INTERPOLATION IN GENERATIVE MODELS

Łukasz Struski[*]   **Jacek Tabor**   **Igor Podolak**   **Aleksandra Nowak**

Faculty of Mathematics and Computer Science
Jagiellonian University
Łojasiewicza 6, 30-348 Kraków, Poland

ABSTRACT

We show how to construct smooth and realistic interpolations for generative models, with arbitrary, not necessarily Gaussian, prior. The crucial idea is based on the construction on the realisticity index of a curve, which maximisation, as we show, leads to a search of a geodesic with respect to the corresponding Riemann structure.

1 Introduction

Since the advent of the VAE model [16] and the adversarial GAN [11], the generative models became an area of intense research, with new models being developed (see e.g. [17, 18, 28]). The central role of a generative model is its latent space, where the data space distribution is mapped. This enables to produce objects which were not present in the training data-set by sampling from the latent distribution, but comes from the data true distribution.

One of the crucial aspects of the latent space is the ability to construct interpolations between latent representations of two different objects. The meaningfulness of the generated in this manner samples is often used as an supporting argument for networks generalisation capability [5, 9]. Therefore a high quality of the midpoints of the obtained path is commonly viewed as a characteristic of a strong and well-defined model. Moreover, the interpolation itself is also often of practical use. In particular, it may allow for modification of existing data-points and help to find the intermittent objects (e.g. missing frames in a video [25, 27] or chemical molecules that lay on a specific path [10]), as well as advance creativity [24]. This encourages research into interpolation algorithms and metrics which could provide a way of their quantitative comparison.

The intuitive constraints for the interpolation include the claim that the generated objects should be meaningful, give a gradual transformation and reflect the internal structure of the data-set. More precisely we require that the transported to the input space interpolation curve is smooth and relatively short, while at the same time it proceeds through regions of high probability in the latent space, see Fig. 1. However, in some cases even for Gaussian prior the linear interpolation fails, as there may appear some discontinuities due to high non-linearity of the generator. The problems with linear interpolation are even more visible, when the prior is not Gaussian (or in general the level sets are not convex), and the linear interpolation goes through the regions of low probability.

In this paper we construct a general interpolation scheme, which works well for arbitrary priors. We consider a notion of a *realisticity index* of an element of the latent space, which naturally generalises to arbitrary curves. In addition, we propose a simple incremental algorithm to produce an interpolation which optimises the introduced index. Moreover, we show that in general the proposed method can be regarded as a search for the geodesics in respectively modified local Riemann structure.

2 Related works

Interpolation, i.e. a traversal between points in the latent space, is studied in several papers on generative models used mainly to show that the learned models do not overfit [9, 11, 12, 16], but also generalise well. We expect that if the

---

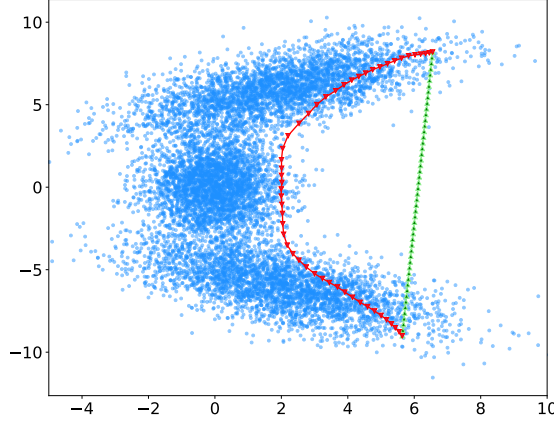[*]Corresponding author: `lukasz.struski@uj.edu.pl`

Figure 1: Data sampled from the semicircle prior distribution (mixture of three Gaussians) with first two coordinates shown. Green line denotes the linear interpolation of two points in the respective GAN model trained on CelebA data-set, while red curve is constructed by our approach.

data-set has some natural hierarchy, then it should be reflected in a well trained model. As it is pointed out in [3], frequently the trained generative models result in latent space with empty spaces between valid mappings. Those regions of low density reduce the generative usability. For example, in the case of Gaussian prior latent, a straightforward linear interpolation may result in a so-called distribution mismatch. This discrepancy appears because a random vector generated by linear interpolations has different distribution then the prior. Original data points have higher norms, while the mid-points of linear interpolation are samples from inside a sphere while high dimensional vectors drawn from latent prior distribution usually have large norms (a so called *bubble effect* [13]). This may result in low quality samples in the middle of the path [29]. The above argument puts the usability of simple linear interpolation in question and motivates further research in this area [15, 20, 29].

The interesting problem is whether the latent space generated during the applied training regime captures the semantic contents of the data space, including its structure, as postulated in [4, 8, 22, 23, 29]. This greatly depends on the nature of the generated latent space, which actually is a Riemannian manifold with locally defined, smoothly changing, dot product [7]. This feature is pointed out in [2], together with a fine discussion. The authors propose to use geodesics on the Riemannian manifold as the shortest path during interpolation between points.

The central question posed by several authors is what type of interpolation should be chosen in a particular model. The predominantly used algorithm is the linear interpolation. However, in a multi-dimensional space with a Gaussian or uniform prior, such interpolation may traverse through highly unlikely locations. Therefore other, e.g. spherical, approaches were proposed [29]. The key suggestion in [23] is that traversal in the latent space should result in *semantic* changes. Those changes could manifest themselves in some features of the generated object being changed or added. The authors clearly show that such transitions in latent space are even capable of vector arithmetic computation, while the same approach in the data pixel space produce only noisy overlaps. Virtually all the objects in some interpolation *must* pertain the data space that the trained model generates, as well as enable arithmetic operations in the latent together with feature disentanglement property [1]. A *good* interpolation should meet the following conditions: sampling should not violate the prior, the path curve should follow the distribution of the modelled data space, be smooth, and enable some latent space feature arithmetical operations. However, a clear definition still awaits completion.

Some approaches to counteract the "emptiness" of the high-dimensional Gaussian, apart from walking on the sphere, were recently proposed, by introducing operations on the distribution of the latent space [1, 6, 15, 18, 20].

In the following sections we show how to build interpolations in the latent space, with respect to standard generative models (GAN-like or AE-like). First, a concept of point and path realisticity is introduced and justified. Then we describe the optimisation algorithm and report experimental results.

3   Realisticity index of a point

Let $X \subset \mathbb{R}^N$ be a data-set. We assume that we are given a generative model for X, which consists of a generator (decoder) $\mathcal{G}$ from the latent $Z = \mathbb{R}^D$ to the input data space:

$$\mathcal{G} : Z \to \mathbb{R}^N,$$

and a density f on Z. By a *realisticity index on Z* we understand a function
$$\mathrm{ri} : Z \to [0, 1]$$
such that high values of $\mathrm{ri}(z)$ indicate that $\mathcal{G}(z)$ is indistinguishable from the elements of X. We interpret the realisticity index of z as the "informal probability" that the point $\mathcal{G}(z)$ is realistic.

In general the optimal choice of ri can be nontrivial and may depend on the generative model in question. However, if we fix our attention on the typical case when the density f in Z is given, then empirically in generative models we often notice that the higher values of the density $f(z)$ indicate better realisticity of $\mathcal{G}z$, as might be observed in the GLOW [17] model.

Density based realisticity index. Concluding, our aim is to construct a density based index, with values in the interval $[0, 1]$ such that *higher density implies higher realisticity index*.

Observe that, given k-values of density $f(x_1), \ldots, f(x_k)$, and putting $f_i = f(x_i)$ the above property is satisfied when we define $\mathrm{ri}(x_i)$ as the normalised rank in the ordered sequence $f_{(i)}$. This motivates us to state the following definition.

Definition 1. *Let $\mathbb{X}$ be a random vector with density f. We define the* realisticity index ri *based on the density f by the formula*
$$\mathrm{ri}(z; f) := p(f(\mathbb{X}) \leq f(z)) = \int_{\{w : f(w) \leq f(z)\}} f(s) ds,$$
where p denotes the probability.

The proposed index lies in the interval $[0, 1]$, and attains 1 only for points of maximal density. If f is clear from the context, for shortness of notation we write $\mathrm{ri}(z)$ instead of $\mathrm{ri}(z; f)$.

To practically apply the above concept, we need to be able to tune it to different generative models. To achieve this, we choose a value $\alpha < 1$ and, similarly to the approach used in GLOW [17], rescale the index by α:
$$\mathrm{ri}_\alpha(z) = \alpha \cdot \mathrm{ri}(z).$$

Example 1. *Let us consider the simplest case when f is the normalised uniform density on the bounded set $U \subset Z$, i.e.*
$$f = \mathrm{uni}_Z = \frac{1}{\mathrm{vol}(U)} \cdot \mathbb{1}_U,$$
where $\mathbb{1}_U$ denotes the characteristic function of the set U. Now one can easily see that
$$\mathrm{ri}(z; f) = \mathbb{1}_U(z).$$

A more interesting and important example is given by the standard normal density function.

Case of the normal density. We shall now compute the probability index of the standard normal density $f = \mathcal{N}(0, I)$ in the D-dimensional latent Z. Let us choose a point z from the latent and let $\mathbb{X}$ denote the random vector with density f. We want to compute the probability
$$\mathrm{ri}(z) = p(f(\mathbb{X}) \leq f(z)).$$
Observe that from the definition of normal density we have
$$p(f(X) \leq f(z)) = p(\|\mathbb{X}\|^2 \geq \|z\|^2) = 1 - p(\|\mathbb{X}\|^2 \leq \|z\|^2).$$
Since $\|\mathbb{X}\|^2$ has the chi-square distribution with D degrees of freedom, we obtain that
$$\mathrm{ri}(z) = 1 - F(\|z\|^2; D),$$
where $F(r; D)$ denotes the cumulative distribution function of the chi-square density $\chi^2(D)$ (with D degrees of freedom).

Let us now proceed with a more detailed asymptotic analysis. Observe that
$$p(\|\mathbb{X}\|^2 \leq \|z\|^2) = p(\sqrt{2}\|\mathbb{X}\| \leq \sqrt{2}\|z\|).$$
If $\mathbb{Y} \sim \chi^2(D)$ then for large D, $\sqrt{2\mathbb{Y}} - \sqrt{2D - 1}$ is approximately normally distributed (see [14, formula (18.23) on p. 426]). Consequently
$$\mathrm{ri}(z) = 1 - p(\sqrt{2}\|\mathbb{X}\| \leq \sqrt{2}\|z\|) \approx 1 - \Phi(\sqrt{2}\|z\| - \sqrt{2D - 1}.$$
where Φ denotes cdf of standard Gaussian. Since $\Phi(r) = \frac{1}{2}\left[1 + \mathrm{erf}(r/\sqrt{2})\right]$ we get
$$\mathrm{ri}(z) \approx \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\sqrt{D - \tfrac{1}{2}} - \|z\|\right). \tag{1}$$

The above formula implies that the realisticity index is practically 1 in the ball $B(0, \sqrt{D - \tfrac{1}{2}} - 3)$, and decreases with growing $\|z\|$ norm, so that it is approximately 0 outside of the ball $B(0, \sqrt{D - \tfrac{1}{2}} + 3)$.

3

Estimation of the realisticity index in the general case.   Clearly, for an arbitrary density f the realisticity index does not have a closed form. In order to practically estimate it, one can draw a relatively large number of n samples $W = (w_i)_i$ from the random variable $\mathbb{X}$ with density f, and compute their densities[2] $(f(w_i))_i$. Since the values of density are non-negative we may consider their log-likelihoods:

$$l_i = \log f(w_i).$$

Then the kernel density approximation of the random variable $\log(f(\mathbb{X}))$ is given by

$$g_W(x) = \tfrac{1}{N} \sum_i \mathcal{N}(l_i, h^2)(x),$$

where h is taken using the Silverman's rule $h = \left(\frac{4}{3n}\right)^{1/5} \hat{\sigma}$ and $\hat{\sigma}$ denotes the deviation of the sample $(l_i)_i$. Consequently the estimator of the cdf of $\log(f(\mathbb{X}))$ is given by:

$$G_W(x) = \tfrac{1}{n} \sum_i \Phi(\tfrac{x-l_i}{h}),$$

where Φ is the cdf of $\mathcal{N}(0,1)$ (expressed using special function erf). Concluding, we obtain that the estimator of the realisticity index $\mathrm{ri}(z; f)$ is given by

$$\mathrm{ri}(z; f) \approx G_W(\log f(z))).$$

4 Realisticity index of an interpolation

Our motivation for the definition of the realisticity index for a path is inspired by transition between movie frames. The interpolation may be viewed as set of frames, where the first frame denotes the beginning of the path and last its end. Interpreting realisticity index as a probability that a given frame is realistic, we can define the respective index of the curve as the product of all its points.

Realisticity index for naturally parameterised curves.   Let $\gamma : [0, T] \to Z$ be an interpolating curve, such that

$$\gamma(0) = x \text{ and } \gamma(T) = y,$$

for some given points $x, y \in Z$. Additionally, we assume that the $\mathcal{G}\gamma$ is naturally parameterised:

$$\|(\mathcal{G}\gamma)'(t)\| = 1 \text{ for } t \in [0, T],$$

We discretize the curve by fixing a time-step T/k (where k denotes the number of frames) and consider the sequence of intervals

$$[\gamma(0), \gamma(T/k)], \ldots, [\gamma(T - T/k), \gamma(T)].$$

To obtain the reality measure of this sequence we compute the product of all realisticities of its points to the power equal to their "duration" T/k

$$\mathrm{ri}(\gamma(t_0))^{T/k} \cdot \ldots \cdot \mathrm{ri}(\gamma(t_k))^{T/k},$$

where t_i are the arbitrarily chosen intermediate points in the intervals $[iT/k, (i+1)T/k]$. By taking the logarithm of the above expression and proceeding with $k \to \infty$ we get

$$\sum_{i=1}^{k} \log \mathrm{ri}\, \gamma(t_i) \cdot \tfrac{T}{k} \to \int_0^T \log \mathrm{ri}(\gamma(t)) dt \text{ as } k \to \infty.$$

Consequently, we introduce the realisticity index of a naturally parameterised curve $\gamma : [0, T] \to \mathbb{R}^N$ by the formula

$$\mathrm{ri}(\gamma) = \exp\left(\int_0^T \log \mathrm{ri}(\gamma(t)) dt\right) \in [0, 1].$$

Since every curve can be uniquely naturally parameterised, we understand its index as the index of its natural reparameterisation. Therefore we arrive at the following general definition.

Definition 2. *Let* ri *be given realisticity index in Z. For an arbitrary curve $\gamma : [0, T] \to Z$ we define the* realisticity index ri *of γ by*

$$\mathrm{ri}(\gamma) = \exp\big(\int_0^T \log \mathrm{ri}(\gamma(t)) \|(\mathcal{G}\gamma)'(t)\| \, dt\big) \in [0, 1].$$

---

[2]Since we work in the scalar case, typically in order to obtain reliable density estimation it is enough to take $N = 5000$.

We further prove that the realisticity index of a curve is equal to its length with respect to a certain Riemann structure on the latent space. We will utilise this result in the next section in order to connect the search of optimal interpolation to the search of geodesics. Directly from the definition we get the formula for the realisticity index in terms of the latent:

$$-\log \text{ri}(\gamma) = -\int_0^T \log \text{ri}(\mathcal{G}\gamma(t)) \cdot \|(\mathcal{G}\alpha)'(t)\| dt$$

$$= -\int_0^T \log \text{ri}(\gamma(t)) \cdot \sqrt{\langle (\mathcal{G}\gamma)'(t), (\mathcal{G}\gamma)'(t) \rangle} dt$$

$$= -\int_0^T \log \text{ri}(\gamma(t)) \sqrt{\gamma'(t)^T [d\mathcal{G}(\gamma(t))]^T d\mathcal{G}(\gamma(t))\gamma'(t)} dt$$

$$= \int_0^T \sqrt{\log^2 \text{ri}(\gamma(t))\gamma'(t)^T [d\mathcal{G}(\gamma(t))]^T d\mathcal{G}(\gamma(t))\gamma'(t)} dt,$$

where $d\mathcal{G}(x)$ denotes the derivative of $\mathcal{G}$ at point x. Consequently, we obtain the following theorem:

Theorem 1. *Let the Riemann structure in the latent space Z be defined by the local scalar product $\langle , \rangle_z$ at a point z using the formula*

$$\langle v, w \rangle_z = v^T A_z w \text{ where } A_z = \log^2(\text{ri}(z)) d\mathcal{G}(z)^T d\mathcal{G}(z).$$

Then

$$\text{ri}(\gamma) = \exp(-\text{length}(\gamma; \langle , \rangle_z)).$$

5 Optimal interpolation

Considering the results from the previous section, we are able to formulate the definition of an ri-optimal curve.

Definition 3. *Let* ri *be a realisticity index in Z for a generative model $\mathcal{G} : Z \to \mathbb{R}^N$. Let $x, y \in Z$ be fixed. We call a curve $\gamma : [0, T] \to Z$ such that $\gamma(0) = x$, $\gamma(T) = y$ ri-optimal (or shortly optimal) interpolation if it has the minimal realisticity index from all curves joining x with y.*

First, we consider the issue of searching for, at least locally, optimal interpolations. Theorem 1 allows us to reformulate the problem as a task of finding geodesics. Consequently, the standard results from Riemann geometry apply (see [26, Chapter 9]). Without loss of generality we can reduce the problem to the case $T = 1$ and optimise the length functional. However, due to the uniqueness of the local minima and local convexity of the functional, we can minimise the energy functional instead:

$$E = \tfrac{1}{2} \int_0^1 \langle d\mathcal{G}(\gamma(t))\gamma'(t), \mathcal{G}(\gamma(t))\gamma'(t) \rangle_{\gamma(t)} dt.$$

The additional advantage is that curves which minimise the energy functional are parameterised proportionally to the natural parameterisation.

Concluding, by applying Theorem 1, the optimal curve $\gamma : [0, 1] \to \mathbb{R}^N$, $\gamma(0) = x, \gamma(1) = y$, with respect to the realisticity index minimises:

$$E_x^y(\gamma) = \tfrac{1}{2} \int_0^1 \log^2(\text{ri}(\gamma(t))) \|d\mathcal{G}(\gamma(t))\gamma'(t)\|^2 dt. \tag{2}$$

In general, the search for geodesics can lead to nontrivial computations involving second derivatives. However, for some special cases, we can significantly simplify the minimisation process. To justify this claim, we first introduce the formula for the discretization of the integral in the energy functional. Let $\gamma : [0, 1] \to \mathbb{R}^N$ be a curve such that

$$\gamma(0) = x \text{ and } \gamma(1) = y.$$

We divide the interval $[0, 1]$ into k equal sub-intervals, and denote the values

$$\gamma(i/k) = x_i \text{ for } i = 0..k.$$

For $k - 1$ vectors $x_1, \dots, x_{k-1}$ in $\mathbb{R}^N$ (where $x_0 = x, x_k = y$), we approximate the value of (2) by

$$2E_x^y(\gamma) \approx \sum_{i=0}^{k-1} \log^2 \frac{\text{ri}(\gamma((i+1)/k)) + \text{ri}(\gamma(i/k))}{2}$$

$$\cdot \left(\frac{\|\gamma((i+1)/k) - \gamma(i/k)\|}{1/k}\right)^2 \cdot \frac{1}{k}$$

$$= k \log^2 \frac{\text{ri}(x) + \text{ri}(x_1)}{2} \|x_1 - x\|^2$$

$$+ k \sum_{i=1}^{k-2} \log^2 \frac{\text{ri}(x_i) + \text{ri}(x_{i+1})}{2} \|x_{i+1} - x_i\|^2$$

$$+ k \log^2 \frac{\text{ri}(x_{k-1}) + \text{ri}(y)}{2} \|y - x_{k-1}\|^2.$$

Considering all the above computations, in order to produce optimal interpolation we need to minimise the function

$$\begin{aligned}
\tfrac{2}{k} \cdot E_x^y(x_1, \ldots, x_{k-1}) = {} & \log^2 \tfrac{\mathrm{ri}(x) + \mathrm{ri}(x_1)}{2} \|x_1 - x\|^2 \\
& + \sum_{i=1}^{k-2} \log^2 \tfrac{\mathrm{ri}(x_i) + \mathrm{ri}(x_{i+1})}{2} \|x_{i+1} - x_i\|^2 \\
& + \log^2 \tfrac{\mathrm{ri}(x_{k-1}) + \mathrm{ri}(y)}{2} \|y - x_{k-1}\|^2,
\end{aligned} \tag{3}$$

over $x_1, \ldots, x_{k-1} \in \mathbb{R}^N$.

In all the experiments in the paper we achieve this goal using the gradient descent method and initialising x_i with the linear interpolation

$$x_i = \left(1 - \tfrac{i}{k}\right) x + \tfrac{i}{k} y \ \text{ for } i = 1..k - 1.$$

Discussion. Having established basic facts and notions concerning ri-optimal interpolations, we discuss some special cases.

Observe, that in the minimisation of (3) we aim to simultaneously maximise the value of realisticity index, while minimising the length of the interpolation in the input space. Consequently, the obtained curve should go through regions of high probability, encouraging the interpolation to produce realistic data points, while on the same time have as small length in the input space as possible, resulting in smooth interpolation.

In the case of r_α-optimal interpolation, with α close to zero, only the second term survives. For simplicity of discussion, we consider the case when $\mathrm{ri}(z) > 0$ for all $z \in Z$. This may happen when $f(z) > 0$ for all $z \in Z$. Directly from (2) we obtain that for a fixed interpolation

$$E_x^y(\gamma) = \tfrac{1}{2} \int_0^1 [\log(1/\alpha) - \log \mathrm{ri}(\gamma(t))]^2 \cdot \|d\mathcal{G}(\gamma(t))\gamma'(t)\|^2 dt.$$

Consequently, for α close to zero,

$$2 \log(1/\alpha)^2 E_x^y(\gamma) \approx \int_0^1 \|d\mathcal{G}(\gamma(t))\gamma'(t)\|^2 dt.$$

Then the minimum over γ of the above functional is obtained for those curves between x and y which have minimal length after the transportation to the input space.

In the following paragraphs we discuss another interesting case, which assumes that the generator is linear with uniform distribution in the latent.

Observation 1. *Let U be a convex bounded set in Z and let f denote the uniform distribution on U. We consider the case of linear generative model, i.e. where $\mathcal{G}$ is a linear map $x \to Ax$ (with A injective).*

Then the ri-*optimal interpolations are given by the linear interpolations.*

Proof. This follows from the fact that we can equivalently compute the realisticity index of a curve γ connecting two points x, y by computing the standard euclidean distance of $A\gamma$ in the convex set AU. More precisely:

$$\mathrm{ri}(\gamma; \mathrm{uni}_U) = \exp(-\mathrm{length}(A\gamma)).$$

Since linear maps move intervals onto intervals, we obtain the assertion of the corollary. $\square$

In practice, a similar behaviour may happen when the derivative of the generator has small variation and the realisticity index is close to being a constant one. Observe, that by equation (1), for a generative model with high dimension of the latent space and Gaussian prior, the realisticity index is almost constant on a linear interpolation of arbitrary randomly chosen points, except for a possibly small neighbourhood of the endpoints. Consequently, when the generator $\mathcal{G}$ does not vary too much in the vicinity of the linear interpolation from the affine function, the linear interpolation will be close to optimal. In practice, we observe this behaviour in auto-encoder based generative models (see section 6). This may follow from the fact that to high variation of the decoder is penalised by its approximate inverse given by the encoder.

6

Figure 2: Examples of results for linear interpolation (on top each examples) and our results (bottom each examples) for Celeb-A data-set, with a semicircle prior trained DCGAN. Starting interpolation had 50 midpoints, and every fifth is shown.

(a) Interpolations for a normal prior trained DCGAN model. In each pair the top one is the linear, the bottom the computed interpolation. Again each computed interpolation had 50 points, with every fifth shown.
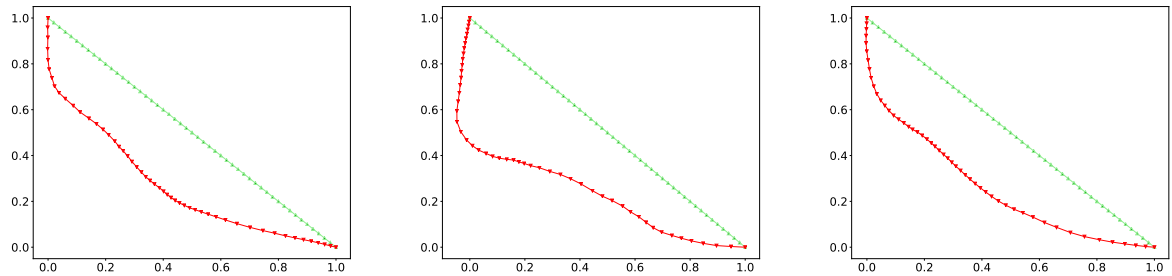
(b) Projections (see (4)) of linear (green line) and proposed (red line) interpolations shown in (a) from top to bottom, respectively.

Figure 3: Results for a normal prior latent DCGAN model.

Figure 4: Linear and proposed interpolation paths for the MNIST data-set (a semicircle latent prior model in left column, a multidimensional normal in right column) with a DCGAN model used.

6 Experiments

In this section we demonstrate our method's ability to produce more meaningful interpolations. In order to achieve this goal we use a DCGAN model [23] trained on MNIST and Celeb-A data-sets [19, 21]. We consider two type of latent space prior distributions: classical, with samples generated from standard normal $\mathcal{N}(0, I)$ prior, and one created from a conjunction of three multidimensional Gaussian distributions

$$p(z) = \frac{1}{3} \sum_{i=1}^{3} \mathcal{N}(\mu_i, \Sigma_i)(z) \text{ for } z \in Z,$$

where $\mu_i := (\tilde{\mu}_i, 0, \ldots, 0)^T \in \mathbb{R}^{20}$ for $\tilde{\mu}_1 = (2, 6)$, $\tilde{\mu}_2 = (0, 0)$, $\tilde{\mu}_3 = (2, -6)$ and $\Sigma_i := \begin{bmatrix} M_{11}^{(i)} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$ for $M_{12} = M_{21} = \mathbf{0}$, M_{22} is 2-D array with 0.5 on the diagonal and zeros elsewhere, and $M_{11}^{(1)} = \begin{bmatrix} 5 & 2 \\ 2 & 2 \end{bmatrix}$, $M_{11}^{(2)} = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}$, $M_{11}^{(3)} = \begin{bmatrix} 5 & -2 \\ -2 & 2 \end{bmatrix}$. From here on we call it a semicircle (in 2-D it resembles a horseshoe, see Fig. 1). We use it mainly as a proof-of-concept distribution.

Results of sampling from a GAN trained with the semicircle prior are shown in Fig. 2. We randomly choose two points from different ends of the distribution and start the algorithm's minimisation procedure (3) with a linear interpolation. It is obvious that the midpoints of initial path are sampled from a very low-density areas, completely outside the prior.
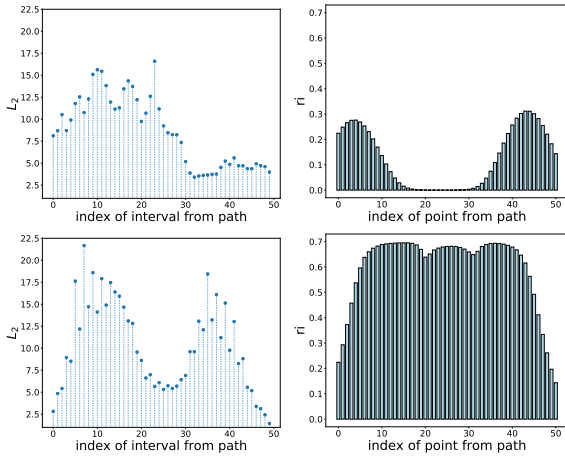

Figure 5: L_2 distances between images generated from consecutive points in path (left column, linear interpolation in top line, proposed path optimisation in bottom) and the realisticity index ri (right column) for the semicircle distribution. Please refer to Fig. 2 for the resulting images in the last example.
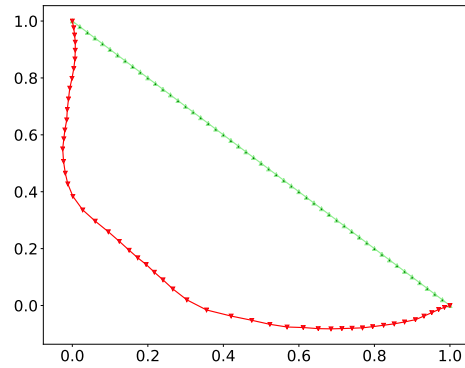
Figure 6: A projection of the linear (green) and proposed (red line) 20-dimensional interpolation corresponding to Fig. 5.

Fig. 5 shows the realisticity index for the linear path and the interpolation found by our method. An example of the reconstruction of the midpoints is presented in figure Fig. 2. As one can easily see in Fig. 5, in the linear case the consecutive points inside the path usually have similar L_2 distances. Our method allows for more dynamic behaviour in this matter, especially at the end points. Images in our path are more different, and at the same time more real, as evidenced by the realisticity index presented in Fig. 5 (charts on the right side).

In order to visualise the obtained path relative to the starting linear interpolation, we projected both paths onto a plane defined by the endpoints and the coordinate system origin. In chart on the left side of Fig. 6 a pair $(x, y) \in \mathbb{R}^2$ is projected using

$$x \cdot \tilde{z}_0 + y \cdot \tilde{z}_k = \tilde{z}_i \quad \text{for} \quad i = 0..k, \tag{4}$$

where $\tilde{z}_i := X(X^T X)^{-1} X^T z_i$, $X = [z_0, z_k]$ are the projection points $z_i \in Z$ of the path in the latent space.

Figure 7: Linear and proposed interpolation in the WAE model. *Left:* The reconstruction of the interpolation in latent space. The leftmost and rightmost images correspond to the reconstruction of original points in test set. *Right:* The projection of the path (red) compared to linear interpolation (green).

Similar experiments were performed for a Gaussian prior DCGAN model, see Fig. 3a. The L_2 distances and realisticity values are shown in Fig. 8, with projections in Fig. 3b. The behaviour of the algorithm is similar to the one with the proof-of-concept semicircle prior distribution. This shows the generality of the proposed solution. We also report the
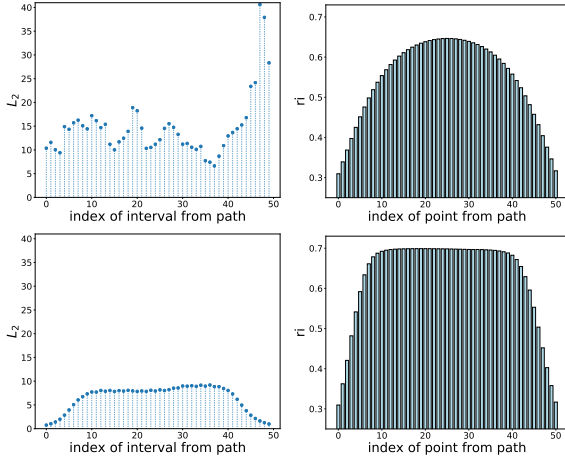

Figure 8: L_2 distances between consecutive points in the last example path from Fig. 3a (linear in top line, proposed in bottom) and the realisticity index ri for points from latent space defined by prior $\mathcal{N}(0, I)$.
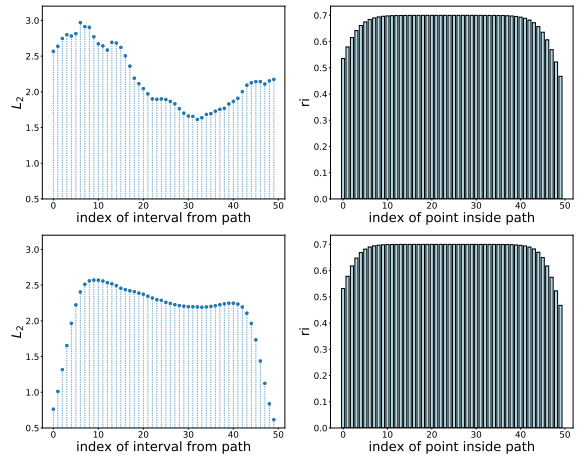
Figure 9: L_2 distances between consecutive points and the ri index for interpolation path in the WAE model.

same metrics for the MNIST data-set with results shown in Fig. 4.

Additionally, we test our approach on the latent space learned by an auto-encoder model. Specifically, we train a Wasserstein auto-encoder [28] on the Celeb-A data-set. We use the architecture and configuration provided by the authors as well as a 64-dimensional Gaussian prior. Fig. 7 presents an example of a reconstructed interpolation path between the encodings of randomly chosen test images. We observe that, in general, the learned interpolation does not differ much from its linear counterpart and that the realisticity index for the latter is rather high to begin with. We argue that this may be the effect of introducing the encoder function. The reconstruction requirement in AE imposes an additional constraint on the generator, encouraging the model to keep semantically similar points close to each other in the latent space. In consequence, the path points seem to have smooth transitions (as seen in Fig. 9). However, the plausibility of the provided argument could be subject for further research.

7 Conclusions

In this paper we studied the problem of generating a meaningful interpolation from a previously trained generative model, either a GAN or a generative auto-encoder. We claim that a good interpolation should both reveal the hidden structure of the data-set, as well as be smooth and follow the true data distribution.

In order to produce curves satisfying these conditions we define a *realisticity* index of a path, which takes into account both density values and and differences between consecutive decoded images to ensure smoothness. The experiments show that the constructed interpolations are in general superior to the linear ones.

References

[1] Eirikur Agustsson, Alexander Sage, Radu Timofte, and Luc Van Gool. Optimal transport maps for distribution preserving operations on latent spaces of generative models. *CoRR*, abs/1711.01970, 2017. 2

[2] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauber. Latent space oddity: on the curvature of deep generative models. *CoRR*, abs/1711.11379, 2017. 2

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8), August 2013. 2

[4] David Berthelot, Colin Raffel, Aurko Roy, and Ian J. Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *CoRR*, abs/1807.07543, 2018. 2

[5] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. *CoRR*, abs/1511.06349, 2015. 1

[6] Andrew Brock, Theodore Lim, James M. Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *CoRR*, abs/1609.07093, 2016. 2

[7] Manfredo Perdigao do Carmo. *Riemannian geometry*. Birkäuser, 1992. 2

[8] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *CoRR*, abs/1605.09782, 2016. 2

[9] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. Adversarially Learned Inference. *CoRR*, 1606.00704, 2016. 1

[10] Rafael Gómez-Bombarelli, David K. Duvenaud, José Miguel Hernández-Lobato, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *CoRR*, abs/1610.02415, 2017. 1

[11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *CoRR*, abs/1406.2661, 2014. 1

[12] I. Higgins, L. Matthey, A. Pal, Ch. Burgess, Glorot X., M. Botvinick, S. Mohamed, and A. Lerchner. β-VAE: Learning basic visual concepts with constrained variational framework. In *Proceedings of International Conference on Learning Reepresentations ICLR*, 2016. 1

[13] Ferenc Husar. Gaussian distributions are soap-bubbles. http://www.inference.vc/high-dimensional-gaus\sian-distributions-are-soap-bubble, 2017. Accessed: 2019-03-18. 2

[14] Norman Lloyd Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. Continuous univariate distributions, vol. 1. 1994. 3

[15] Yannic Kilcher, Aurélien Lucchi, and Thomas Hofmann. Semantic interpolation in implicit models. *CoRR*, abs/1710.11381, 2017. 2

[16] D. P Kingma and M. Welling. Auto-Encoding Variational Bayes. *CoRR*, 1312.6114, 2013. 1

[17] Diederick P. Kingma and Prafulla Dhariwal. Glow: generative flow with 1×1 convolutions. *CoRR*, abs/1807.03039, 2018. 1, 3

[18] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *CoRR*, abs/1512.09300, 2015. 1, 2

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 8

[20] Damian Lesniak, Igor Sieradzki, and Igor Podolak. Distribution interpolation trade off in generative models. In *Proceedings of International Conference on Learning Representations (ICLR)*, May 2019. 2

[21] 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. 8

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. 2

[23] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. 2, 8

[24] Adam Roberts, Jesse Engel, and Douglas Eck. Hierarchical vaeriational eutoencoders for music. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2017. 1

[25] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. *CoRR*, abs/1611.06624, 2017. 1

[26] Michael D Spivak. *A comprehensive introduction to differential geometry, Volume One, Third Edition*. Publish or perish, 1999. 5

[27] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Single-view to multi-view: reconstructing unseen views with a convolutional network. *CoRR*, abs/1511.06702, 2015. 1

[28] Iliya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and B. Schoelkopf. Wasserstein Auto-Encoders. *CoRR*, 1711.01558, 2017. 1, 9

[29] Tom White. Sampling generative networks: Notes on a few effective techniques. *CoRR*, abs/1609.04468, 2016. 2