in Figure 9. As is readily visible, the latent distribution of CWAE is as close, or perhaps even closer, to the normal distribution than that of the other models.

To summarize, both in terms of perceptual quality and satisfying normality objective, CWAE matches WAE-MMD. The next section will provide more quantitative studies.

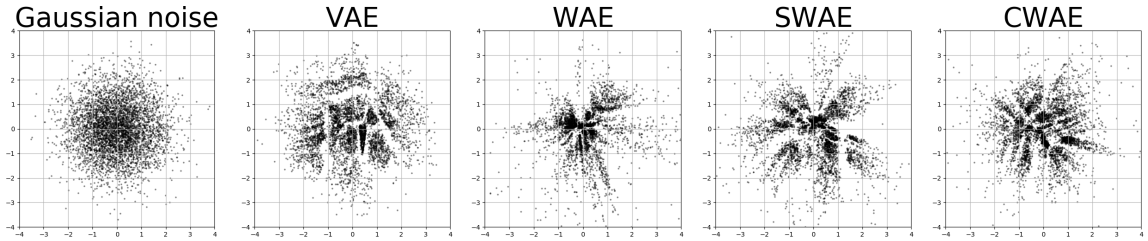

Figure 9: The latent distribution of CWAE is close to the normal distribution. Each subfigure presents points sampled from two-dimensional latent spaces, VAE, WAE, SWAE, and CWAE (left to right). All trained on the MNIST data-set.

7.3. Quantitative tests

In order to quantitatively compare CWAE with other models, in the first experiment we follow the experimental setting and use the same architecture as in Tolstikhin et al. (2017). In particular, we employ the Fréchet Inception Distance (FID) (Heusel et al., 2017).

In agreement with the qualitative studies, we observe FID of CWAE to be similar or slightly better than WAE-MMD. We highlight that CWAE on CELEB A achieves 49.69 FID

8. Since (4) is valid for dimensions $D \geq 20$, to implement CWAE in 2-dimensional latent space we apply equality ${}_1F_1(1/2, 1, -s) = e^{-\frac{s}{2}} I_0\left(\frac{s}{2}\right)$ jointly with the approximate formula (Abramowitz and Stegun, 1964, p. 378) for the Bessel function of the first kind I_0.

Table 2: Comparison between classical cost function of WAE, SWAE and a version with a logarithm (method names with a -LOG suffix).

Data-set	Method	Learning rate	λ	Skewness	Kurtosis (normalised)	Rec. error	FID score
MNIST	SWAE	0.001	1.0	6.01	10.72	10.99	29.76
	SWAE-LOG	0.0005	1.0	2.36	12.20	11.42	24.89
	WAE	0.0005	1.0	11.70	8.34	11.14	27.65
	WAE-LOG	0.001	1.0	18.22	61.04	13.17	36.08
	CWAE	0.001	1.0	12.21	35.88	11.25	**23.63**
FASHION MNIST	SWAE	0.001	100.0	1.15	18.14	10.56	54.48
	SWAE-LOG	0.001	10.00	11.01	0.88	14.11	55.17
	WAE	0.001	100.0	2.82	4.33	10.01	58.79
	WAE-LOG	0.005	1.0	53.37	66.01	16.14	99.51
	CWAE	0.001	10.0	5.11	65.96	10.36	**49.95**
CIFAR10	SWAE	0.001	1.0	517.32	121.17	25.42	141.91
	SWAE-LOG	0.0005	1.0	157.14	234.52	26.25	**119.89**
	WAE	0.001	1.0	1105.73	2097.14	25.04	129.37
	WAE-LOG	0.001	1.0	1.1×10^8	4.9×10^5	28.25	136.25
	CWAE	0.001	1.0	176.60	1796.66	25.93	120.02
CELEB A	SWAE	0.0001	100.0	141.17	222.02	85.97	53.85
	SWAE-LOG	0.0005	10.0	132.54	465.39	85.82	53.46
	WAE	0.0005	100.0	162.67	604.09	86.38	51.51
	WAE-LOG	0.0001	1.0	514.43	2154.39	82.53	58.10
	CWAE	0.0005	5.0	130.08	542.42	86.89	**49.69**

score compared to 51.51 and 53.85 achieved by WAE-MMD and SWAE, respectively, see Figure 8 and Table 1.

Next, motivated by Remark 8, we propose a novel method for quantitative assessment of the models based on their comparison to standard normal distribution in the latent. To achieve this we decided to use one of the most popular statistical normality tests, i.e. Mardia tests (Henze, 2002). Mardia's normality tests are based on verifying whether the skewness $b_{1,D}(\cdot)$ and kurtosis $b_{2,D}(\cdot)$ of a sample $X = (x_i)_{i=1..n} \subset \mathbb{R}^D$

$$b_{1,D}(X) = \frac{1}{n^2}\sum_{j,k}(x_j^T x_k)^3, \text{ and } b_{2,D}(X) = \frac{1}{n}\sum_j \|x_j\|^4$$

are close to that of standard normal density. The expected Mardia's skewness and kurtosis for the standard multivariate normal distribution are 0 and $D(D+2)$, respectively. To enable easier comparison in experiments we consider also the value of the normalised Mardia's kurtosis given by $b_{2,D}(X) - D(D+2)$, which equals zero for the standard normal density.

Results are presented in Figure 8 and Table 1. In Figure 8 we report for CELEB A data-set the value of FID score, Mardia's skewness and kurtosis during learning process of WAE, SWAE and CWAE (measured on the validation data-set).

WAE, SWAE and CWAE models obtain the best reconstruction error, comparable to AE. VAE model exhibits a slightly worse reconstruction error, but values of kurtosis and skewness indicating their output are closer to normal distribution. As expected, the output of AE is far from normal distribution; its kurtosis and skewness grow during learning. This arguably less standard evaluation, which we hope will find its way to being adapted by the community, serves as yet another evidence that *CWAE has strong generative capabilities, which at least match the performance of WAE-MMD.* Moreover, we observe that VAE model's output distribution is closest to the normal distribution, at the expense of the reconstruction error, which is reflected by some blurred reconstructions, typically associated with that approach.

At the end of this subsection we compare our method with classical approaches WAE-MMD and SWAE with modified cost function. More precisely, similarly to CWAE we use logarithm in cost function in WAE and SWAE, see Table 2.

At it was mentioned, adding logarithm to WAE-MMD does not work, since penalty used in WAE-MMD is not precisely the population MMD, but a sample-based U-statistic. In consequence, cost function can be negative from time to time. Therefore, the log version is not suitable for the WAE-MMD version. On the other hand, logarithm improves learning process in case of CWAE and SWAE.

7.4. Comparison of the learning speed

We expect our closed-form formula to lead to a speedup in training time. Indeed, we found that for batch-sizes up to 1024 CWAE is faster (in terms of time needed per batch) than other models. More precisely, CWAE is approximately $2\times$ faster up to 256 batch-size.

Figure 10 gives a comparison of mean learning time for different most frequently used batch-sizes. Time spent on processing a batch is smaller for CWAE for a practical range of batch-sizes $[32, 512]$. For batch-sizes larger than 1024, CWAE is slower due to its quadratic complexity with respect to the batch-size. However, we note that batch-sizes even larger than 512 are relatively rarely used in practice for training auto-encoders.

7.5. Architecture

For WAE, SWAE and CWAE models and each data-set we performed a grid search over parameter $\lambda \in \{1, 5, 10, 100\}$ and learning rate values from $\{0.01, 0.005, 0.001, 0.0005, 0.0001\}$. For VAE and AE models, we examined only different learning rates. All models were trained on a 128-element mini-batches. For every model, we report results for a configuration that achieved the lowest value of FID Score.

MNIST/Fashion-MNIST (28×28 images): an encoder-decoder feed-forward architecture:

encoder three feed-forward ReLU layers, 200 neurons each

latent 8-dimensional,

decoder three feed-forward ReLU layers, 200 neurons each.

CIFAR-10 data-set ($32 \times$ images with three color layers): a convolution-deconvolution network

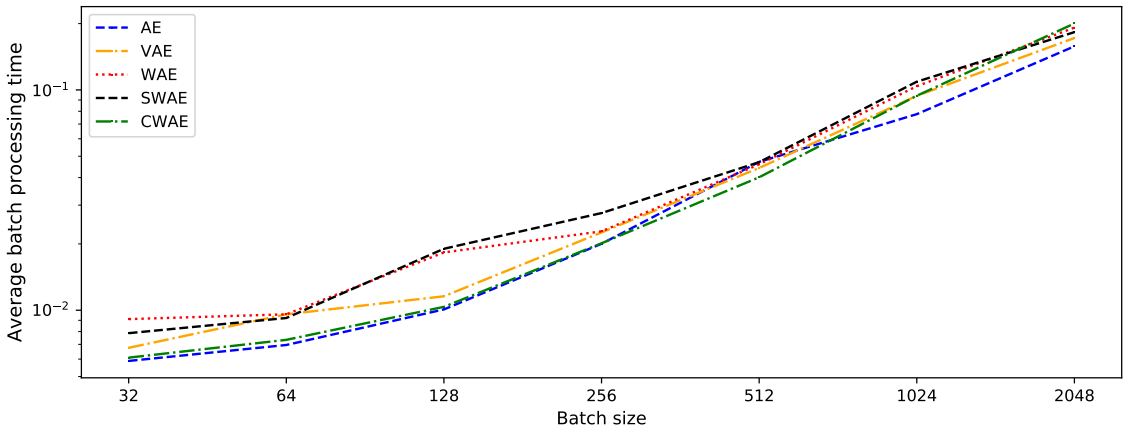

Figure 10: Comparison of mean batch learning time (times are in log-scale) for different algorithms in seconds. All for the same architecture introduced in Tolstikhin et al. (2017) and all requiring a similar number of epochs to train the full model. The times may differ for computer architectures with more/less memory on a GPU card (minimal value in our experiments was about 6×10^{-3}).

encoder

four convolution layers with 2×2 filters, the second one with 2×2 strides, other non-strided (3, 32, 32, and 32 channels) with ReLU activation,

128 ReLU neurons dense layer,

latent 64-dimensional,

decoder

two dense *ReLU* layers with 128 and 8192 neurons,

two transposed-convolution layers with 2×2 filters (32 and 32 channels) and ReLU activation,

a transposed convolution layer with 3×3 filter and 2×2 strides (32 channels) and ReLU activation,

a transposed convolution layer with 2×2 filter (3 channels) and sigmoid activation.

CELEB A (with images centered and cropped to 64×64 with 3 color layers): in order to have a direct comparison to WAE-MMD model on CELEB A, an identical architecture was used as that in Tolstikhin et al. (2017) utilised for the WAE-MMD model (WAE-GAN architecture is, naturally, different):

encoder

four convolution layers with 5×5 filters, each layer was followed by a batch normalization (consecutively 128, 256, 512, and 1024 channels) and ReLU activation,

latent 64-dimensional,

decoder

dense 1024 neuron layer,

three transposed-convolution layers with 5×5 filters, and each layer followed by a batch normalization with ReLU activation (consecutively 512, 256, and 128 channels),

transposed-convolution layer with 5×5 filter and 3 channels, clipped output value.

The last layer returns the reconstructed image. The results for all the above architectures are given in Table 1. All networks were trained with the Adam optimiser (Kingma and Ba, 2014). The hyper-parameters used were *learning rate* $= 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$. MNIST and CIFAR 10 models were trained for 500 epochs, CELEB A for 55.

Similarly to Tolstikhin et al. (2017), models were trained using Adam for 55 epochs, with the same optimiser parameters.

8. Cramer-Wold Generator (CWG)

To overcome saddle-point problems Deshpande et al. (2018); Li et al. (2017) introduced an alternative definition that is based on a single objective, which is a measure between distributions. The model introduced is a generative Sliced Wasserstein Generator (SWG). It is not strictly a GAN model, since no adversarial training is used, hence a generator name is used. It uses the Sliced-Wasserstein distance, estimated with a finite sum as in Kolouri et al. (2018), to express dissimilarity between P_X and $P_{\mathcal{Z}}$.

We want to show here than an analogous model which is based on the Cramer-Wold rather than Wasserstein distance is also possible. Accordingly, we shall call it a Cramer-Wold Generator (CWG). The requirements of a generator lead to the following cost function[9]

$$\text{CWG cost} = d^2_{\text{cw}}(X, \mathcal{D}(Z)),$$

where $X = (x_i)_{i=1,\dots,n} \subset \mathbb{R}^N$ is data sample and $\mathcal{D}(Z) = (\mathcal{D}(z_i))_{i=1,\dots,n} \subset \mathbb{R}^N$ means a sample from $N(0, I)$ mapped into the data space by the decoder.

We implemented CWG model and trained it on MNIST and Fashion MNIST datasets by using the original SWG architecture from Deshpande et al. (2018). Further investigation into the CWG model is fully justified by the results of our qualitative and quantitative tests that compare CWG and SWG models (see Figures 11 and 12).

9. Conclusions

In this paper we present a new kernel based on Cramer-Wold distance which gives way to a new auto-encoder based generative model CWAE. It matches, and in some cases improves, results of WAE-MMD, while using *a cost function given by a simple closed analytic formula.*

9. It should be noticed that in this case to calculate d^2_{CW} (by applying Formula (3)) we should use $\gamma = \hat{\sigma}(4/3n)^{2/5}$, where $\hat{\sigma}$ denotes the standard deviation from joined X and $\mathcal{D}(Z)$ samples. This follows from the Silverman's rule. In CWAE model it was reasonable to take $\hat{\sigma} = 1$ since we used d^2_{CW} on the latent, where an encoded sample was trained to be standard Gaussian, but now this assumption cannot be maintained.

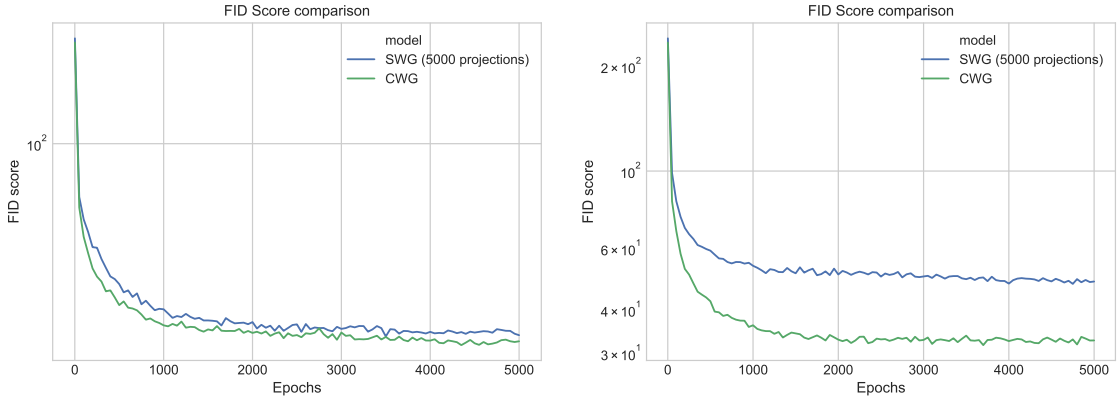

Figure 11: Comparison between SWG and CWG models with FID score on MNIST (left-hand side image) and Fashion MNIST (right-hand side image) data-sets by using original SWG architecture from Deshpande et al. (2018).

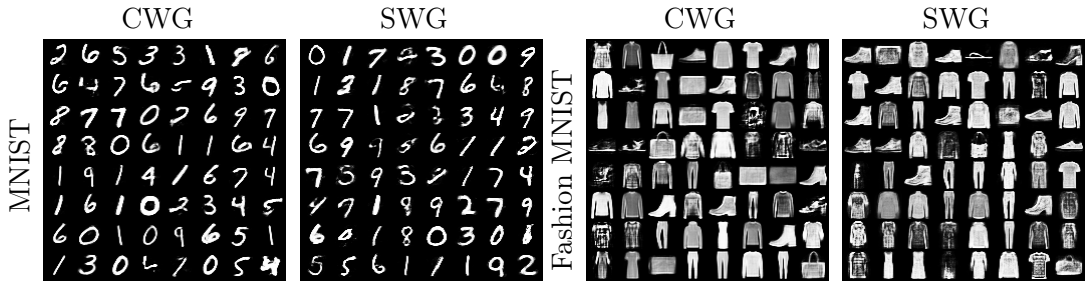

Figure 12: Randomly sampled images from SWG and CWG models trained on MNIST and Fashion MNIST using original SWG architecture from Deshpande et al. (2018).

We hope this result will encourage future work in developing simpler to optimise analogs of strong neural models.

The use of the introduced Cramer-Wold kernel and metric between samples and distributions is crucial to the construction of CWAE. This metric can be effectively computed for Gaussian mixtures. Use of Cramer-Wold kernel allows to construct methods less sensitive to changes in training parameters and faster minimizing a cost function. All this was shown in the CWAE generative auto-encoder model, which proved to be fast, more stable, and less sensitive to the changes of training parameters as compared to other methods, like WAE or SWAE. Finally, the proposed Cramer-Wold generator model shows that future work could explore use of this metric in other settings, particularly in adversarial models.

10. Acknowledgements

The work of P. Spurek was supported by the National Centre of Science (Poland) Grant No. 2019/33/B/ST6/00894. The work of J. Tabor was supported by the National Centre of Science (Poland) Grant No. 2017/25/B/ST6/01271.

I. Podolak carried out this work within the research project "Bio-inspired artificial neural network" (grant no. POIR.04.04.00-00-14DE/18-00) within the Team-Net program of the Foundation for Polish Science co-financed by the European Union under the European Regional Development Fund.

References

M. Abramowitz and I.A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55 of *National Bureau of Standards Applied Mathematics Series*. U.S. Government Printing Office, Washington, D.C., 1964.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 214–223, International Convention Centre, Sydney, Australia, 2017. PMLR.

S Aubert and CS Lam. Invariant integration over the unitary group. *J. of Mathematical Physics*, 44(12):6112–6131, 2003.

S. Axler, P. Bourdon, and R. Wade. *Harmonic function theory*, volume 137 of *Graduate Texts in Mathematics*. Springer, New York, 1992.

R.W. Barnard, G. Dahlquist, K. Pearce, L. Reichel, and K.C. Richards. Gram polynomials and the kummer function. *J. of Approximation Theory*, 94(1):128–143, 1998.

A. W. Bowman and P. J. Foster. Adaptive smoothing and density-based tests of multivariate normality. *J. of the American Statistical Association*, 88(422):529–537, 1993.

D. Braun. Invariant integration over the orthogonal group. *J. of Physics A: Mathematical and General*, 39(47):14581, 2006.

H. Cramér and H. Wold. Some theorems on distribution functions. *J. London Mathematical Society*, 11(4):290–294, 1936.

S.R. Deans. *The Radon transform and some of its applications*. Wiley, New York, 1983.

I. Deshpande, Z. Zhang, and A. G. Schwing. Generative modeling using the sliced Wasserstein distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3483–3491, 2018.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

I.S. Gradshteyn and I.M. Ryzhik. *Table of integrals, series, and products*. Elsevier/Academic Press, Amsterdam, 2015.

N. Henze. Invariant tests for multivariate normality: a critical review. *Statistical Papers*, 43 (4):467–506, 2002.

M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a Nash equilibrium. *arXiv:1706.08500*, 2017.

D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

D.P. Kingma, S. Mohamed, D.J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

S. Kolouri, Ch.E. Martin, and G.K. Rohde. Sliced-Wasserstein autoencoder: an embarrassingly simple generative model. *arXiv:1804.01947*, 2018.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. MMD GAN: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pages 2203–2213, 2017.

P. Mattila. *Geometry of sets and measures in Euclidean spaces: fractals and rectifiability*, volume 44. Cambridge university press, 1999.

K. Muandet, K. Fukumizu, B. Sriperumbudur, B. Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

B.W. Silverman. *Density estimation for statistics and data analysis*. Monographs on Statistics and Applied Probability. Chapman & Hall, London, 1986.

M. Śmieja, M. Wołczyk, J. Tabor, and B. C. Geiger. SeGMA: Semi-Supervised Gaussian Mixture Auto-Encoder. *arXiv preprint arXiv:1906.09333*, 2019.

I. Tolstikhin. WAE. `https://github.com/tolstikhin/wae`, 2018.

I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. Wasserstein auto-encoders. *arXiv:1711.01558*, 2017.

F. Tricomi and A. Erdélyi. The asymptotic expansion of a ratio of gamma functions. *Pacific J. of Mathematics*, 1:133–142, 1951.

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