OPTIMAL TRANSPORT MAPS FOR DISTRIBUTION PRESERVING OPERATIONS ON LATENT SPACES OF GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models such as Variational Auto Encoders (VAEs) and Generative Adversarial Networks (GANs) are typically trained for a fixed prior distribution in the latent space, such as uniform or Gaussian. After a trained model is obtained, one can sample the Generator in various forms for exploration and understanding, such as interpolating between two samples, sampling in the vicinity of a sample or exploring differences between a pair of samples applied to a third sample. In this paper, we show that the latent space operations used in the literature so far induce a distribution mismatch between the resulting outputs and the prior distribution the model was trained on. To address this, we propose to use distribution matching transport maps to ensure that such latent space operations preserve the prior distribution, while minimally modifying the original operation. Our experimental results validate that the proposed operations give higher quality samples compared to previous approaches.

1 INTRODUCTION & RELATED WORK

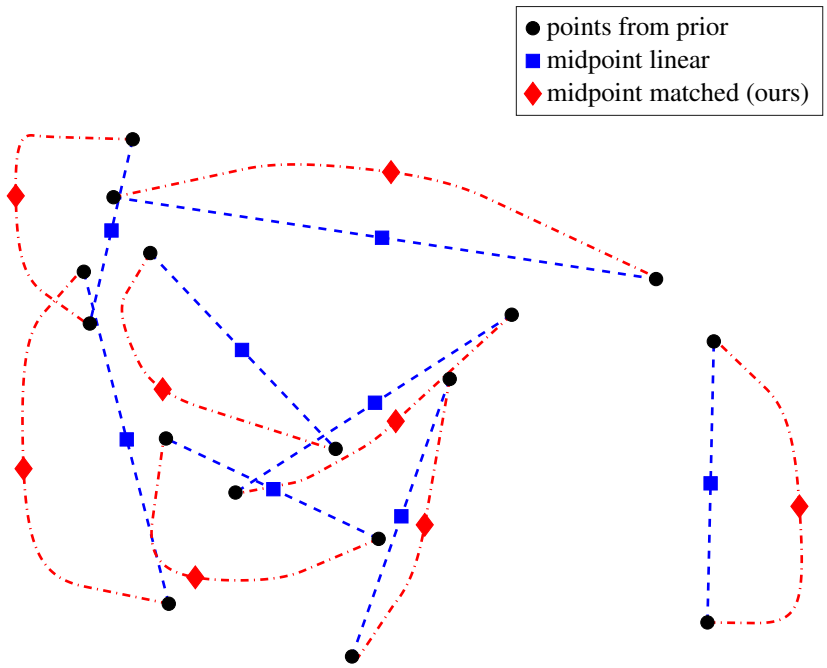
Generative models such as Variational Autoencoders (VAEs) (Kingma & Welling, 2013) and Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) have emerged as popular techniques for unsupervised learning of intractable distributions. In the framework of Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), the generative model is obtained by jointly training a generator G and a discriminator D in an adversarial manner. The discriminator is trained to classify synthetic samples from real ones, whereas the generator is trained to map samples drawn from a fixed prior distribution to synthetic examples which fool the discriminator. Variational Autoencoders (VAEs) (Kingma & Welling, 2013) are also trained for a fixed prior distribution, but this is done through the loss of an autoencoder that minimizes the variational lower bound of the data likelihood. For both VAEs and GANs, using some data $\mathcal{X}$ we end up with a trained generator G, that is supposed to map latent samples $\boldsymbol{z}$ from the fixed prior distribution to outputs samples $G(\boldsymbol{z})$ which (hopefully) have the same distribution as the data

However, there is the concern of over-fitting: if the model has enough parameters and is improperly trained, it could just learn to memorize the training examples, without learning a meaningful representation of the data. Therefore, a desirable property of the generator is to learn a *smooth* representation of the data space, where we can transition between output samples in a realistic manner, without leaving the "data manifold". Therefore, it is a common practice in the literature of generative models to explore how the output samples $G(z)$ of the learned models behave under various arithmetic operations on the latent samples z.

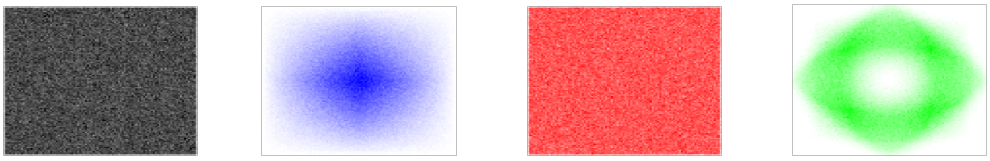
In this paper, we show that the operations typically used so far, such as linear interpolation (Goodfellow et al., 2014), spherical interpolation (White, 2016), vicinity sampling and vector arithmetic (Radford et al., 2015), cause a distribution mismatch between the latent prior distribution and the results of the operations. This is problematic, since the generator G was trained on a fixed prior and expects to see inputs with statistics consistent with that distribution.

To address this, we propose to use distribution matching transport maps, to obtain analogous latent space operations (e.g. interpolation, vicinity sampling) which preserve the prior distribution of the latent space.

In Figure 1 we showcase how our proposed technique gives an interpolation operator which avoids distribution mismatch when interpolating between samples of a uniform distribution.



(a) Between eight pairs of points drawn from a uniform prior (b) we show the trajectories of linear interpolation and our matched interpolation, as well as the associated midpoints.



(b) The uniform prior distribution. (c) linear midpoint distribution (d) matched midpoint distribution (ours) (e) spherical midpoint distribution of White (2016)

Figure 1: We show an example distribution mismatch induced by previous interpolation schemes when using a uniform prior in two dimensions (i.e. on $[-1, 1]^2$). Our matched interpolation is a minimal modification of the linear trajectory which traverses through the space such that all points along the path are distributed identically to the prior.

1.1 Generative Models and Sample Operations

In the literature there are dozens of papers that use sample operations to explore the learned models. Bengio et al. (2013) use linear interpolation between neighbors in the latent space to study how well deep vs shallow representations can disentangle the latent space of Contractive Auto Encoders (CAEs) (Rifai et al., 2011).

In the seminal GAN paper of Goodfellow et al. (2014), the authors use linear interpolation between latent samples to visualize the transition between outputs of a GAN trained on MNIST. Dosovitskiy et al. (2015) linearly interpolate the latent codes of an auto encoder trained on a synthetic chair dataset.

Radford et al. (2015) also linearly interpolate between samples to evaluate the quality of the learned representation. Furthermore, motivated by the semantic word vectors of Mikolov et al. (2013), they explore using vector arithmetic on the samples to change semantics such as adding a smile to a generated face.

Operation	Expression
2-point interpolation	$\boldsymbol{y} = t\boldsymbol{z}_1 + (1-t)\boldsymbol{z}_2$, $t \in [0,1]$
n-point interpolation	$\boldsymbol{y} = \sum_{i=1}^{n} t_i \boldsymbol{z}_i$ with $\sum_i t_i = 1$
Vicinity sampling	$\boldsymbol{y}_j = \boldsymbol{z}_1 + \epsilon \boldsymbol{u}_j$ for $i = 1, \cdots, k$
Analogies	$\boldsymbol{y} = \boldsymbol{z}_3 + (\boldsymbol{z}_2 - \boldsymbol{z}_1)$

Table 1: Examples of interesting sample operations which need to be adapted if we want the distribution of the result $\boldsymbol{y}$ to match the prior distribution. For interpolation and vicinity sampling, we want the distribution to be identical to the prior distribution.

Reed et al. (2016) and Brock et al. (2016) use also interpolation in recent works ignoring the distribution mismatch.

1.2 Distribution Mismatch and Related Approaches

While there are numerous works performing operations on samples, most of them have ignored the problem of distribution mismatch, such as the one presented in Figure 1d. Kingma & Welling (2013) and Makhzani et al. (2015) sidestep the problem when visualizing their models, by not performing operations on latent samples, but instead restrict the latent space to 2-d and uniformly sample the percentiles of the distribution on a 2-d grid. This way, the samples have statistics that are consistent with the prior distribution. However, this approach does not scale up to higher dimensions - whereas the latent spaces used in the literature can have hundreds of dimensions.

Related to our work, White (2016) experimentally observe that there is a distribution mismatch between the distance to origin for points drawn from uniform or Gaussian distribution and points obtained with linear interpolation, and propose to use a so-called *spherical linear interpolation* to reduce the mismatch, obtaining higher quality interpolated samples. However, the proposed approach has no theoretical guarantees.

In this work, we propose a generic method to fully preserve the desired prior distribution when using sample operations. The approach works as follows: we are given a 'desired' operation, such as linear interpolation $\boldsymbol{y} = t\boldsymbol{z}_1 + (1-t)\boldsymbol{z}_2$, $t \in [0,1]$. Since the distribution of $\boldsymbol{y}$ does not match the prior distribution of $\boldsymbol{z}$, we search for a warping $f : \mathbb{R}^d \to \mathbb{R}^d$, such that $\tilde{\boldsymbol{y}} = f(\boldsymbol{y})$ has the same distribution as $\boldsymbol{z}$. In order to have the modification $\tilde{\boldsymbol{y}}$ as faithful as possible to the original operation $\boldsymbol{y}$, we use optimal transform maps (Santambrogio, 2015; Villani, 2003; 2008) to find a minimal modification of $\boldsymbol{y}$ which recovers the prior distribution $\boldsymbol{z}$.

This is illustrated in Figure 1a, where each point $\tilde{\boldsymbol{y}}$ of the matched curve is obtained by warping a corresponding point $\boldsymbol{y}$ of the linear trajectory, while not deviating too far from the line.

2 From Distribution mismatch to optimal transport

With an implicit models such as GANs (Goodfellow et al., 2014) and VAEs (Kingma & Welling, 2013), we use the data $\mathcal{X}$, drawn from an unknown random variable $\boldsymbol{x}$, to learn a generator $G : \mathbb{R}^d \mapsto \mathbb{R}^{d'}$ with respect to a fixed prior distribution $p_{\boldsymbol{z}}$, such that $G(\boldsymbol{z})$ approximates $\boldsymbol{x}$. Once the model is trained, we can sample from it by feeding latent samples $\boldsymbol{z}$ through G.

We now bring our attention to *operations* on latent samples $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_k$ from $p_{\boldsymbol{z}}$, i.e. mappings

$$\kappa : \mathbb{R}^d \times \cdots \times \mathbb{R}^d \to \mathbb{R}^d. \tag{1}$$

We remind the reader that samples of a random variable are also random variables, which we can then *realize* into to vectors $z_1, \cdots, z_k \in \mathbb{R}^d$. Thus, $\boldsymbol{y} = \kappa(\boldsymbol{z}_1, \cdots, \boldsymbol{z}_k)$ is a random variable (commonly referred to as a *statistic*) which can be numerically evaluated on a realization $y = \kappa(z_1, \cdot, z_k)$.

Therefore, our analysis on operations on samples is done through the underlying random variable $\boldsymbol{y}$. The same treatment is typically used to analyze other statistics over random variables, such as the sample mean, sample variance and test statistics.

In Table 1 we show example operations which have been commonly used in the literature. As discussed in the Introduction, such operations can provide valuable insight into how the trained generator G changes as one creates related samples $\boldsymbol{y}$ from some source samples. The most common such operation is the linear interpolation, which we can view as a operation

$$\boldsymbol{y}_t = t\boldsymbol{z}_1 + (t-1)\boldsymbol{z}_2, \tag{2}$$

where $\boldsymbol{z}_1, \boldsymbol{z}_2$ latent samples from the prior $p_{\boldsymbol{z}}$ and $\boldsymbol{y}_t$ is parameterized by $t \in [0, 1]$.

Now, assume $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are i.i.d, and let $Z_1, Z_2 \sim Z$ be their (scalar) first components. Then the first component of $\boldsymbol{y}_t$ is $Y_t = tZ_1 + (t-1)Z_2$, and we can compute:

$$Var[Y_t] = Var[tZ_1 + (1-t)Z_2] = t^2 Var[Z_1] + (1-t)^2 Var[Z_2] = (1 + 2t(t-1))Var[Z]. \tag{3}$$

Since $(1 + 2t(t-1)) \neq 1$ for all $t \in]0, 1[$, it is in general impossible for $\boldsymbol{y}_t$ to have the same distribution as $\boldsymbol{z}$, which means that distribution mismatch is *inevitable* when using linear interpolation. A similar analysis reveals the same for all of the operations in Table 1.

This is problematic: since the generator G is only trained for samples from $p_{\boldsymbol{z}}$, we have no guarantees on how suitable $\boldsymbol{y}_t$ is as an input for G.

This leaves us with a dilemma: we have various intuitive operations (see Table 1) which we would want to be able to perform on samples, but their resulting distribution happens to be inconsistent with the distribution we trained G for.

2.1 On the possibly paradoxical relationship between the likelihood and the loss

It is obvious that we cannot expect a model G to perform well outside the domain it was trained on, but what if we are still within the support of the original training distribution?

Since we might train on a distribution with infinitely large support (e.g. gaussian in $\mathbb{R}^d$), but visit some regions with vanishingly small probability, clearly we cannot expect G to perform well on the entire support.

But we could use a uniform distribution for $\boldsymbol{z}$ (say on $[-1, 1]^d$), such that every infinitesimal region of the support is sampled with equal probability. Now, assuming the trained model is smooth, we might assume that this ensures the model performs similarly well everywhere in the support.

Unfortunately however, this is *not guaranteed* and turns out also to not be the case in practice.

For a concrete example, suppose we want to learn an unknown mapping $f : \mathbb{R}^d \to \mathbb{R}$ over a prior distribution $p_{\boldsymbol{z}}$ using linear regression with respect to an l_2 loss:

$$\mathbf{H} = \arg\min_{\mathbf{H}'} E_{\boldsymbol{z} \sim p_{\boldsymbol{z}}}[l_2(f(z), \mathbf{H}'\boldsymbol{z})], \tag{4}$$

with $l_2(f(z), \mathbf{H}'\boldsymbol{z}) = \|f(\boldsymbol{z}) - \mathbf{H}'\boldsymbol{z}\|^2$.

Now, consider the case where f and $p_{\boldsymbol{z}}(z)$ are spherically symmetric, such that $f(z) = g(\|z\|)$ and $p_{\boldsymbol{z}}(z) = s(\|z\|)$ (e.g. $f(\boldsymbol{z}) = sin(\|z\|$ and $\boldsymbol{z}$ drawn from an isotropic gaussian, i.e. $p_{\boldsymbol{z}}(z) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with I as identity. In this case, due to symmetry, it is easy to see that $\mathbf{H} = \mathbf{0}$, and thus the pointwise loss is $l_2 = \|g(\|z\|)\|^2$.

Since we make no assumptions on g (except measurability), at a given distance $\|z\| = r$, we can have an arbitrary relationship between the likelihood of a point in the prior, $s(r)$, and the corresponding pointwise loss of the optimal regressor. We can therefore make the following observations:

- A uniform prior (e.g. uniform on the sphere $\{z \in \mathbb{R}^d : \|z\| < R\}$), where $s(r)$ is constant, does not give any guarantee of a uniformly distributed pointwise loss. Thus, even for a uniform prior, we cannot expect the model to perform equally well everywhere on the support.
- We can not expect the loss l_2 to be low for regions that have a high likelihood. E.g. for $g = s$ the inverse relationship holds. The intuition for this is that minimizing $l_2(z)$ may be 'harder' for regions of high probability, such that the resulting loss will still be higher than elsewhere in the space, even though regions of high probability are 'more important' since we are minimizing the expected loss.

A particular example is when $\boldsymbol{z}$ is drawn from an isotropic Gaussian $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $g(r) = 1/r^2$. In this case, if we look at the midpoint of linear interpolation, $\boldsymbol{y} = \frac{1}{2}\boldsymbol{z}_1 + \frac{1}{2}\boldsymbol{z}_2$, we have $\boldsymbol{y} \sim \mathcal{N}(\mathbf{0}, \frac{1}{2}\mathbf{I})$. Now, both $\boldsymbol{z}$ and $\boldsymbol{y}$ have spherically symmetric distributions with likelihoods decaying exponentally with r. However, the "radius" of $\boldsymbol{y}$ is smaller by a factor of $\sqrt{1}2$ compared to $\boldsymbol{z}$, which means that points drawn from $\boldsymbol{y}$ will have a higher likelihood when evaluated at the density of $p_{\boldsymbol{z}}$, i.e.

$$E_{\boldsymbol{y}}[p_{\boldsymbol{z}}(\boldsymbol{y})] \geq E_{\boldsymbol{z}}[p_{\boldsymbol{z}}(\boldsymbol{z})]. \tag{5}$$

So while the points of $\boldsymbol{y}$ are on average 'high likelihood' points in the distribution of $\boldsymbol{z}$, we have for the loss l_2:

$$E_{\boldsymbol{y}}[l_2(\boldsymbol{y}, f(\boldsymbol{y}))] = E_{\boldsymbol{y}}[g(\|\boldsymbol{y}\|)] \geq E_{\boldsymbol{z}}[g(\|\boldsymbol{z}\|)] = E_{\boldsymbol{y}}[l_2(\boldsymbol{y}, f(\boldsymbol{y}))] \tag{6}$$

.

So therefore, it is clear that in the general case, we have no guarantees on the relationship between the likelihood of points in the training distribution $p_{\boldsymbol{z}}$ and the expected loss when using a different distribution $\boldsymbol{y}$, even in the case where $\boldsymbol{y}$ is concentrated at regions of 'high probability' in $p_{\boldsymbol{z}}$.

While we have made these observations based on a simple setting (linear regression, l_2 loss and symmetry constraints), we will give a heuristic reasoning why this might be happening in practice, while experimentally also showing that this is the case.

To this end, in the next section we will give a short example of *the curse of dimensionality* and discuss how the points of the prior distribution $p_{\boldsymbol{z}}$ are concentrated in a thin spherical shell if it has i.i.d components.

We note that the analysis of the next section can bee seen as a more rigorous version of an observation made by White (2016), who experimentally shows that there is a significant difference between the average norm of the midpoint of linear interpolation and the points of the prior, for uniform and Gaussian distributions.

2.2 On the curse of dimensionality and geometric outliers

Suppose our latent space has a prior with $\boldsymbol{z} = [Z_1, \cdots, Z_d] \in [-1, 1]^d$ with i.i.d entries $Z_i \sim Z$. In this case, we can look at the squared norm

$$\|\boldsymbol{z}\|^2 = \sum_{i=1}^{d} Z_i^2. \tag{7}$$

From the Central Limit Theorem (CLT), we know that as $d \to \infty$,

$$\sqrt{d}(\frac{1}{d}\|\boldsymbol{z}\|^2 - \mu_{Z^2}) \to \mathcal{N}(0, \sigma_{Z^2}^2), \tag{8}$$

in distribution. Thus, assuming d is large enough such that we are close to convergence, we can approximate the distribution of $\|\boldsymbol{z}\|^2$ as $\mathcal{N}(d\mu_{Z^2}, d\sigma_{Z^2}^2)$. In particular, this implies that almost all points lie on a relatively thin spherical shell, since the mean grows as $d\mu_{Z^2}$ whereas the standard deviation grows only as $\sqrt{d}\sigma_{Z^2}$.

Now consider an operator such as the midpoint of linear interpolation, $\boldsymbol{y} = \frac{1}{2}\boldsymbol{z}_1 + \frac{1}{2}\boldsymbol{z}_2$, with components $Y^{(i)} = \frac{1}{2}Z_1^{(i)} + \frac{1}{2}Z_2^{(i)}$. Furthermore, let's assume Z is symmetric around 0, such that $E[Z] = 0$.

In this case, we can compute:

$$E[(Y^{(i)})^2] = Var[\frac{1}{2}Z_1 + \frac{1}{2}Z_2] = \frac{1}{2}Var[Z] = \frac{1}{2}\mu_{Z^2}^2 \tag{9}$$

$$Var[(Y^{(i)})^2] = Var[(\frac{1}{2}Z_1 + \frac{1}{2}Z_2)^2] = \frac{1}{4}Var[Z^2] = \frac{1}{4}\sigma_{Z^2}^2. \tag{10}$$

Thus, the distribution of $\|\boldsymbol{y}\|^2$ can be approximated with $\mathcal{N}(\frac{1}{2}d\mu_{Z^2}, \frac{1}{4}d\sigma_{Z^2}^2)$.

Therefore, $\boldsymbol{y}$ also mostly lies on a spherical shell, but with a different radius than $\boldsymbol{z}$. In fact, the shells will intersect at regions which have a vanishing probability for large d. In other words, when looking at the squared norm $\|\boldsymbol{y}\|^2$, $\|\boldsymbol{y}\|^2$ is a (strong) out-lier with respect to the distribution of $\|\boldsymbol{z}\|^2$.

While we already made the point that likelihood is not a guaranteed indicator for a good pointwise loss, the norm is a scalar statistic which one can imagine that a learned generator G specializes with respect to. In Figure 3 in the experiments, we illustrate how dramatically different the distributions are, having almost no common support, when using a uniformly distributed prior space in 100 dimensions.

2.3 DISTRIBUTION MATCHING WITH OPTIMAL TRANSPORT

We have seen that distribution mismatch appears with common operators, such as linear interpolation. Furthermore, we have seen that the result can cause a dramatic increase in the expected loss for the generator G, because of the mismatched training distribution.

In order to address this, we propose a simple and intuitive strategy for constructing distribution preserving operators, via optimal transport:

Strategy 1 (Optimal Transport Matched Operations)**.**

1. *We construct an 'intuitive' operator $\boldsymbol{y} = \kappa(\boldsymbol{z}_1, \cdots, \boldsymbol{z}_k)$.*
2. *We analytically (or numerically) compute the resulting (mismatched) distribution $p_{\boldsymbol{y}}$*
3. *We search for a minimal modification $\tilde{\boldsymbol{y}} = f(\boldsymbol{y})$ (in the sense that $E_{\boldsymbol{y}}[c(\tilde{\boldsymbol{y}}, \boldsymbol{y})]$ is minimal with respect to a cost c), such that distribution is brought back to the prior, i.e. $p_{\tilde{\boldsymbol{y}}} = p_{\boldsymbol{z}}$.*

The cost function in step 3 could e.g. be the euclidean distance $c(x, y) = \|x - y\|$, and is used to measure how faithful the modified operator, $\tilde{\boldsymbol{y}} = f(k(\boldsymbol{z}_1, \cdots, \boldsymbol{z}_k))$ is to the original operator k. Finding the map f which gives a minimal modifications can be challenging, but fortunately it is a well studied problem from optimal transport theory. We refer to the modified operation $\tilde{y}$ as the *distribution matching* of $\boldsymbol{y}$, with respect to the cost c and prior distribution $p_{\boldsymbol{z}}$.

For completeness, we introduce key concept of optimal transport theory in a simplified setting, i.e. assuming probability distributions are in euclidian space and skipping measure theoretical formalism. We refer to Villani (2003; 2008) and Santambrogio (2015) for a thorough and formal treatment of optimal transport.

The problem of step (3) above was first posed byMonge (1781) and can more formally be stated as:

Problem 1 (Santambrogio (2015) Problem 1.1)**.** *Given probability distributions $p_{\boldsymbol{x}}, p_{\boldsymbol{y}}$, with domains $\mathcal{X}, \mathcal{Y}$ respectively, and a cost function $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$, we want to minimize*

$$inf \left\{ E_{\boldsymbol{x} \sim p_{\boldsymbol{x}}}[c(\boldsymbol{x}, f(\boldsymbol{x}))] \middle| f : \mathcal{X} \to \mathcal{Y}, f(\boldsymbol{x}) \sim p_{\boldsymbol{y}} \right\} \tag{MP}$$

We refer to the minimizer $f^* \mathcal{X} \to \mathcal{Y}$ of (MP) *(if it exists), as the optimal transport map from $p_{\boldsymbol{x}}$ to $p_{\boldsymbol{y}}$ with respect to the cost c.*

However, the problem remained unsolved until a relaxed problem was studied by Kantorovich (1942):

Problem 2 (Santambrogio (2015) Problem 1.2)**.** *Given probability distributions $p_{\boldsymbol{x}}, p_{\boldsymbol{y}}$, with domains $\mathcal{X}, \mathcal{Y}$ respectively, and a cost function $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$, we want to minimize*

$$inf \left\{ E_{(\boldsymbol{x}, \boldsymbol{y}) \sim p_{\boldsymbol{x}, \boldsymbol{y}}}[c(\boldsymbol{x}, \boldsymbol{y})] \middle| (\boldsymbol{x}, \boldsymbol{y}) \sim p_{\boldsymbol{x}, \boldsymbol{y}}, \boldsymbol{x} \sim p_{\boldsymbol{x}}, \boldsymbol{y} \sim p_{\boldsymbol{y}} \right\} \tag{KP}$$

We refer to the joint distribution $p_{\boldsymbol{x}, \boldsymbol{y}}$ which minimizes (KP) *as the optimal transport plan from $p_{\boldsymbol{x}}$ to $p_{\boldsymbol{y}}$ with respect to the cost c.*

The key difference is to relax the deterministic relationship between $\boldsymbol{x}$ and $f(\boldsymbol{x})$ to a joint probability distribution $p_{\boldsymbol{x}, \boldsymbol{y}}$ with marginals $p_{\boldsymbol{x}}$ and $p_{\boldsymbol{y}}$ for $\boldsymbol{x}$ and $\boldsymbol{y}$. In the case of Problem 1, the minimization might be over the empty set since it is not guaranteed that there exists a mapping f such that $f(\boldsymbol{x}) \sim \boldsymbol{y}$. In contrast, for Problem 2, one can always construct a joint density $p_{\boldsymbol{x}, \boldsymbol{y}}$ with $p_{\boldsymbol{x}}$ and $p_{\boldsymbol{y}}$ as marginals, such as the trivial construction where $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent, i.e. $p_{\boldsymbol{x}, \boldsymbol{y}}(x, y) := p_{\boldsymbol{x}}(x) p_{\boldsymbol{y}}(y)$.

We note that we can treat a joint density $p_{\boldsymbol{x}, \boldsymbol{y}}(x, y)$ over $\mathcal{X} \times \mathcal{Y}$ as a stochastic mapping from $\mathcal{X}$ to $\mathcal{Y}$, where given a fixed x, we obtain values $f(x)$ by sampling from the conditional distribution, i.e.

$f(x) \sim p_{\boldsymbol{y}|x}$. Thus, we can view the Problem KP as a relaxation of Problem MP where f is allowed to be a stochastic mapping.

While the relaxed problem of Kantorovich, (KP) is much more studied in the optimal transport literature, for our purposes of constructing operators it is desirable for the mapping f to be deterministic as in (MP).

To this end, we will choose the cost function c such that the two problems coincide and where we can find an analytical solution f or at least an efficient numerical solution.

In particular, we note that most operators in Table 1 are all *pointwise*, such that if the points $\boldsymbol{z}_i$ have i.i.d. components, then the result $\boldsymbol{y}$ will also have i.i.d. components.

Now, if we combine this with a constraint the cost c to be additive over the components of $\boldsymbol{x}, \boldsymbol{y}$, we obtain the following simplification:

Theorem 1. *Suppose $p_{\boldsymbol{x}}$ and $p_{\boldsymbol{y}}$ have i.i.d components and c over $\mathcal{X} \times \mathcal{Y} = \mathbb{R}^d \times \mathbb{R}^d$ decomposes as*

$$c(x, y) = \sum_{i=1}^{d} C(x^{(i)}, y^{(i)}), . \tag{11}$$

Then, the minimization problems (MP) *and* (KP) *turn into d identical scalar problems for the distributions p_X and p_Y of the components of $\boldsymbol{x}$ and $\boldsymbol{y}$:*

$$inf \left\{ E_{X \sim p_X}[C(X, T(X))] \middle| T : \mathbb{R} \to \mathbb{R}, T(X) \sim p_Y \right\} \tag{MP-1-D}$$

$$inf \left\{ E_{(X,Y) \sim p_{X,Y}}[C(X, Y)] \middle| (X, Y) \sim p_{X,Y}, X \sim p_X, Y \sim p_Y \right\}, \tag{KP-1-D}$$

such that an optimal transport map T for (MP-1-D) *gives an optimal transport map f for* (MP) *by pointwise application of T, i.e. $f(x)^{(i)} := T(x^{(i)})$, and an optimal transport plan $p_{X,Y}$ for* (KP-1-D) *gives an optimal transport plan $p_{\boldsymbol{x},\boldsymbol{y}}(x, y) := \prod_{i=1}^{d} p_{X,Y}(x^{(i)}, y^{(i)})$ for* (KP).

Proof. See Appendix. □

Fortunately, under some mild constraints, the scalar problems have a known solution:

Theorem 2 (Theorem 2.9 in Santambrogio (2015))**.** *Let $h : \mathbb{R} \to \mathbb{R}^+$ be convex and suppose the cost C takes the form $C(x, y) = h(x - y)$. Given an continuous source distribution p_X and a target distribution p_Y on $\mathbb{R}$ having a finite optimal transport cost in* (KP-1-D)*, then*

$$T^{mon}_{X \to Y}(x) := F_Y^{[-1]}(F_X(x)), \tag{12}$$

defines an optimal transport map from p_X to p_Y for (MP-1-D)*, where $F_X(x) := \int_{-\infty}^{x} p_X(x')dx'$ is the Cumulative Frequencey Distribution (CDF) of X and $F_Y^{[-1]}(y) := \inf\{t \in \mathbb{R} | F_Y(t) \geq t\}$ is the pseudo-inverse of F_Y. Furthermore, the joint distribution of $(X, T^{mon}_{X \to Y}(X))$ defines an optimal transport plan for* (KP-1-D)*.*

The mapping $T^{\text{mon}}_{X \to Y}(x)$ in Theorem 2 is non-decreasing and is known as the *monotone transport map* from X to Y. It is easy to verify that $T^{\text{mon}}_{X \to Y}(X)$ has the distribution of X, in particular $F_X(X) \sim \text{Uniform}(0, 1)$ and if $U \sim \text{Uniform}(0, 1)$ then $F_Y^{[-1]}(U) \sim Y$. These two properties happen to uniquely define the monotone transport map,

Lemma 1 (Theorem 2.5 in Santambrogio (2015))**.** *Suppose a mapping $g(x)$ is non-decreasing and maps a continious distribution p_X to a distribution p_Y, i.e.*

$$g(X) \sim Y, \tag{13}$$

then g is the monotone transport map $T^{mon}_{X \to Y}$.

According to Lemma 1, an alternative way of computing $T^{\text{mon}}_{X \to Y}$ is to find some g that is non-decreasing and transforms p_X to p_Y.

Now, combining Theorems 1 and 2, we obtain a concrete realization of the Strategy 1 outlined above. We choose the cost c such that it admits to Theorem 1, such as $c(\boldsymbol{x}, \boldsymbol{y}) := \|\boldsymbol{x} - \boldsymbol{y}\|_1$, and use an

operation that is pointwise, we just need to compute the monotone transport map in (12). That is, if $\boldsymbol{z}$ has i.i.d components with distribution p_Z, we need to compute the compnent distribution p_Y of the result $\boldsymbol{y}$ of the operation, the CDFs F_Z, F_Y and obtain

$$T^{\text{mon}}_{Y\to Z}(y) := F_Z^{[-1]}(F_Y(y)) \tag{14}$$

as the component-wise modification of $\boldsymbol{y}$, i.e. $\tilde{\boldsymbol{y}}^{(i)} := T^{\text{mon}}_{Y\to Z}(\boldsymbol{y}^{(i)})$.

In the next sections, we illustrate this for a few examples, in particlar for linear interpolation and vicinity sampling, using a uniform or a gaussian prior. We picked the examples where we can analytically compute the uniform transport map, but note that it is also easy to compute $F_Z^{[-1]}$ and $(F_Y(y))$ numerically, since one only needs to estimate CDFs in one dimension.

Since the components of all random variables in these examples are i.i.d, for such a random vector $\boldsymbol{x}$ we will implicitly write X for a scalar random variable that has the distribution of the components of $\boldsymbol{x}$.

2.4 Example 1:Uniform Linear Interpolation

Now suppose $\boldsymbol{z}$ has uniform components $\mathbb{Z} \sim \text{Uniform}(-1, 1)$. In this case, $p_Z(z) = 1/2$ for $-1 < z < 1$.

Now let $\boldsymbol{y}_t = t\boldsymbol{z}_1 + (1-t)\boldsymbol{z}_2$ denote the linear interpolation between two points $\boldsymbol{z}_1, \boldsymbol{z}_2$. We then obtain the component distribution p_{Y_t} as the convolution of p_{tZ} and $p_{(1-t)Z}$. Due to symmetry we can assume that $t > 1/2$, since $p_{Y_t} = p_{Y_{1-t}}$.

First we note that $p_{tZ} = 1/(2t)$ for $-t < z < t$ and $p_{(1-t)Z} = 1/(2(1-t))$ for $-(1-t) < z < 1-t$. We then obtain, via convolution:

$$p_{Y_t}(y) = \frac{1}{2(1-t)(2t)} \begin{cases} 0 & \text{if } y < -1 \\ y+1 & \text{if } -1 < y < -t + (1-t) \\ 2-2t & \text{if } -t + (1-t) < y < t - (1-t) \\ -y+1 & \text{if } t - (1-t) < y < 1 \\ 0 & \text{if } 1 < y \end{cases} \tag{15}$$

$$\tag{16}$$

Now, we obtain the CDF F_{Y_t} by computing

$$F_{Y_t}(y) = \int_{-\infty}^{y} p_{Y_t}(y')dy' \tag{17}$$

$$= \frac{1}{2(1-t)(2t)} \begin{cases} 0 & \text{if } y < -1 \\ \frac{1}{2}(y+1)(y+1) & \text{if } -1 < y < 1-2t \\ 2(1-t)(y+t) & \text{if } 1-2t < y < 2t-1 \\ 2(1-t)(3t-1) + (-\frac{1}{2}y^2 + y + \frac{1}{2}(2t-1)^2 - (2t-1)) & \text{if } 2t-1 < y < 1 \\ 2(1-t)(2t) & \text{if } 1 < y \end{cases} \tag{18}$$

Now since $p_Z(z) = 1/2$ for $|z| < 1$, we have $F_Z(z) = \frac{1}{2}z + \frac{1}{2}$ for $|z| < 1$. This gives $F_Z^{[-1]}(p) = 2(p - \frac{1}{2})$.

Now, we just compose the two mappings to obtain $T^{\text{mon}}_{Y_t\to Z}(y) = F_Z^{[-1]}(F_{Y_t}(y))$.

3 Example 2: Uniform Vicinity Sampling

Let $\boldsymbol{z}$ again have uniform components on $[-1, 1]$. For vicinity sampling, we want to obtain new points $\boldsymbol{z}'_1, \cdot, \boldsymbol{z}'_k$ which are close to $\boldsymbol{z}$. We thus define

$$\boldsymbol{z}'_i := \boldsymbol{z} + \epsilon \boldsymbol{u}_i, \tag{19}$$

where $\boldsymbol{u}_i$ also has uniform compnents, such that each coordinate of $\boldsymbol{z}_i'$ differs at most by ϵ from $\boldsymbol{z}$. By identifying $tZ_i' = tZ + (1-t)U_i$ with $t = 1/(1+\epsilon)$, we see that tZ_i' has identical distribution to the linear interpolation Y_t in the previous example. Thus $g_t(Z_i') := T^{\text{mon}}_{Y_t \to Z}(tZ_i')$ will have the distribution of Z, and by Lemma1 is then the monotone transport map from Z_i' to Z.

3.1 Example 3: Gaussian Linear Interpolation, Vicinity Sampling and Analogies

Suppose $\boldsymbol{z}$ has components $Z \sim \mathcal{N}(0, \sigma^2)$. In this case, we can compute linear interpolation as before, $\boldsymbol{y}_t = t\boldsymbol{z}_1 + (1-t)\boldsymbol{z}_2$. Since the sum of gaussians is gaussian, we get, $Y_t \sim \mathcal{N}(0, t^2\sigma^2 + (1-t)^2\sigma^2)$. Now, it is easy to see that with a proper scaling factor, we can adjust the variance of Y_t back to σ^2. That is, $\frac{1}{\sqrt{t^2+(1-t)^2}} Y_t \sim \mathcal{N}(0, \sigma^2)$, so by Lemma 1 $g_t(y) := \frac{1}{\sqrt{t^2+(1-t)^2}} y$ is the monotone transport map from Y_t to Z.

By adjusting the vicinity sampling operation to

$$\boldsymbol{z}_i' := \boldsymbol{z} + \epsilon \mathbf{e}_i, \tag{20}$$

where $\mathbf{e}_i \sim \mathcal{N}(0, 1)$, we can similarly find the monotone transport map $g_\epsilon(y) = \frac{1}{\sqrt{1+\epsilon^2}} y$.

Another operation which has been used in the literature is the "analogy", where from samples $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3$, one wants to apply the difference between $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, to $\boldsymbol{z}_3$. The transport map is then $g(y) = \frac{1}{\sqrt{3}} y$

4 Experiments

4.1 Comparison of distributions

To validate the correctness of the matched operators obtained above, we numerically simulate the distributions for toy examples, as well as prior distributions typically used to train Generative Models.

For Figure 1, we sample 1 million pairs of points uniformly on $[-1, 1]^2$, and estimate numerically the midpoint distribution of linear interpolation, our proposed matched interpolation and the spherical interpolation of White (2016). It is reassuring to see that the matched interpolation gives midpoint which are also uniformly distributed. In contrast, the linear interpolation condenses more towards the origin, forming a pyramid-shaped distribution (the result of convolving two boxes in 2-d). Since the spherical interpolation of White (2016) follows a great circle with varying radius between the two points, we see that the resulting distribution has a 'hole' in it since it always circles around the origin.

For Figure 3, we sample 100000 points in $d = 100$ dimensions, using either i.i.d. uniform components on $[-1, 1]$ or Gaussian $\mathcal{N}(0, 1)$. We see there is a dramatic difference between the prior and We also show the spherical (SLERP) interpolation of White (2016) which has a matching first moment, but otherwise also induces a distribution mismatched. In contrast, our matched interpolation, which we derive in the next section, fully preserves the prior distribution and perfectly aligns.

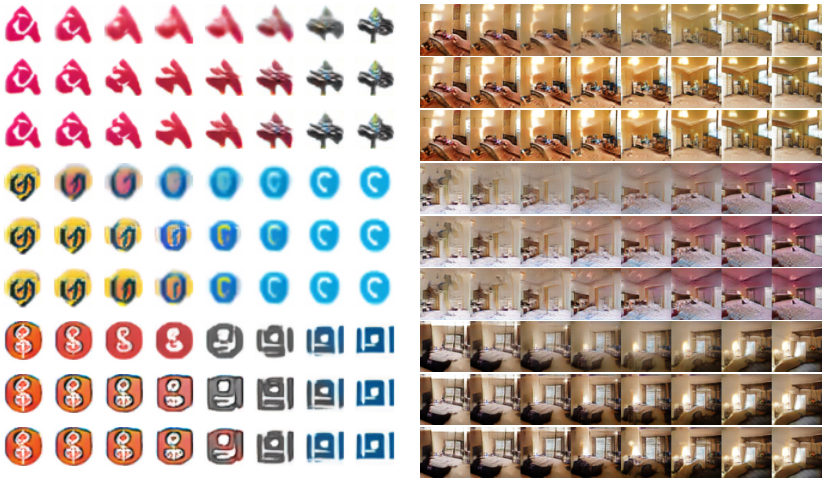
4.2 Qualitative results

In this section we will present some concrete examples for the differences in generator output dependent on the exact operation used to traverse the latent space. To this end, the output for linear interpolation, SLERP (spherical linear interpolation) and our proposed matched transport interpolation will be compared.

We used DCGAN models trained for LSUN and an icon dataset (https://data.vision.ee.ethz.ch/cvl/lld/) to qualitatively evaluate.

We begin with the classic example of 2-point interpolation: Figure 2 shows three examples each for an interpolation between 2 points in latent space. Each example is first done via linear interpolation, then SLERP and finally matched interpolation.

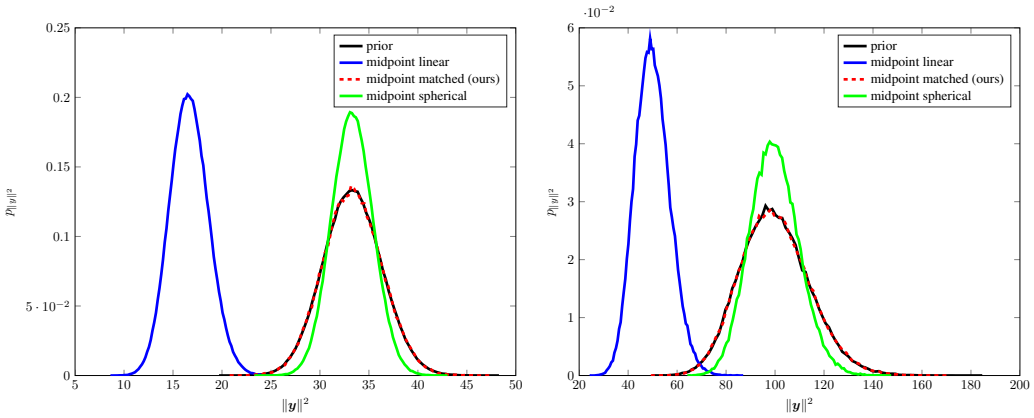
It is immediately obvious that linear interpolation produces inferior results with generally more blurry, less saturated and less detailed output images. SLERP and matched interpolation are slightly different, however it is not obvious which is superior.



(a) Icon dataset (b) LSUN dataset

Figure 2: 2-point interpolation: Each example shows linear, SLERP and transport matched interpolation from top to bottom respectively. Both outputs are produced with DCGAN trained on 100-dimensional, uniformly distributed latent vectors. The output resolution for the icon dataset 2a is 32, for 2b 64 pixel.

An even stronger effect can be seen when we do 4-point interpolation, showcased in Figures 4 (LSUN) and 5 (icons). Because the outputs where generated at a higher resolution, we can observe a slight loss in detain for the SLERP version here as well. Most details are lost and artifacts start do dominate the image towards the midpoint for linear interpolation. The increase in artifacts is also very apparent in figures



(a) uniform distribution (b) Gaussian distribution

Figure 3: We the distribution of the squared norm $\|\boldsymbol{y}\|^2$ of midpoints for two prior distributions in 100 dimensions: (a) components uniform on $[-1, 1]$ and (b) components Gaussian $\mathcal{N}(0, 1)$, for linear interpolation, our proposed matched interpolation and the spherical interpolation of White (2016). Both linear interpolation and spherical interpolation introduce a distribution mismatch, whereas our proposed matched interpolation preserves the prior distribution for both priors.

(a) Linear interpolation (b) spherical interpolation (c) our matched interpolation

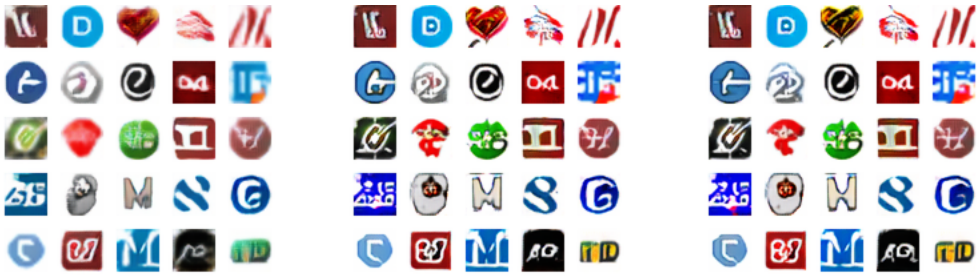
Figure 4: Interpolation from 4 sampled points from LSUN dataset. The 4 sampled points are in the corners while the middle sampled point is in the middle of the 5×5 grid for each of the interpolation methods considered: linear, SLERP and our proposed uniform-matched interpolation.

(a) Linear interpolation (b) SLERP interpolation (c) our matched interpolation

Figure 5: Interpolation between 4 random points on the icon dataset, at 32x32 pixel. Similar effects as for the LSUN example are noticable.

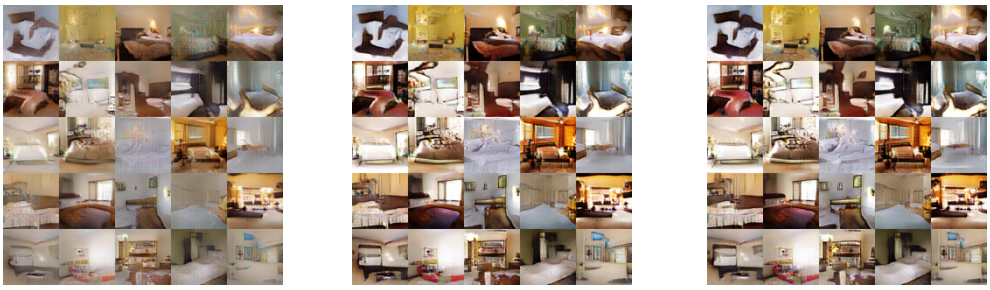
5 Conclusions

We have shown that the common latent space operations used for Generative Models induce distribution mismatch from the prior distribution the models were trained for. This problem has been ignored by the literature so far, partially due to belief that this should not be a problem for uniform priors. However, our statistical and experimental analysis shows that the problem is real, with the operations used so far producing significantly lower quality samples compared to their inputs. To address the distribution mismatch, we propose to use optimal transport to minimally modify (in l_1 distance) the operations such that they fully preserve the prior distribution. We give analytical formulas of the resulting (matched) operations for various examples, which are easily implemented. The matched operators give a significantly higher quality samples compared to the originals, having the potential to become standard tools for evaluating and exploring generative models.



(a) Linear interpolation (b) SLERP interpolation (c) matched interpolation

Figure 6: Midpoints sampling for linear SLERP and uniform-matched interpolation when starting from the same pairs of samples points on icon dataset.



(a) Linear interpolation (b) SLERP interpolation (c) matched interpolation

Figure 7: Midpoints sampling for linear SLERP and uniform-matched interpolation when starting from the same pairs of samples points on LSUN.

REFERENCES

Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. Better mixing via deep representations. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 552–560, 2013.

Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016.

Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1538–1546, 2015.

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

Leonid Vitalievich Kantorovich. On the translocation of masses. In *Dokl. Akad. Nauk SSSR*, volume 37, pp. 199–201, 1942.

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

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc., 2013. URL `http://papers.nips.cc/paper/`

5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, 1781.

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

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 833–840, 2011.

Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 2015.

Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

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

Tom White. Sampling generative networks. *arXiv preprint arXiv:1609.04468*, 2016.

6 APPENDIX

6.1 PROOF OF THEOREM 1

Proof. We will show it for the Kantorovich problem, the Monge version is similar.

Starting from (KP), we compute

$$inf\left\{E_{(\boldsymbol{x},\boldsymbol{y})\sim p_{\boldsymbol{x},\boldsymbol{y}}}[c(\boldsymbol{x},\boldsymbol{y})]\middle|(\boldsymbol{x},\boldsymbol{y})\sim p_{\boldsymbol{x},\boldsymbol{y}},\boldsymbol{x}\sim p_{\boldsymbol{x}},\boldsymbol{y}\sim p_{\boldsymbol{y}}\right\} \tag{21}$$

$$= inf\left\{E_{(\boldsymbol{x},\boldsymbol{y})\sim p_{\boldsymbol{x},\boldsymbol{y}}}[\sum_{i=1}^{d} C(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)})]\middle|(\boldsymbol{x},\boldsymbol{y})\sim p_{\boldsymbol{x},\boldsymbol{y}},\boldsymbol{x}\sim p_{\boldsymbol{x}},\boldsymbol{y}\sim p_{\boldsymbol{y}}\right\} \tag{22}$$

$$= inf\left\{\sum_{i=1}^{d} E_{(\boldsymbol{x},\boldsymbol{y})\sim p_{\boldsymbol{x},\boldsymbol{y}}}[C(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)})]\middle|(\boldsymbol{x},\boldsymbol{y})\sim p_{\boldsymbol{x},\boldsymbol{y}},\boldsymbol{x}\sim p_{\boldsymbol{x}},\boldsymbol{y}\sim p_{\boldsymbol{y}}\right\} \tag{23}$$

$$\geq \sum_{i=1}^{d} inf\left\{E_{(\boldsymbol{x},\boldsymbol{y})\sim p_{\boldsymbol{x},\boldsymbol{y}}}[C(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)})]\middle|(\boldsymbol{x},\boldsymbol{y})\sim p_{\boldsymbol{x},\boldsymbol{y}},\boldsymbol{x}\sim p_{\boldsymbol{x}},\boldsymbol{y}\sim p_{\boldsymbol{y}}\right\} \tag{24}$$

$$= \sum_{i=1}^{d} inf\left\{E_{(X,Y)\sim p_{X,Y}}[C(X,Y)]\middle|(X,Y)\sim p_{X,Y},X\sim p_X,Y\sim p_Y\right\} \tag{25}$$

$$= d\cdot inf\left\{E_{(X,Y)\sim p_{X,Y}}[C(X,Y)]\middle|(X,Y)\sim p_{X,Y},X\sim p_X,Y\sim p_Y\right\}, \tag{26}$$

$$\tag{27}$$

where the inequality in (24) is due to each term being minimized separately.

Now let $\mathcal{P}_d(X,Y)$ be the set of joints $p_{\boldsymbol{x},\boldsymbol{y}}$ with $p_{\boldsymbol{x},\boldsymbol{y}}(x,y) = \prod_{i=1}^{d} p_{X,Y}(x^{(i)},y^{(i)})$ where $p_{X,Y}$ has marginals p_X and p_Y. In this case $\mathcal{P}_d(X,Y)$ is a subset of all joints $p_{\boldsymbol{x},\boldsymbol{y}}$ with marginals $p_{\boldsymbol{x}}$ and $p_{\boldsymbol{y}}$, where the pairs $(\boldsymbol{x}^{(1)},\boldsymbol{y}^{(1)}),\ldots,(\boldsymbol{x}^{(d)},\boldsymbol{y}^{(d)}))$ are constrained to be i.i.d. Starting again from (23) can compute:

$$inf\left\{\sum_{i=1}^{d} E_{(\boldsymbol{x},\boldsymbol{y})\sim p_{\boldsymbol{x},\boldsymbol{y}}}[C(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)})]\middle|(\boldsymbol{x},\boldsymbol{y})\sim p_{\boldsymbol{x},\boldsymbol{y}},\boldsymbol{x}\sim p_{\boldsymbol{x}},\boldsymbol{y}\sim p_{\boldsymbol{y}}\right\}$$

$$\leq inf\left\{\sum_{i=1}^{d} E_{(\boldsymbol{x},\boldsymbol{y})\sim p_{\boldsymbol{x},\boldsymbol{y}}}[C(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)})]\middle|p_{\boldsymbol{x},\boldsymbol{y}}\in\mathcal{P}_d(X,Y)\right\} \tag{28}$$

$$= inf\left\{\sum_{i=1}^{d} E_{(\boldsymbol{x},\boldsymbol{y})\sim p_{\boldsymbol{x},\boldsymbol{y}}}[C(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)})]\middle|p_{\boldsymbol{x},\boldsymbol{y}}\in\mathcal{P}_d(X,Y)\right\} \tag{29}$$

$$= inf\left\{\sum_{i=1}^{d} E_{(X,Y)\sim p_{X,Y}}[C(X,Y)]\middle|(X,Y)\sim p_{X,Y},X\sim p_X,Y\sim p_Y\right\} \tag{30}$$

$$= d\cdot inf\left\{E_{(X,Y)\sim p_{X,Y}}[C(X,Y)]\middle|(X,Y)\sim p_{X,Y},X\sim p_X,Y\sim p_Y\right\}, \tag{31}$$

$$\tag{32}$$

where the inequality in (28) is due to minimizing over a smaller set.

Since the two inequalities above are in the opposite direction, equality must hold for all of the expressions above, in particular:

$$inf\left\{E_{(\boldsymbol{x},\boldsymbol{y})\sim p_{\boldsymbol{x},\boldsymbol{y}}}[c(\boldsymbol{x},\boldsymbol{y})]\middle|(\boldsymbol{x},\boldsymbol{y})\sim p_{\boldsymbol{x},\boldsymbol{y}},\boldsymbol{x}\sim p_{\boldsymbol{x}},\boldsymbol{y}\sim p_{\boldsymbol{y}}\right\} \tag{33}$$

$$= d\cdot inf\left\{E_{(X,Y)\sim p_{X,Y}}[C(X,Y)]\middle|(X,Y)\sim p_{X,Y},X\sim p_X,Y\sim p_Y\right\} \tag{34}$$

Thus, (KP) and (KP-1-D) equal up to a constant, and minimizing one will minimize the other, and the minimization of the former can be done over $p_{X,Y}$ with $p_{\boldsymbol{x},\boldsymbol{y}}(x,y) = \prod_{i=1}^{d} p_{X,Y}(x^{(i)},y^{(i)})$. □