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 the original operations.

1 Introduction & Related Work

Generative models such as Variational Auto Encoders (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 $X$ we end up with a trained generator $G$, that is supposed to map latent samples $z$ from the fixed prior distribution to output samples $G(z)$ which (hopefully) have the same distribution as the data.

In order to understand and visualize the learned model $G(z)$, it is a common practice in the literature of generative models to explore how the output $G(z)$ behaves 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. We show that this, somewhat paradoxically, is also a problem if the support of resulting (mismatched) distribution is within the support of a uniformly distributed prior, whose points all have equal likelihood during training.

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, while minimally changing the original operation. 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. The points of the (red) matched trajectories...
Figure 1: We show examples of distribution mismatches induced by the previous interpolation schemes when using a uniform prior in two dimensions. Our matched interpolation avoids this with a minimal modification to the linear trajectory, traversing through the space such that all points along the path are distributed identically to the prior.

are obtained as minimal deviations (in expectation of $l_1$ distance) from the the points of the (blue) linear trajectory.

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.

Reed et al. (2016) use linear interpolation to explore their proposed GAN model which operates jointly in the visual and textual domain. Brock et al. (2016) combine GANs and VAEs for a neural photo editor, using masked interpolations to edit an embedded photo in the latent space.

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 \( y = tz_1 + (1-t)z_2, t \in [0,1] \). Since the distribution of \( y \) does not match the prior distribution of \( z \), we search for a warping \( f : \mathbb{R}^d \to \mathbb{R}^d \), such that \( \tilde{y} = f(y) \) has the same distribution as \( z \). In order to have the modification \( \tilde{y} \) as faithful as possible to the original operation \( y \), we use optimal transform maps (Santambrogio, 2015; Villani, 2003; 2008) to find a minimal modification of \( y \) which recovers the prior distribution \( z \).

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

## 2 From distribution mismatch to optimal transport

With 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 \( x \), to learn a generator \( G : \mathbb{R}^d \to \mathbb{R}^d \) with respect to a fixed prior distribution \( p_z \), such that \( G(z) \) approximates \( x \). Once the model is trained, we can sample from it by feeding latent samples \( z \) through \( G \).

We now bring our attention to operations on latent samples \( z_1, \ldots, z_k \) from \( p_z \), i.e. mappings

\[
\kappa : \mathbb{R}^d \times \cdots \times \mathbb{R}^d \to \mathbb{R}^d.
\]

We give a few examples of such operations in Table 1.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Expression</th>
<th>(Gaussian) Matched Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-point interpolation</td>
<td>( y = tz_1 + (1-t)z_2 ), ( t \in [0,1] )</td>
<td>( y = y/\sqrt{t^2 + (1-t)^2} )</td>
</tr>
<tr>
<td>( n )-point interpolation</td>
<td>( y = \sum_{i=1}^n t_i z_i ) with ( \sum_i t_i = 1 )</td>
<td>( y = y/\sqrt{\sum_{i=1}^n t_i^2} )</td>
</tr>
<tr>
<td>Vicinity sampling</td>
<td>( y_j = z_1 + cu_j ) for ( j = 1, \ldots, k )</td>
<td>( y_j = y_j/\sqrt{1+c^2} )</td>
</tr>
<tr>
<td>Analogies</td>
<td>( y = z_3 + (z_2 - z_1) )</td>
<td>( y = y/\sqrt{3} )</td>
</tr>
</tbody>
</table>

Table 1: Examples of interesting sample operations which need to be adapted if we want the distribution of the result \( y \) to match the prior distribution. If the prior is Gaussian, our proposed matched operation simplifies to a proper re-scaling factor (see third column) for additive operations.

**Under review as a conference paper at ICLR 2018**
2.1 Distribution Matching with Optimal Transport

In order to address the distribution mismatch, 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 \( y = \kappa(z_1, \ldots, z_k) \).
2. We analytically (or numerically) compute the resulting (mismatched) distribution \( p_y \).
3. We search for a minimal modification \( \tilde{y} = f(y) \) (in the sense that \( E_y[\text{cost}(\tilde{y}, y)] \) is minimal with respect to a cost \( c \)), such that distribution is brought back to the prior, i.e. \( p_{\tilde{y}} = p_z \).

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, \( \tilde{y} = f(\kappa(z_1, \ldots, z_k)) \) is to the original operator \( k \). Finding the map \( f \) which gives a minimal modification 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 matched version of \( y \), with respect to the cost \( c \) and prior distribution \( p_z \).

For completeness, we introduce the key concept of optimal transport theory in a simplified setting, i.e. assuming probability distributions are in euclidean 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 by Monge (1781) and can more formally be stated as:

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

\[
\inf \left\{ E_{x \sim p_x}[c(x, f(x))] \mid f : \mathcal{X} \to \mathcal{Y}, f(x) \sim p_y \right\} \quad \text{(MP)}
\]

We refer to the minimizer \( f^* : \mathcal{X} \to \mathcal{Y} \) of (MP) (if it exists), as the optimal transport map from \( p_x \) to \( p_y \) 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 \( p_x, p_y \), with domains \( X, Y \) respectively, and a cost function \( c : X \times Y \to \mathbb{R}^+ \), we want to minimize

\[
\inf \left\{ \mathbb{E}_{(x, y) \sim p_{x,y}} [c(x, y)] \mid (x, y) \sim p_{x,y}, x \sim p_x, y \sim p_y \right\}, \tag{KP}
\]

where \((x, y) \sim p_{x,y}, x \sim p_x \) denotes that \((x, y)\) have a joint distribution \( p_{x,y} \) which has (previously specified) marginals \( p_x \) and \( p_y \).

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

The key difference is to relax the deterministic relationship between \( x \) and \( f(x) \) to a joint probability distribution \( p_{x,y} \) with marginals \( p_x \) and \( p_y \) for \( x \) and \( y \). 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 \( f(x) \sim y \). In contrast, for Problem 2 one can always construct a joint density \( p_{x,y} \) with \( p_x \) and \( p_y \) as marginals, such as the trivial construction where \( x \) and \( y \) are independent, i.e. \( p_{x,y}(x, y) := p_x(x)p_y(y) \).

Note that given a joint density \( p_{x,y}(x, y) \) over \( X \times Y \), we can view \( y \) conditioned on \( x = x \) for a fixed \( x \) as a stochastic function \( f(x) \) from \( X \) to \( Y \), since given a fixed \( x \) do not get a specific function value \( f(x) \) but instead a random variable \( f(x) \) that depends on \( x \), with \( f(x) \sim y(x) = x \) with density \( p_y(y|x) := p_{x,y}(x, y)/p_x(x) \). In this case we have \((x, f(x)) \sim p_{x,y}\), so 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 \( z_i \) have i.i.d. components, then the result \( y \) will also have i.i.d. components.

If we combine this with the constraint for the cost \( c \) to be additive over the components of \( x, y \), we obtain the following simplification:

**Theorem 1.** Suppose \( p_x \) and \( p_y \) have i.i.d components and \( c \) over \( X \times Y = \mathbb{R}^d \times \mathbb{R}^d \) decomposes as

\[
c(x, y) = \sum_{i=1}^d C(x^{(i)}, y^{(i)}). \tag{4}
\]

Consequently, the minimization problems \((MP)\) and \((KP)\) turn into \( d \) identical scalar problems for the distributions \( p_{X, Y} \): \( X \) and \( Y \) of the components of \( x \) and \( y \):

\[
\inf \left\{ \mathbb{E}_{X \sim p_X} [C(X, T(X))] \mid T : \mathbb{R} \to \mathbb{R}, T(X) \sim p_Y \right\}, \tag{MP-1-D}
\]

\[
\inf \left\{ \mathbb{E}_{(X, Y) \sim p_{X,Y}} [C(X, Y)] \mid (X, Y) \sim p_{X,Y}, X \sim p_X, Y \sim p_Y \right\}, \tag{KP-1-D}
\]

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 \( p_{x,y}(x, y) := \prod_{i=1}^d p_{X,Y}(x^{(i)}, y^{(i)}) \) 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 a 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

\[
T_{X \to Y}^{\text{mon}}(x) := F_Y^{-1}[F_X(x)], \tag{5}
\]
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 Distribution Function (CDF) of \( X \) and \( F_Y^{-1}(y) := \inf\{t \in \mathbb{R} | F_Y(t) \geq y\} \) is the pseudo-inverse of \( F_Y \). Furthermore, the joint distribution of \((X, T_{X \rightarrow Y}(X))\) defines an optimal transport plan for (KP-1-D).

The mapping \( T_{X \rightarrow Y}^{\text{mon}}(x) \) in Theorem 2 is non-decreasing and is known as the monotone transport map from \( X \) to \( Y \). It is easy to verify that \( T_{X \rightarrow Y}^{\text{mon}}(X) \) has the distribution of \( Y \), in particular \( F_X(X) \sim \text{Uniform}(0,1) \) and if \( U \sim \text{Uniform}(0,1) \) then \( F_Y^{-1}(U) \sim Y \).

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 \( c(x,y) := \|x - y\|_1 \), and use an operation that is pointwise, so we just need to compute the monotone transport map in (5). That is, if \( z \) has i.i.d components with distribution \( p_Z \), we just need to compute the component distribution \( p_Y \) of the result \( y \) of the operation, the CDFs \( F_Z, F_Y \) and obtain

\[
T_{Y \rightarrow Z}^{\text{mon}}(y) := F_Y^{-1}(F_Y(y))
\]

as the component-wise modification of \( y \), i.e. \( \tilde{y}^{(i)} := T_{Y \rightarrow Z}^{\text{mon}}(y^{(i)}) \).

In Figure 3 we show the monotone transport map for the linear interpolation \( y = tz_1 + (1-t)z_2 \) for various values of \( t \). The detailed calculations and examples for various operations are given in Appendix 5.3, for both Uniform and Gaussian priors. The Gaussian case has a particularly simple resulting transport map for additive operations, where it is just a linear transformation through a scalar multiplication, summarized in the third column of Table 1.

3 Experiments

3.1 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 in the literature.

**Priors vs. interpolations in 2-D**  For Figure 1 we sample 1 million pairs of points in two dimensions, from a uniform prior (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 midpoints which are identically distributed to the prior. 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, “circling” around the origin for both priors.

**Priors vs. interpolations in 100-D**  For Figure 2 we sample 1 million pairs of points in \( d = 100 \) dimensions, using either i.i.d. uniform components on \([-1,1]\) or Gaussian \( \mathcal{N}(0,1) \) and compute the
3.2 Qualitative Results

In this section we will present some concrete examples for the differences in generator output dependent on the exact sample operation used to traverse the latent space of a generative model. To this end, the generator output for latent samples produced with linear interpolation, SLERP (spherical linear interpolation) of White (2016) and our proposed matched interpolation will be compared. Please refer to Table 1 for an overview of the operators used in this Section.

Setup We used DCGAN (Radford et al., 2015) generative models trained on LSUN bedrooms (Yu et al., 2015), CelebA (Liu et al., 2015) and LLD (Sage et al., 2017), an icon dataset, to qualitatively evaluate. For LSUN, the model was trained for two different output resolutions, providing $64 \times 64$ pixel and a $128 \times 128$ pixel output images (where the latter is used in figures containing larger sample images). The models for LSUN and the icon dataset where both trained on a uniform latent prior distribution, while for CelebA a Gaussian prior was used. The dimensionality of the latent space is 100 for both LSUN and CelebA, and 512 for the model trained on the icon model. Furthermore we use improved Wasserstein GAN (iWGAN) with gradient penalty (Gulrajani et al., 2017) trained on CIFAR-10 at $32 \times 32$ pixels with a 128-dimensional Gaussian prior to produce the inception scores presented in Section 3.3.
Figure 6: Midpoint sampling for linear, SLERP and uniform-matched interpolation when using the same pairs of sample points on LSUN (64 × 64) with uniform prior.

Figure 7: 4-point interpolation between 4 sampled points (corners) from DCGAN trained on LSUN (128 × 128) using a uniform prior. The same interpolation is shown using linear, SLERP and distribution matched interpolation.

2-point interpolation We begin with the classic example of 2-point interpolation: Figure 4 shows three examples per dataset for an interpolation between 2 points in latent space. Each example is first done via linear interpolation, then SLERP and finally matched interpolation. In Figure 15 in the Appendix we show more densely sampled examples.

It is immediately obvious in Figures 4a and 4b 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 visually obvious which one is superior. Differences between the various interpolation methods for CelebA (Figure 14) are much more subtle to the point that they are virtually indistinguishable when viewed side-by-side. This is not an inconsistency.

Figure 8: 4-point interpolation between 4 sampled points (corners) from DCGAN trained on icon dataset using a uniform prior. The same interpolation is shown using linear, SLERP and distribution matched interpolation.
Figure 9: Vicinity sampling on LSUN dataset (128 × 128) with uniform prior. The sample in the middle is perturbed in random directions producing the surrounding sample points.

(a) Vicinity sampling
(b) Matched vicinity sampling

Figure 10: Vicinity sampling on LLD icon dataset with uniform prior. The sample in the middle is perturbed in random directions producing the surrounding sample points.

though: while distribution mismatch can cause large differences, it can also happen that the model generalizes well enough that it does not matter.

**Midpoint interpolation** In all cases, the point where the interpolation methods diverge the most, is at the midpoint of the interpolation where \( t = 0.5 \). Thus we provide 25 such interpolation midpoints in Figures 5 (LLD icons) and 6 (LSUN) for direct comparison.

**4-point interpolation** An even stronger effect can be observed when we do 4-point interpolation, showcased in Figure 7 (LSUN) and Figure 8 (LLD icons). The higher resolution of the LSUN output

Figure 11: Random walk for LLD, LSUN (64 x 64) and CelebA. The random walks consist of a succession of steps in random directions, calculated for the same sequence of directions using (non-matched) vicinity sampling in the upper rows and our proposed matched vicinity sampling in the lower rows.
An interesting property of our matched vicinity sampling is that we can obtain a sharpness and saturation when compared to both spherical and matched operators. Furthermore we provide two examples for vicinity sampling in Figures 9 and 10. Analogous to the previous observations, the output under a linear operator lacks definition, highlighting the very apparent loss of detail and increasing prevalence of artifacts towards the midpoint in the linear version, compared to SLERP compared and our matched interpolation.

**Random walk** An interesting property of our matched vicinity sampling is that we can obtain a random walk in the latent space by applying it repeatedly: we start at a point \( y_0 \) drawn from the prior, and then obtain point \( y_i \) by sampling a single point in the vicinity of \( y_{i-1} \), using some fixed 'step size' \( \epsilon \).

We show an example of such a walk in Figure 11 using \( \epsilon = 0.5 \). As a result of the repeated application of the vicinity sampling operation, the divergence from the prior distribution in the non-matched case becomes stronger with each step, resulting in completely unrecognizable output images on the LSUN and LLD icon models. Even for the CelebA model where differences were minimal before, they are quite apparent in this experiment. The random walk thus perfectly illustrates the need for respecting the prior distribution when performing any operation in latent space, as the adverse effects can cumulate through the repeated application of operators that do not comply to the prior distribution.

### 3.3 Quantitative Results

We quantitatively confirm the observations of the previous section by using the Inception score [Salimans et al., 2016]. In Table 2 we compare the Inception score of our trained models (i.e. using random samples from the prior) with the score when sampling midpoints from the 2-point and 4-point interpolations described above, reporting mean and standard deviation with 50,000 samples, as well as relative change to the original model scores if they are significant. Compared to the original scores of the trained models, our matched operations are statistically indistinguishable (as expected) while the linear interpolation gives a significantly lower score in all settings (up to 29% lower). As observed for the quality visually, the SLERP heuristic gives similar scores to the matched operations.

### 4 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 mostly ignored by the literature so far, partially due to the 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.
REFERENCES


5 Appendix

5.1 On the curse of dimensionality and geometric outliers

We note that the analysis here can be seen as a more rigorous version of an observation made by White (2016), who experimentally showed 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.

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

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

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

$$\sqrt{d} \left( \frac{1}{d} \|z\|^2 - \mu_{Z^2} \right) \rightarrow N(0, \sigma_{Z^2}^2), \tag{8}$$

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

We note that this property is well known for i.i.d Gaussian entries (see e.g. Ex. 6.14 in MacKay (2003)). For Uniform distribution on the hypercube it is also well known that the mass is concentrated in the corner points (which is consistent with the claim here since the corner points lie on a sphere).

Now consider an operator such as the midpoint of linear interpolation, $y = \frac{1}{2}z_1 + \frac{1}{2}z_2$, with components $Y^{(i)} = \frac{1}{2}Z^{(i)}_1 + \frac{1}{2}Z^{(i)}_2$. Furthermore, let’s assume the component distribution $p_Z$ is symmetric around 0, such that $E[Z] = 0$.

In this case, we can compute:

$$E[(Y^{(i)})^2] = Var\left[\frac{1}{2}Z^{(i)}_1 + \frac{1}{2}Z^{(i)}_2\right] = \frac{1}{2}Var[Z] = \frac{1}{2}\mu_{Z^2} \tag{9}$$

$$Var[(Y^{(i)})^2] = Var\left[\left(\frac{1}{2}Z^{(i)}_1 + \frac{1}{2}Z^{(i)}_2\right)^2\right] = \frac{1}{4}Var[Z^2] = \frac{1}{4}\sigma_{Z^2}^2. \tag{10}$$

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

Therefore, $y$ also mostly lies on a spherical shell, but with a different radius than $z$. 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 $\|y\|^2$, $\|y\|^2$ is a (strong) outlier with respect to the distribution of $\|z\|^2$.

5.2 Proof of Theorem

Proof. We will show it for the Kantorovich problem, the Monge version is similar.
Starting from \([\text{KP}]\), we compute
\[
\inf \left\{ E(x,y) \sim p_{x,y} \left[ c(x,y) \right] \left| (x,y) \sim p_{x,y}, x \sim p_x, y \sim p_y \right. \right\} 
\]
\[
= \inf \left\{ E(x,y) \sim p_{x,y} \left[ \sum_{i=1}^{d} C(x^{(i)}, y^{(i)}) \right] \left| (x,y) \sim p_{x,y}, x \sim p_x, y \sim p_y \right. \right\} 
\]
\[
= \inf \left\{ \sum_{i=1}^{d} E(x,y) \sim p_{x,y} \left[ C(x^{(i)}, y^{(i)}) \right] \left| (x,y) \sim p_{x,y}, x \sim p_x, y \sim p_y \right. \right\} 
\]
\[
\geq \sum_{i=1}^{d} \inf \left\{ E(x,y) \sim p_{x,y} \left[ C(x^{(i)}, y^{(i)}) \right] \left| (x,y) \sim p_{x,y}, x \sim p_x, y \sim p_y \right. \right\} 
\]
\[
= \sum_{i=1}^{d} \inf \left\{ E(X,Y) \sim p_{X,Y} \left[ C(X,Y) \right] \left| (X,Y) \sim p_{X,Y}, X \sim p_X, Y \sim p_Y \right. \right\} 
\]
\[
= d \cdot \inf \left\{ E(X,Y) \sim p_{X,Y} \left[ C(X,Y) \right] \left| (X,Y) \sim p_{X,Y}, X \sim p_X, Y \sim p_Y \right. \right\}, 
\]
where the inequality in (14) is due to each term being minimized separately.

Now let \(\mathcal{P}_d(X,Y)\) be the set of joints \(p_{x,y}\) with \(p_{x,y}(x,y) = \prod_{i=1}^{d} p_{X,Y}(x^{(i)}, y^{(i)})\) 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 \(p_{x,y}\) with marginals \(p_x\) and \(p_y\), where the pairs \((x^{(1)}, y^{(1)}), \ldots, (x^{(d)}, y^{(d)})\) are constrained to be i.i.d. Starting again from (13) we can compute:
\[
\inf \left\{ \sum_{i=1}^{d} E(x,y) \sim p_{x,y} \left[ C(x^{(i)}, y^{(i)}) \right] \left| (x,y) \sim p_{x,y}, x \sim p_x, y \sim p_y \right. \right\} 
\]
\[
\leq \inf \left\{ \sum_{i=1}^{d} E(x,y) \sim p_{x,y} \left[ C(x^{(i)}, y^{(i)}) \right] \left| p_{x,y} \in \mathcal{P}_d(X,Y) \right. \right\} 
\]
\[
= \inf \left\{ \sum_{i=1}^{d} E(x,y) \sim p_{x,y} \left[ C(x^{(i)}, y^{(i)}) \right] \left| p_{x,y} \in \mathcal{P}_d(X,Y) \right. \right\} 
\]
\[
= \inf \left\{ \sum_{i=1}^{d} E(X,Y) \sim p_{X,Y} \left[ C(X,Y) \right] \left| (X,Y) \sim p_{X,Y}, X \sim p_X, Y \sim p_Y \right. \right\} 
\]
\[
= d \cdot \inf \left\{ E(X,Y) \sim p_{X,Y} \left[ C(X,Y) \right] \left| (X,Y) \sim p_{X,Y}, X \sim p_X, Y \sim p_Y \right. \right\}, 
\]
where the inequality in (18) 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:
\[
\inf \left\{ E(x,y) \sim p_{x,y} \left[ c(x,y) \right] \left| (x,y) \sim p_{x,y}, x \sim p_x, y \sim p_y \right. \right\} 
\]
\[
= d \cdot \inf \left\{ E(X,Y) \sim p_{X,Y} \left[ C(X,Y) \right] \left| (X,Y) \sim p_{X,Y}, X \sim p_X, Y \sim p_Y \right. \right\} 
\]
Thus, \([\text{KP}]\) and \([\text{KP-1-D}]\) equal up to a constant, and minimizing one will minimize the other. Therefore the minimization of the former can be done over \(p_{X,Y}\) with \(p_{x,y}(x,y) = \prod_{i=1}^{d} p_{X,Y}(x^{(i)}, y^{(i)})\).

### 5.3 Calculations for Examples

In the next sections, we illustrate how to compute the matched operations for a few examples, in particular for linear interpolation and vicinity sampling, using a uniform or a Gaussian prior.
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 $x$ we will implicitly write $X$ for a scalar random variable that has the distribution of the components of $x$.

When computing the monotone transport map $T^{\text{mon}}_{X \rightarrow Y}$, the following Lemma is helpful.

**Lemma 1** (Theorem 2.5 in Santambrogio (2015)). Suppose a mapping $g(x)$ is non-decreasing and maps a continuous distribution $p_X$ to a distribution $p_Y$, i.e.
\[ g(X) \sim Y, \]  
then $g$ is the monotone transport map $T^{\text{mon}}_{X \rightarrow Y}$.

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

**Example 1: Uniform Linear Interpolation**

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

Now let $y_t = tz_1 + (1 - t)z_2$ denote the linear interpolation between two points $z_1, z_2$, with component distribution $p_Y$. Due to symmetry we can assume that $t > 1/2$, since $p_Y = p_Y^\star$. We then obtain $p_Y$ as the convolution of $p_Z$ and $p_{(1-t)Z}$, i.e. $p_Y = p_Z \ast p_{(1-t)Z}$. First we note that $p_Z = 1/(2t)$ for $-t < z < t$ and $p_{(1-t)Z} = 1/(2(1-t))$ for $-(1-t) < z < 1-t$. We can then compute:
\[ p_Y(y) = (p_Z \ast p_{(1-t)Z})(y) \]
\[ = \frac{1}{2(1-t)(2t)} \begin{cases} 
0 & \text{if } y < -1 \\
\frac{1}{2}(y + 1)(y + 1) & \text{if } -1 < y < -t + (1-t) \\
2(1-t)(y + t) & \text{if } -t + (1-t) < y < t - (1-t) \\
-2t & \text{if } t - (1-t) < y < 1 \\
0 & \text{if } 1 < y
\end{cases} \]

(26)

The CDF $F_Y$ is then obtained by computing
\[ F_Y(y) = \int_{-\infty}^{y} p_Y(y')dy' \]
\[ = \frac{1}{2(1-t)(2t)} \begin{cases} 
0 & \text{if } y < -1 \\
\frac{1}{2}(y + 1)(y + 1) & \text{if } -1 < y < -2t \\
2(1-t)(y + t) & \text{if } -2t < y < 2t - 1 \\
2(1-t)(3t - 1) + \left(-\frac{1}{2}y^2 + y + \frac{1}{2}(2t - 1)^2 - (2t - 1)\right) & \text{if } 2t - 1 < y < 1 \\
2(1-t)(2t) & \text{if } 1 < y
\end{cases} \]

(27)

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 $T^{\text{mon}}_{Y_1 \rightarrow Z}(y) = F_Z^{[-1]}(F_Y(y))$.

**Example 2: Uniform Vicinity Sampling and Random Walk**

Let $z$ again have uniform components on $[-1, 1]$. For vicinity sampling, we want to obtain new points $z'_1, \ldots, z'_k$ which are close to $z$. We thus define
\[ z'_i := z + \epsilon u_i, \]  
(31)

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 $F_Z^{[-1]}(p) = 2(p - \frac{1}{2})$.
where \( u_i \) also has uniform components, such that each coordinate of \( z'_i \) differs at most by \( \epsilon \) from \( z \).

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 \( g_t(Z'_i) := T_{Y_t \rightarrow Z}(tZ'_i) \) will have the distribution of \( Z \), and by Lemma 1 this is then the monotone transport map from \( Z'_i \) to \( Z \).

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

Suppose \( z \) has components \( Z \sim \mathcal{N}(0, \sigma^2) \). In this case, we can compute linear interpolation as before, \( y_t = tz_1 + (1 - t)z_2 \). 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 \( \sigma^2 \). 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 \( z'_i := z + \epsilon e_i \), \( \epsilon \sim \mathcal{N}(0, 1) \), 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 \( z_1, z_2, z_3 \), one wants to apply the difference between \( z_1 \) and \( z_2 \), to \( z_3 \). The transport map is then \( g(y) = \frac{1}{\sqrt{3}} y \).
5.4 ADDITIONAL EXPERIMENTS

Figure 12: 4-point interpolation between 4 sampled points (corners) from DCGAN trained on CelebA with Gaussian prior. The same interpolation is shown using linear, SLERP and distribution matched interpolation.

Figure 13: Midpoint sampling for linear, SLERP and uniform-matched interpolation when using the same pairs of sample points on CelebA with Gaussian prior.

Figure 14: Vicinity sampling on CelebA dataset with Gaussian prior. The sample in the middle is perturbed in random directions producing the surrounding sample points.
Figure 15: 2-point interpolation in detail: Each example shows linear, SLERP and transport matched interpolation from top to bottom respectively, with 16 points taken from the path, produced with DCGAN using a uniform prior distribution.