Isometric Autoencoders

Matan Atzmon[*], **Amos Gropp**[*], **Yaron Lipman**
Weizmann Institute of Science
{matan.atzmon, amos.gropp, yaron.lipman}@weizmann.ac.il

Abstract

High dimensional data is often assumed to be concentrated near a low-dimensional manifold. Autoencoders (AE) is a popular technique to learn representations of such data by pushing it through a neural network with a low dimension bottleneck while minimizing a reconstruction error. Using high capacity AE often leads to a large collection of minimizers, many of which represent a low dimensional manifold that fits the data well but generalizes poorly.

Two sources of bad generalization are: extrinsic, where the learned manifold possesses extraneous parts that are far from the data; and intrinsic, where the encoder and decoder introduce arbitrary distortion in the low dimensional parameterization. An approach taken to alleviate these issues is to add a regularizer that favors a particular solution; common regularizers promote sparsity, small derivatives, or robustness to noise.

In this paper, we advocate an isometry (i.e., distance preserving) regularizer. Specifically, our regularizer encourages: (i) the decoder to be an isometry; and (ii) the encoder to be a pseudo-isometry, where pseudo-isometry is an extension of an isometry with an orthogonal projection operator. In a nutshell, (i) preserves all geometric properties of the data such as volume, length, angle, and probability density. It fixes the intrinsic degree of freedom since any two isometric decoders to the same manifold will differ by a rigid motion. (ii) Addresses the extrinsic degree of freedom by minimizing derivatives in orthogonal directions to the manifold and hence disfavoring complicated manifold solutions. Experimenting with the isometry regularizer on dimensionality reduction tasks produces useful low-dimensional data representations, while incorporating it in AE models leads to an improved generalization.

1 Introduction

A common assumption is that the high dimensional data $\mathcal{X} \subset \mathbb{R}^D$ is sampled from some distribution $P(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^D$, concentrated on or near some lower d-dimensional submanifold $\mathcal{M} \subset \mathbb{R}^D$, $d < D$. The task of estimating $P(\boldsymbol{x})$ can therefore be decomposed into: (i) approximate the manifold $\mathcal{M}$; and (ii) approximate the probability density P restricted to, or concentrated near $\mathcal{M}$.

In this paper we focus on task (i), mostly known as *manifold learning*. A common approach to approximate the d-dimensional manifold $\mathcal{M}$, e.g., in [Tenenbaum et al., 2000, Roweis and Saul, 2000, Belkin and Niyogi, 2002, Maaten and Hinton, 2008, McQueen et al., 2016, McInnes et al., 2018], is to embed $\mathcal{X}$ in $\mathbb{R}^d$. This is often done by first constructing a graph $\mathcal{G}$ by connecting nearby samples in $\mathcal{X}$ and optimizing for the locations of the samples in $\mathbb{R}^d$ where the target is minimizing distortions of the edge lengths in $\mathcal{G}$.

Autoencoders (AE) can also be seen as a method to learn low dimensional manifold representation of high dimensional data $\mathcal{X}$. AE are trying to reconstruct $\mathcal{X}$ as the image of its low dimensional embedding. When restricting AE to linear encoders and decoders it learns linear subspaces; with mean

[*]Equal Contribution

Preprint. Under review.

squared reconstruction loss they coincide with the subspaces of principle component analysis (PCA). Using higher capacity neural networks as the encoder and decoder in the AE allows complex manifolds to be approximated. To avoid overfitting, different regularizers are added to the AE loss. Popular regularizers include sparsity promoting [Ranzato et al., 2007, Ranzato et al., 2008, Glorot et al., 2011], contractive or penalizing large derivatives [Rifai et al., 2011a, Rifai et al., 2011b], and denoising [Vincent et al., 2010, Poole et al., 2014].

Similarly to manifold learning methods, a natural AE regularization is promoting distance preservation of the encoder [Pai et al., 2019, Zhan et al., 2018, Peterfreund et al., 2020]. There are two potential drawbacks to considering only the encoder: First, it can map different parts of $\mathcal{M}$ to the same low dimensional location (non-injectivity), second, it cannot enforce isometry[2] in areas of $\mathcal{M}$ not covered by $\mathcal{X}$. Restricting the decoder and encoder to isometries is beneficial for several reasons. First, Nash-Kuiper Embedding Theorem [Nash, 1956] asserts that non-expansive maps can be approximated arbitrary well with isometries if $D \geq d+1$ and hence promoting an isometry does not limit the expressive power of the decoder. Second, the low dimensional representation of the data computed with an isometric encoder preserves the geometric structure of the data. In particular volume, length, angles and probability densities are preserved between the low dimensional representation $\mathbb{R}^d$, and the learned manifold $\mathcal{N}$. Lastly, given a manifold $\mathcal{N}$ there is a huge space of possible decoders parameterizing it. Restricting the decoder to an isometry fixes this degree of freedom up to a global rigid transformation of the low dimensional space.

Unfortunately, promoting isometry of the decoder [Kato et al., 2019] and/or the encoder (when restricted to the learned manifold) is not sufficient to provide a good approximation to $\mathcal{M}$. Indeed, there exists many isometries $\mathbb{R}^d \to \mathbb{R}^D$ that reconstruct $\mathcal{X}$ and do not provide a good approximation to $\mathcal{M}$. As an illustrative example, Figure 1 depicts two isometries: the isometry on the right hand side interpolates the data points but provides a complex, undesired interpretation to the data, while the isometry on the left provides an intuitive interpretation of the data.

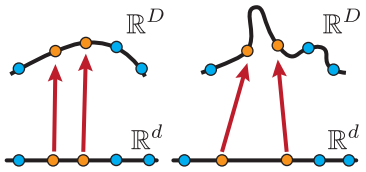

Figure 1: Two isometries reconstructing the data $\mathcal{X}$.

In this paper we advocate a novel regularization promoting AE isometry, I-AE in short. Our key idea is to promote both isometry of the decoder and *pseudo-isometry* of the encoder. Pseudo-isometric encoder is defined to be an *extension by projection* of a restricted isometry. The I-AE regularization pushes the differential of the encoder, $\boldsymbol{B} \in \mathbb{R}^{d\times D}$ to be the pseudo-inverse of the differential of the decoder $\boldsymbol{A} \in \mathbb{R}^{D\times d}$, namely, $\boldsymbol{B} = \boldsymbol{A}^+$. This means that *locally* our decoder and encoder behave like PCA, where the encoder can be seen as a composition of a projection on the linear subspace spanned by the decoder, followed by an orthogonal transformation (isometry) to the low dimensional space. In other words, the encoder is ideally an isometry in the learned manifold's tangent directions, and contractive in the learned manifold's normal directions. Therefore, I-AE would tend to avoid, for example, the large orthogonal derivatives as could happen between the orange data points in the right hand side solution in Figure 1, and favor *simple* isometric decoders, such as the one on the left hand side. To promote orthogonal $\boldsymbol{A}$, $\boldsymbol{B}$ (and consequently $\boldsymbol{B} = \boldsymbol{A}^+$) we derive a simple symmetric characterization and a corresponding tractable loss.

I-AE provides a geometric construction that aligns well with the general motivation behind regularized AE, namely, learning a manifold $\mathcal{N}$ capturing the variations in tangent directions of $\mathcal{M}$ while ignoring orthogonal variations which often represent noise [Alain and Bengio, 2014]. Experiments confirm that optimizing the I-AE loss results in a close-to-isometric encoder/decoder explaining the data. We further demonstrate the efficacy of I-AE for dimensionality reduction of different standard datatsets, showing its benefits over manifold learning and other AE baselines.

2 Related works

Manifold learning. Manifold learning generalize classic dimensionality reduction methods such as PCA [Pearson, 1901] and MDS [Kruskal, 1964, Sammon, 1969], by aiming to preserve the local geometry of the data. [Tenenbaum et al., 2000] use the nn-graph to approximate the geodesic distances over the manifold, followed by MDS to preserve it in the lower dimension. [Roweis and Saul, 2000,

[2]Isometry is a map that preserves distances between all pairs of points.

Belkin and Niyogi, 2002, Donoho and Grimes, 2003] use spectral methods to minimize different distortion energy functions over the graph matrix. [Coifman et al., 2005, Coifman and Lafon, 2006] approximate the heat diffusion over the manifold by a random walk over the nn-graph, to gain a robust distance measure on the manifold. Stochastic neighboring embedding algorithms [Hinton and Roweis, 2003, Maaten and Hinton, 2008] captures the local geometry of the data as a mixture of Gaussians around each data points, and try to find a low dimension mixture model by minimizing the KL-divergence. In a relatively recent work, [McInnes et al., 2018] use iterative spectral and embedding optimization using fuzzy sets. Several works tried to adapt classic manifold learning ideas to neural networks and autoencoders. [Pai et al., 2019] suggest to embed high dimensional points into a low dimension with a neural network by constructing a metric between pairs of data points and minimizing the metric distortion energy. [Kato et al., 2019] suggest to learn an isometric decoder by using noisy latent variables. They prove under certain conditions that it encourages isometric decoder. [Peterfreund et al., 2020] suggest autoencoders that promote the isometry of the encoder over the data by approximating its differential gram matrix using sample covariance matrix. [Zhan et al., 2018] encourage distance preserving autoencoders by minimizing metric distortion energy in common feature space.

Generative models. There is an extensive literature on extending autoencoders to a generative model (task (ii) in section 1). That is, learning a probability distribution in addition to approximating the data manifold $\mathcal{M}$. Variational autoencoder (VAE) [Kingma and Welling, 2014] and its variants [Makhzani et al., 2015, Burda et al., 2016, Sønderby et al., 2016, Higgins et al., 2017, Tolstikhin et al., 2018, Park et al., 2019, Zhao et al., 2019] are examples to such methods. In essence, these methods augment the AE structure with a learned probabilistic model in the low dimensional (latent) space $\mathbb{R}^d$ that is used to approximate the probability P that generated the observed data $\mathcal{X}$. More relevant to our work, are recent works suggesting regularizers for deterministic autoencoders that together with ex-post density estimation in latent space forms a generative model. [Ghosh et al., 2020] suggested to reduce the decoder degrees of freedom, either by regularizing the norm of the decoder weights or the norm of the decoder differential. Other regularizers of the differential of the decoder, aiming towards a deterministic variant of VAE, were recently suggested in [Kumar and Poole, 2020, Kumar et al., 2020]. In contrast to our method, these methods do not regularize the encoder.

3 Isometric autoencoders

We consider high dimensional data points $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^n \subset \mathbb{R}^D$ sampled from some probability distribution $P(\boldsymbol{x})$ in $\mathbb{R}^D$ concentrated on or near some d dimensional submanifold $\mathcal{M} \subset \mathbb{R}^D$, where $d < D$.

Our goal is to compute *isometric autoencoder* (I-AE) defined as follows. Let $g : \mathbb{R}^D \to \mathbb{R}^d$ denote the encoder, and $f : \mathbb{R}^d \to \mathbb{R}^D$ the decoder; $\mathcal{N}$ is the learned manifold, i.e., the image of the decoder, $\mathcal{N} = f(\mathbb{R}^d)$. I-AE is defined by the following requirements:

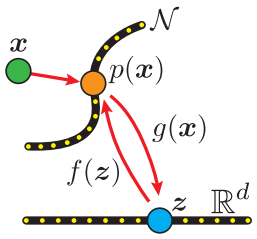

Figure 2: I-AE.

(i) The data $\mathcal{X}$ is close to $\mathcal{N}$.

(ii) g is the inverse of f when restricted to $\mathcal{N}$.

(iii) f is an isometry.

(iv) g is a pseudo-isometry (defined shortly).

Figure 2 is an illustration of I-AE. Let θ denote the parameters of f, and ϕ the parameters of g. We enforce the requirements (i)-(iv) by prescribing a loss function $L(\theta, \phi)$ and optimize it using standard stochastic gradient descent (SGD). We next break down the loss L to its different components.

Condition (i) is promoted with the standard reconstruction loss in AE:

$$L_{\text{rec}}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} \left\| f(g(\boldsymbol{x}_i)) - \boldsymbol{x}_i \right\|^2, \tag{1}$$

where $\|\cdot\|$ is the 2-norm. Condition (ii) is promoted using the loss:

$$L_{\text{inv}}(\theta, \phi) = \mathbb{E}_{\boldsymbol{z}} \left\| g(f(\boldsymbol{z})) - \boldsymbol{z} \right\|^2, \tag{2}$$

where $\boldsymbol{z} \sim P_{\text{inv}}(\mathbb{R}^d)$, and $P_{\text{inv}}(\boldsymbol{z})$ is some probability measure on $\mathbb{R}^d$.

Before handeling conditions (iii),(iv) let us first define the notions of isometry and pseudo-isometry. A differentiable mapping f between the euclidean spaces $\mathbb{R}^d$ and $\mathbb{R}^D$ is a (local) isometry if it has an orthogonal differential matrix $df(\boldsymbol{z}) \in \mathbb{R}^{D\times d}$ everywhere,

$$df(\boldsymbol{z})^T df(\boldsymbol{z}) = \boldsymbol{I}_d, \tag{3}$$

where $\boldsymbol{I}_d \in \mathbb{R}^{d\times d}$ is the identity matrix, and $df(\boldsymbol{z})_{ij} = \frac{\partial f^i}{\partial z_j}(\boldsymbol{z})$. The benefit in restricting the decoder f to an isometry is discussed in Section 1. Here, we elaborate on one aspect of these benefits: for a fixed manifold $\mathcal{N}$ there is a huge space of possible decoders such that $\mathcal{N} = f(\mathbb{R}^d)$. For isometric f, this space is reduced considerably: Indeed, consider two isometries parameterizing $\mathcal{N}$, i.e., $f_1, f_2 : \mathbb{R}^d \to \mathcal{N}$. Then, since composition of isometries is an isometry we have that $f_2^{-1} \circ f_1 : \mathbb{R}^d \to \mathbb{R}^d$ is a dimension-preserving isometry and hence a rigid motion. That is, all decoders of the same manifold are the same up to a rigid motion.

For the encoder the situation is different. Since $D > d$ the encoder g cannot be an isometry in the standard sense. Therefore we define the notion of *pseudo-isometry*. For that end we define the projection operator $\boldsymbol{p}$ on a submanifold $\mathcal{N} \subset \mathbb{R}^D$ as

$$\boldsymbol{p}(\boldsymbol{x}) = \operatorname*{arg\,min}_{\boldsymbol{x}' \in \mathcal{N}} \|\boldsymbol{x} - \boldsymbol{x}'\| .$$

Definition 1. *We say the g is a pseudo-isometry if there exists a d-dimensional submanifold $\mathcal{N} \subset \mathbb{R}^D$ so that $g = g \circ \boldsymbol{p}$ and $g|_{\mathcal{N}} : \mathcal{N} \to \mathbb{R}^d$ is an isometry.*

Intuitively, g extends the standard notion of isometry by projecting every point on a submanifold $\mathcal{N}$ and then applying an isometry between the d-dimensional manifolds $\mathcal{N}$ and $\mathbb{R}^d$. See Figure 2 for an illustration.

First-order characterization. To encourage f, g to satisfy the isometry and the pseudo-isometry properties (resp.) we will first provide a first-order necessary and sufficient characterization using their differentials.

Theorem 1. *Let f be a decoder and g an encoder satisfying conditions (ii),(iii),(iv). Then their differentials $\boldsymbol{A} = df(\boldsymbol{z}) \in \mathbb{R}^{D\times d}$, $\boldsymbol{B} = dg(f(\boldsymbol{z})) \in \mathbb{R}^{d\times D}$ satisfy*

$$\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}_d \tag{4}$$

$$\boldsymbol{B}\boldsymbol{B}^T = \boldsymbol{I}_d \tag{5}$$

$$\boldsymbol{B} = \boldsymbol{A}^+ \tag{6}$$

The theorem asserts that the differentials of the encoder and decoder are orthogonal (rectangular) matrices, and that the encoder is the pseudo-inverse of the differential of the decoder. Before proving this theorem, let us first use it to construct the relevant loss for promoting the isometry of f and pseudo-isometry of g. We need to promote conditions (4), (5), (6). Since we want to avoid computing the full differentials $\boldsymbol{A} = df(\boldsymbol{z})$, $\boldsymbol{B} = dg(f(\boldsymbol{z}))$, we will replace (4) and (5) with stochastic estimations based on the following lemma: denote the unit $d-1$-sphere by $\mathcal{S}^{d-1} = \left\{\boldsymbol{z} \in \mathbb{R}^d \middle| \|\boldsymbol{z}\| = 1\right\}$.

Lemma 1. *Let $\boldsymbol{A} \in \mathbb{R}^{D\times d}$, where $d \le D$. If $\|\boldsymbol{A}\boldsymbol{u}\| = 1$ for all $\boldsymbol{u} \in \mathcal{S}^{d-1}$, then $\boldsymbol{A}$ is column-orthogonal, that is $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}_d$.*

Therefore, the isometry promoting loss, encouraging (4), is defined by

$$L_{\text{iso}}(\theta) = \mathbb{E}_{\boldsymbol{z},\boldsymbol{u}}\Big(\|df(\boldsymbol{z})\boldsymbol{u}\| - 1\Big)^2, \tag{7}$$

where $\boldsymbol{z} \sim P_{\text{iso}}(\mathbb{R}^d)$, and $P_{\text{iso}}(\mathbb{R}^d)$ is a probability measure on $\mathbb{R}^d$; $\boldsymbol{u} \sim P(\mathcal{S}^{d-1})$, and $P(\mathcal{S}^{d-1})$ is the standard rotation invariant probability measure on the $d-1$-sphere $\mathcal{S}^{d-1}$. The pseudo-isometry promoting loss, encouraging (5) would be

$$L_{\text{piso}}(\phi) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{u}}\Big(\left\|\boldsymbol{u}^T dg(\boldsymbol{x})\right\| - 1\Big)^2, \tag{8}$$

where $\boldsymbol{x} \sim P(\mathcal{M})$ and $\boldsymbol{u} \sim P(\mathcal{S}^{d-1})$. As-usual, the expectation with respect to $P(\mathcal{M})$ is computed empirically using the data samples $\mathcal{X}$.

Lastly, (6) might seem challenging at first look, however the orthogonality of $\boldsymbol{A}, \boldsymbol{B}$ can be used to show it is automatically satisfied due to the requirement (ii) above. This is justified in the following lemma:

Lemma 2. *Let $\boldsymbol{A} \in \mathbb{R}^{D\times d}$, and $\boldsymbol{B} \in \mathbb{R}^{d\times D}$. If $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}_d = \boldsymbol{B}\boldsymbol{B}^T$ and $\boldsymbol{B}\boldsymbol{A} = \boldsymbol{I}_d$ then $\boldsymbol{B} = \boldsymbol{A}^+ = \boldsymbol{A}^T$.*

Indeed, differentiating both sides of the equation $\boldsymbol{z} = g(f(\boldsymbol{z}))$, and using the chain rule, we get $\boldsymbol{I}_d = dg(f(\boldsymbol{z}))df(\boldsymbol{z}) = \boldsymbol{B}\boldsymbol{A}$. Given (4), (5), Lemma 2 now implies that $\boldsymbol{B} = \boldsymbol{A}^+$, as desired.

Summing all up, we define our loss for I-AE by

$$L(\theta, \phi) = \lambda_{\text{rec}} L_{\text{rec}}(\theta, \phi) + \lambda_{\text{inv}} L_{\text{inv}}(\theta, \phi) + \lambda_{\text{iso}} L_{\text{iso}}(\theta) + \lambda_{\text{piso}} L_{\text{piso}}(\phi), \tag{9}$$

where λ are weight parameters.

Details and proofs. Let us prove Theorem 1 characterizing the relation of the differentials of isometries and pseudo-isometries, $\boldsymbol{A} = df(\boldsymbol{z}) \in \mathbb{R}^{D\times d}$, $\boldsymbol{B} = dg(f(\boldsymbol{z})) \in \mathbb{R}^{d\times D}$. First, by definition of isometry (equation 3), $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}_d$. We denote by $T_{\boldsymbol{x}}\mathcal{N}$ the d-dimensional tangent space to $\mathcal{N}$ at $\boldsymbol{x} \in \mathcal{N}$; accordingly, $T_{\boldsymbol{x}}\mathcal{N}^\perp$ denotes the normal tangent space.

Lemma 3. *The differential $d\boldsymbol{p}(\boldsymbol{x}) \in \mathbb{R}^{D\times D}$ at $\boldsymbol{x} \in \mathcal{N}$ of the projection operator $\boldsymbol{p} : \mathbb{R}^D \to \mathcal{N}$ is*

$$d\boldsymbol{p}(\boldsymbol{x})\boldsymbol{u} = \begin{cases} \boldsymbol{u} & \boldsymbol{u} \in T_{\boldsymbol{x}}\mathcal{N} \\ 0 & \boldsymbol{u} \in T_{\boldsymbol{x}}\mathcal{N}^\perp \end{cases} \tag{10}$$

That is, $d\boldsymbol{p}(\boldsymbol{x})$ is the orthogonal projection on the tangent space of $\mathcal{N}$ at $\boldsymbol{x}$.

Proof. First, consider the squared distance function to $\mathcal{N}$ defined by $\eta(\boldsymbol{x}) = \frac{1}{2}\min_{\boldsymbol{x}'\in\mathcal{N}} \|\boldsymbol{x} - \boldsymbol{x}'\|^2$. The envelope theorem implies that $\nabla\eta(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{p}(\boldsymbol{x})$. Differentiating both sides and rearranging we get $d\boldsymbol{p}(\boldsymbol{x}) = \boldsymbol{I}_D - \nabla^2\eta(\boldsymbol{x})$. As proved in [Ambrosio and Soner, 1994] (Theorem 3.1), $\nabla^2\eta(\boldsymbol{x})$ is the orthogonal projection on $T_{\boldsymbol{x}}\mathcal{N}^\perp$. □

Thus $d\boldsymbol{p}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{A}^T$, since $\boldsymbol{A}\boldsymbol{A}^T$ is the linear projection on $\mathcal{T}_{\boldsymbol{x}}\mathcal{N}$. This leads us to the characterization of $\boldsymbol{B}$. Indeed, let $\boldsymbol{x} = f(\boldsymbol{z}) \in \mathcal{N}$. Then $\boldsymbol{p}(\boldsymbol{x}) = \boldsymbol{x}$, and $g(\boldsymbol{x}) = g(\boldsymbol{p}(\boldsymbol{x}))$ as g is a pseudo-isometry. Using the chain rule implies $\boldsymbol{B} = dg(\boldsymbol{p}(\boldsymbol{x}))d\boldsymbol{p}(\boldsymbol{x}) = \boldsymbol{B}\boldsymbol{A}\boldsymbol{A}^T$. Then, $\boldsymbol{B}\boldsymbol{B}^T = \boldsymbol{B}\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{B}^T = \boldsymbol{B}\boldsymbol{A}(\boldsymbol{B}\boldsymbol{A})^T = \boldsymbol{I}_d$ where the last equality follows from the fact that $\boldsymbol{B}\boldsymbol{A}$ is an orthogonal matrix as the restriction of g to $\mathcal{N}$ is an isometry. We saw above that $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}_d$ and $\boldsymbol{B}\boldsymbol{A} = \boldsymbol{I}_d$, therefore lemma 2 implies that $\boldsymbol{B} = \boldsymbol{A}^+$.

Proof of Lemma 1. Writing the SVD of $\boldsymbol{A} = \boldsymbol{U}\Sigma\boldsymbol{V}^T$, where $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_d)$ are the singular values of $\boldsymbol{A}$, we get that $\sum_{i=1}^d \sigma_i^2\boldsymbol{v}_i^2 = 1$ for all $\boldsymbol{v} \in \mathcal{S}^{d-1}$. Plugging $\boldsymbol{v} = \boldsymbol{e}_j$, $j \in [d]$ (the standard basis) we get that all $\sigma_i = 1$ for $i \in [d]$ and $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{V}^T$ is orthogonal as claimed. □

Proof of Lemma 2. Let $\boldsymbol{U} = [\boldsymbol{A}, \boldsymbol{V}]$, $\boldsymbol{V} \in \mathbb{R}^{D\times(D-d)}$, be a completion of $\boldsymbol{A}$ to an orthogonal matrix in $\mathbb{R}^{D\times D}$. Now, $\boldsymbol{I}_d = \boldsymbol{B}\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{B}^T = \boldsymbol{I}_d + \boldsymbol{B}\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{B}^T$, and since $\boldsymbol{B}\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{B}^T \succeq 0$ this means that $\boldsymbol{B}\boldsymbol{V} = 0$, that is $\boldsymbol{B}$ takes to null the orthogonal space to the column space of $\boldsymbol{A}$. A direct computation shows that $\boldsymbol{B}\boldsymbol{U} = \boldsymbol{A}^T\boldsymbol{U}$ which in turn implies $\boldsymbol{B} = \boldsymbol{A}^T = \boldsymbol{A}^+$. □

Implementation. Implementing the losses in equation 2, equation 7, and equation 8 requires making a choice for the probability densities and approximating the expectations. We take $P = P_{\text{inv}} = P_{\text{iso}}$ to be either uniform or gaussian fit to the latent codes $g(\mathcal{X})$; and $P(\mathcal{M})$ is approximated as the uniform distribution on $\mathcal{X}$, as mentioned above. The expectations are estimated using Monte-Carlo sampling. That is, at each iteration we draw samples $\hat{\boldsymbol{x}} \in \mathcal{X}$, $\hat{\boldsymbol{z}} \sim P(\mathbb{R}^d)$, $\boldsymbol{u} \sim P(\mathcal{S}^{d-1})$ and use the approximations

$$\begin{aligned} L_{\text{inv}}(\theta, \phi) &\approx \|g(f(\hat{\boldsymbol{z}})) - \hat{\boldsymbol{z}}\|^2 \\ L_{\text{iso}}(\theta) &\approx \left(\|df(\hat{\boldsymbol{z}})\hat{\boldsymbol{u}}\| - 1\right)^2 \\ L_{\text{piso}}(\phi) &\approx \left(\left\|\hat{\boldsymbol{u}}^T dg(\hat{\boldsymbol{x}})\right\| - 1\right)^2 \end{aligned}$$

The right differential multiplication $df(\hat{\boldsymbol{z}})\hat{\boldsymbol{u}}$ and left differential multiplication $\hat{\boldsymbol{u}}^T dg(\hat{\boldsymbol{x}})$ are computed using forward and backward mode automatic differentiation (resp.). Their derivatives with respect to the networks' parameters θ, ϕ are computed by another backward mode automatic differentiation.

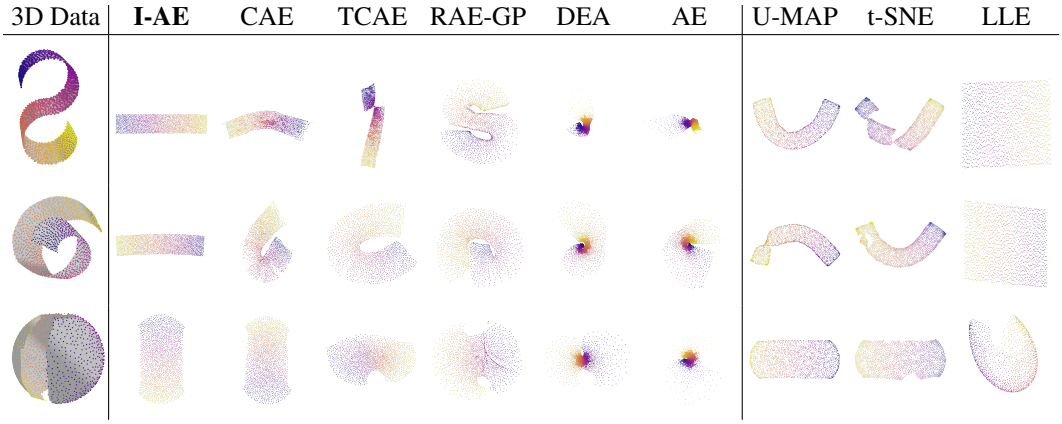

Figure 4: Evaluation of $3D \rightarrow 2D$ embeddings.

4 Experiments

4.1 Evaluation

We start by evaluating the effectiveness of our suggested I-AE regularizer, addressing the following questions: (i) does our suggested loss $L(\theta, \phi)$ in equation 9 drives I-AE training to converge to an isometry? (ii) What is the effect of the L_{piso} term? In particular, does it encourage better manifold approximations as conjectured? To that end, we examined the I-AE training on data points $\mathcal{X}$ sampled uniformly from 3D surfaces with known global parameterizations. Figure 4 shows qualitative comparison of the learned embeddings for various AE regularization techniques: Vanilla autoencoder (AE); Contractive autoencoder (CAE) [Rifai et al., 2011b]; Contractive autoencoder with decoder weights tied to the encoder weights (TCAE) [Rifai et al., 2011a]; Gradient penalty on the decoder (RAE-GP) [Ghosh et al., 2020]; Denoising autoencoder with gaussian noise (DAE) [Vincent et al., 2010]. For fairness in evaluation, all methods were trained using the same training hyper-parameters. See supplementary for the complete experiment details including mathematical formulation of the different AE regularizers. In addition, we compared versus popular classic manifold learning techniques: U-MAP [McInnes et al., 2018], t-SNE [Maaten and Hinton, 2008] and LLE. [Roweis and Saul, 2000]. The results demonstrate that I-AE is able to learn an isometric embedding, showing some of the advantages in our method: sampling density and distances between input points is preserved in the learned low dimensional space.

In addition, for the AE methods, we quantitatively evaluate how close is the learnt decoder to an isometry. For this purpose, we triangulate a grid of planar points $\{\boldsymbol{z}_i\} \subset \mathbb{R}^2$. We denote by $\{e_{ij}\}$ the triangles edges incident to grid points $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$. Then, we measured the edge lengths ratio, $l_{ij} = \|g(\boldsymbol{z}_i) - g(\boldsymbol{z}_i)\| / \|e_{ij}\|$ expected to be ≈ 1 for all edges e_{ij} in an isometry. In Table 1 we log the standard deviation (Std) of $\{l_{ij}\}$ for I-AE compared to other regularized AEs. For a fair comparison, we scaled $\boldsymbol{z}_i$ so the mean of l_{ij} is 1 and measured standard deviation. As can be seen in the table, the distribution of $\{l_{ij}\}$ for I-AE is significantly more concentrated than the different AE baselines.

	I-AE	CAE	TCAE	RAE-GP	DAE	AE
S Shape	**0.03**	0.36	0.26	1.22	2.53	1.85
Swiss Roll	**0.02**	1.00	0.38	1.75	1.80	1.63
Open Sphere	**0.07**	0.21	0.21	0.50	1.09	1.29

Table 1: Std of $\{l_{ij}\}$.

Finally, to support the claim that the L_{piso} term has a significant role in converging to simpler solutions (see figure 1), we ran AE training with and without the L_{iso} term. Thus, we still expect the decoder to approximate an isometry that passes through the input points, nevertheless, possessing more degrees of freedom that might yield to complex solutions. Indeed, inset figure 3 shows in gray the learnt decoder surface without L_{piso} (left), containing extra (unnatural) surface parts compared to the learnt surface with L_{iso} (right).

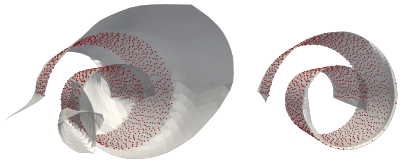

Figure 3: Decoder surfaces without L_{piso} (left) and with (right).

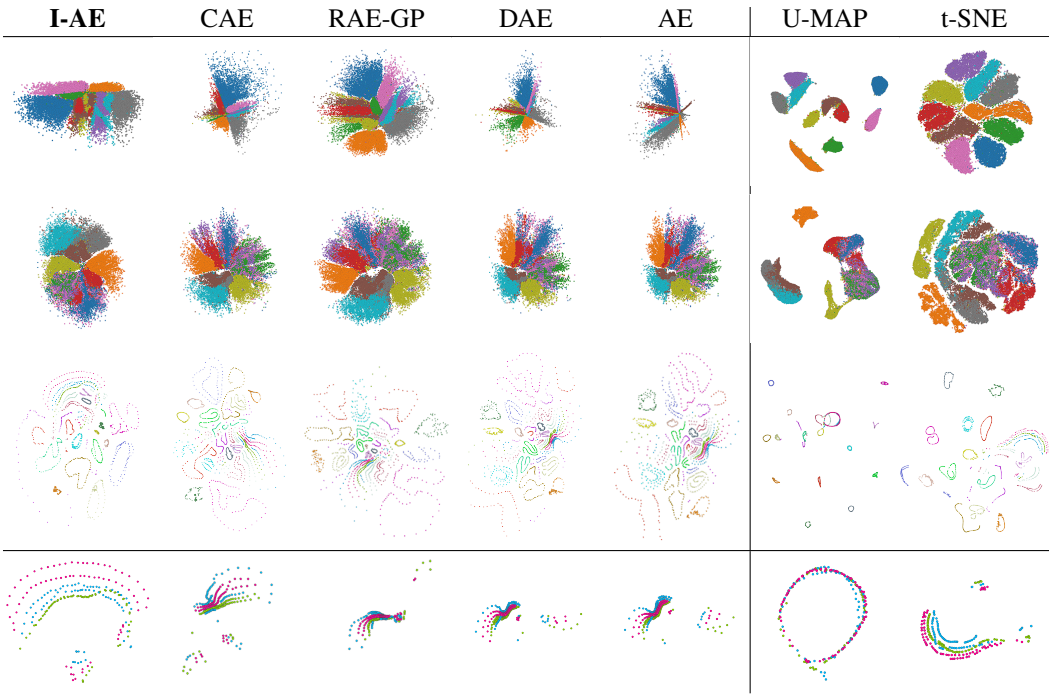

Figure 5: Results of data visualization experiment. Top 3 rows show MNIST; FMNIST; and COIL20. Different colors indicate different classes. Bottom row: zoom-ins on 3 classes from the COIL20 visualization.

4.2 Data visualization

In this experiment we evaluate our method in the task of high dimension data visualization, i.e., reducing high dimensional data into two dimensions to be visually interpreted by the human eye. Usually the data is not assumed to lie on a manifold with such a low dimension, and it is therefore impossible to preserve all of its geometric properties. A common artifact when squeezing higher dimensional data into the plane is crowding [Maaten and Hinton, 2008], that is planar embedded points are crowded around the origin.

We evaluate our method on three standard datasets of images: MNIST [LeCun, 1998] (60k handwritten digits), Fashion-MNIST (60k Zalando's article images) [Xiao et al., 2017] and COIL20 [Nene et al., 1996] (20 different images of object rotated with 72 even rotations). For baselines we take: Vanilla AE; CAE; GP-RAE; DAE; U-MAP; and t-SNE. We use the same architecture for all methods on each dataset: MNIST: Both encoder and decoder are Fully-connected (MLP) networks; FMNIST and COIL20: Both encoder and decoder are Fully Convolutional Neural Network (CNN). Full implementation details and hyper-parameters values can be found in the supplementary.

The results are presented in figure 5; where each embedded point $\boldsymbol{z}$ is colored by its ground-truth class/label. We make several observation. First, in all the datasets our method is more resilient to crowding compared to the baseline AEs, and provide a more even spread. U-MAP and t-SNE produce better separated clusters. However, this separation can come at a cost: See the COIL20 result (third row) and blow-ups of three of the classes (bottom row). In this dataset we expect evenly spaced points that correspond to the even rotations of the objects in the images. Note (in the blow-ups) that U-MAP maps the three classes on top of each other (non-injectivity of the "encoder"), t-SNE is somewhat better but does not preserve well the distance between pairs of data points (we expect them to be more or less equidistant in this dataset). In I-AE the rings are better separated and points are more equidistant; the baseline AEs tend to densify the points near the origin. Lastly, considering the inter and intra-class variations for the MNIST and FMNIST datasets, we are not sure that isometric embeddings are expected to produce strongly separated clusters as in U-MAP and t-SNE (e.g., think about similar digits of different classes and dissimilar digits of the same class).

4.3 Generalization in high dimensional space

Next, we evaluate how well our suggested isometric prior induces manifolds that generalizes well to unseen data. We experimented with three different images datasets: MNIST [LeCun, 1998]; CIFAR10 [Krizhevsky et al., 2009]; and CelebA [Liu et al., 2015]. We quantitatively estimate methods performance by measuring the L_2 distance and the *Fréchet Inception Distance* (FID) [Heusel et al., 2017] on a held out test set. For each dataset, we used the official train-test splits.

For comparison versus baselines we have selected among relevant existing AE based methods the following: Vanilla AE (AE); autoencoder trained with weight decay (AEW); Contractive autoencoder (CAE); autoencoder with spectral weights normalization (RAE-SN); and autoencoder with L_2 regularization on decoder weights (RAE-SN). RAE-L_2 and RAE-SN were recently successfully applied to this data in [Ghosh et al., 2020], demonstrating state-of-the-art performance on this task. In addition, we compare versus the Wasserstein Auto-Encoder (WAE) [Tolstikhin et al., 2018], chosen as state-of-the-art among generative autoencoders.

For evaluation fairness, all methods were trained using the same training hyper-parameters: network architecture, optimizer settings, batch size, number of epochs for training and learning rate scheduling. See supplementary for specific hyper-parameters values. In addition, we generated a validation set out of the training set using 10k samples for the MNIST and CIFAR-10 experiment, whereas for the CelebA experiment we used the official validation set. For each training epoch, we evaluated the reconstruction L_2 loss on the validation set and chose the final network weights to be the one that achieves the minimum reconstruction. We experimented with two variants of I-AE regularizers: L_{piso} and $L_{\text{piso}} + L_{\text{iso}}$. Table 2 logs the results. Note that I-AE produced competitive results with the current SOTA on this task.

		Methods							
Dataset	Distance	L_{piso}	$L_{\text{piso}} + L_{\text{iso}}$	AE	AEW	CAE	RAE-SN	RAE-L_2	WAE
MNIST	L_2	**0.96**	**0.99**	1.14	1.0	1.15	1.35	1.14	1.64
	FID	6.09	7.94	**4.95**	**5.59**	6.46	10.72	11.41	6.99
CIFAR-10	L_2	**20.19**	21.05	**20.16**	20.33	20.23	21.02	20.2	21.08
	FID	70.14	**56.04**	74.79	**68.71**	71.71	70.79	71.05	74.2
CelebA	L_2	20.38	**19.93**	20.51	**19.74**	20.46	20.78	20.58	20.88
	FID	**34.68**	40.73	40.53	40.00	39.52	40.45	**38.86**	38.98

Table 2: Manifold approximation quality on test images. We log the L_2 and FID distances (lower is better) from reconstructed images to the input images. The L_2 numbers are reported $*10^3$. The top performance scores are highlighted as: **First**, **Second**.

5 Conclusions

We have introduced I-AE, a regularizer for autoencoders that promotes isometry of the decoder and pseudo-isometry of the encoder. Our goal was two-fold: (i) producing a favorable low dimensional manifold approximation to high dimensional data, isometrically parameterized for preserving, as much as possible, its geometric properties; and (ii) avoiding complex isometries based on the notion of psuedo-isometry. Our regularizers are simple to implement and can be easily incorporated into existing autoencoders architectures. We have tested I-AE on common manifold learning tasks, demonstrating the usefulness of isometric autoencoders.

An interesting future work venue is to consider task (ii) from section 1, namely incorporating a probabilistic model and examine the potential benefits of the isometry prior for generative models. One motivation is the fact that isometries push probability distributions by a simple change of coordinates, $P(\boldsymbol{z}) = P(f(\boldsymbol{z}))$.

Broader Impact

In this work we propose a novel regularization for autoencoders promoting isometry, and show results in different tasks of dimensionality reduction such as data visualization. The methods developed in this paper could be used for interdisciplinary research. In fact, this work came to be from a discussion with a neuro-scientist that studies phenomena in high dimensional unsupervised data, such as neural activity in the brain.

References

[Alain and Bengio, 2014] Alain, G. and Bengio, Y. (2014). What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593.

[Ambrosio and Soner, 1994] Ambrosio, L. and Soner, H. M. (1994). Level set approach to mean curvature flow in arbitrary codimension.

[Belkin and Niyogi, 2002] Belkin, M. and Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pages 585–591.

[Burda et al., 2016] Burda, Y., Grosse, R. B., and Salakhutdinov, R. (2016). Importance weighted autoencoders. *CoRR*, abs/1509.00519.

[Coifman and Lafon, 2006] Coifman, R. R. and Lafon, S. (2006). Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30.

[Coifman et al., 2005] Coifman, R. R., Lafon, S., Lee, A. B., Maggioni, M., Nadler, B., Warner, F., and Zucker, S. W. (2005). Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the national academy of sciences*, 102(21):7426–7431.

[Donoho and Grimes, 2003] Donoho, D. L. and Grimes, C. (2003). Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596.

[Ghosh et al., 2020] Ghosh, P., Sajjadi, M. S. M., Vergari, A., Black, M., and Scholkopf, B. (2020). From variational to deterministic autoencoders. In *International Conference on Learning Representations*.

[Glorot et al., 2011] Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.

[Heusel et al., 2017] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637.

[Higgins et al., 2017] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. *Iclr*, 2(5):6.

[Hinton and Roweis, 2003] Hinton, G. E. and Roweis, S. T. (2003). Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 857–864.

[Kato et al., 2019] Kato, K., Zhou, J., Sasaki, T., and Nakagawa, A. (2019). Rate-distortion optimization guided autoencoder for isometric embedding in euclidean latent space. *arXiv preprint arXiv:1910.04329*.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. *CoRR*, abs/1312.6114.

[Krizhevsky et al., 2009] Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

[Kruskal, 1964] Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

[Kumar and Poole, 2020] Kumar, A. and Poole, B. (2020). On implicit regularization in β-vaes. *arXiv preprint arXiv:2002.00041*.

[Kumar et al., 2020] Kumar, A., Poole, B., and Murphy, K. (2020). Regularized autoencoders via relaxed injective probability flow. *arXiv preprint arXiv:2002.08927*.

[LeCun, 1998] LeCun, Y. (1998). The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

[Liu et al., 2015] Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738.

[Maaten and Hinton, 2008] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

[Makhzani et al., 2015] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

[McInnes et al., 2018] McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

[McQueen et al., 2016] McQueen, J., Meila, M., and Joncas, D. (2016). Nearly isometric embedding by relaxation. In *Advances in Neural Information Processing Systems*, pages 2631–2639.

[Nash, 1956] Nash, J. (1956). The imbedding problem for riemannian manifolds. *Annals of mathematics*, pages 20–63.

[Nene et al., 1996] Nene, S. A., Nayar, S. K., Murase, H., et al. (1996). Columbia object image library (coil-20).

[Pai et al., 2019] Pai, G., Talmon, R., Bronstein, A., and Kimmel, R. (2019). Dimal: Deep isometric manifold learning using sparse geodesic sampling. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 819–828. IEEE.

[Park et al., 2019] Park, Y., Kim, C. D., and Kim, G. (2019). Variational laplace autoencoders. In *ICML*.

[Paszke et al., 2017] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

[Pearson, 1901] Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

[Peterfreund et al., 2020] Peterfreund, E., Lindenbaum, O., Dietrich, F., Bertalan, T., Gavish, M., Kevrekidis, I. G., and Coifman, R. R. (2020). Loca: Local conformal autoencoder for standardized data coordinates. *arXiv preprint arXiv:2004.07234*.

[Poole et al., 2014] Poole, B., Sohl-Dickstein, J., and Ganguli, S. (2014). Analyzing noise in autoencoders and deep networks. *arXiv preprint arXiv:1406.1831*.

[Ranzato et al., 2008] Ranzato, M., Boureau, Y.-L., and Cun, Y. L. (2008). Sparse feature learning for deep belief networks. In *Advances in neural information processing systems*, pages 1185–1192.

[Ranzato et al., 2007] Ranzato, M., Poultney, C., Chopra, S., and Cun, Y. L. (2007). Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems*, pages 1137–1144.

[Rifai et al., 2011a] Rifai, S., Muller, X., Glorot, X., Mesnil, G., Bengio, Y., and Vincent, P. (2011a). Learning invariant features through local space contraction. *arXiv preprint arXiv:1104.4153*.

[Rifai et al., 2011b] Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. (2011b). Contractive auto-encoders: Explicit invariance during feature extraction. In *ICML*.

[Roweis and Saul, 2000] Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326.

[Sammon, 1969] Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on computers*, 100(5):401–409.

[Sønderby et al., 2016] Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. (2016). Ladder variational autoencoders. In *Advances in neural information processing systems*, pages 3738–3746.

[Tenenbaum et al., 2000] Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323.

[Tolstikhin et al., 2018] Tolstikhin, I. O., Bousquet, O., Gelly, S., and Schölkopf, B. (2018). Wasserstein auto-encoders. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[Ulyanov, 2016] Ulyanov, D. (2016). Multicore-tsne. `https://github.com/DmitryUlyanov/Multicore-TSNE`.

[Vincent et al., 2010] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408.

[Xiao et al., 2017] Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

[Zhan et al., 2018] Zhan, Y., Yu, J., Yu, Z., Zhang, R., Tao, D., and Tian, Q. (2018). Comprehensive distance-preserving autoencoders for cross-modal retrieval. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, page 1137–1145, New York, NY, USA. Association for Computing Machinery.

[Zhao et al., 2019] Zhao, S., Song, J., and Ermon, S. (2019). Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5885–5892.

6 Appendix

6.1 Implementation details

All experiments were conducted on a Tesla V100 Nvidia GPU using PYTORCH framework [Paszke et al., 2017].

6.1.1 Notations

Table 3 describes the notation for the different network layers.

Notation	Description
LIN n	Linear layer. n denotes the output dimension.
FC n	FullyConnected layer with SoftPlus ($\beta = 100$) non linear activation. n denotes the output dimension.
FC_B n	Block consisting of Lin n, followed by a batch normalization layer and SoftPlus ($\beta = 100$) non linear activation.
CONV c, k, s, p	Convolutional layer with kernel of size $k \times k$, c output channels, s stride, and p padding.
CONV_B c, k, s, p	Block consisting of CONV c, k, s, p, followed by a batch normalization layer and SoftPlus($\beta = 100$) non linear activation.
CONVT c, k, s, p	Convolutional transpose layer with kernel of size $k \times k$, c output channels, s stride, and p padding.
CONVT_B c, k, s, p	Block consisting of CONVT c, k, s, p, followed by a batch normalization layer and SoftPlus($\beta = 100$) non linear activation.

Table 3: Layers notation.

6.1.2 Evaluation

Architecture. We used an autoencoder consisted of 5 FC 256 layers followed by a LIN 2 layer for the encoder; similarly, 5 FC 256 layers followed by a LIN 3 layer were used for the decoder.

Training details. All methods were trained for a relatively long period of 100K epochs. Training was done with the ADAM optimizer [Kingma and Ba, 2014], setting a fixed learning rate of 0.001 and a full batch. I-AE parameters were set to: $\lambda_{\text{rec}} = 100$, $\lambda_{\text{inv}} = 0$, $\lambda_{\text{iso}} = 0.1$, $\lambda_{\text{piso}} = 0.1$.

Baselines. The following regularizers were used as baselines: Contractive autoencoder (CAE) [Rifai et al., 2011b]; Contractive autoencoder with decoder weights tied to the encoder weights (TCAE) [Rifai et al., 2011a]; Gradient penalty on the decoder (RAE-GP) [Ghosh et al., 2020]; Denoising autoencoder with gaussian noise (DAE) [Vincent et al., 2010]. For both CAE, and TCAE

the regularization term is $\|dg(\boldsymbol{x})\|^2$. For RAE-GP the regularization term is $\|df(\boldsymbol{z})\|^2$. For U-MAP [McInnes et al., 2018], we set the number of neighbours to 30. For t-SNE [Maaten and Hinton, 2008], we set perplexity$= 50$.

6.1.3 Data visualization

Architecture. Table 4 lists the complete architecture details of this experiment.

MNIST		FMNIST		COIL20	
Encoder	Decoder	Encoder	Decoder	Encoder	Decoder
FC_B 128	FC_B 1024	CONV_B 128,4,2,1	CONVT_B 512,4,1,0	CONV_B 128,4,2,1	CONVT_B 2048,4,1,0
FC_B 256	FC_B 512	CONV_B 256,4,2,1	CONVT_B 256,4,2,1	CONV_B 256,4,2,1	CONVT_B 1024,4,2,1
FC_B 512	FC_B 256	CONV_B 512,4,2,1	CONVT_B 128,4,2,1	CONV_B 512,4,2,1	CONVT_B 512,4,2,1
FC_B 1024	FC_B 128	CONV_B 1024,4,2,1	CONVT 1,4,2,1	CONV_B 1024,4,2,1	CONVT_B 256,4,2,1
LIN 2	LIN 784	CONV 2,2,2,1		CONV_B 2048,4,2,1	CONVT_B 128,4,2,1
				CONV_B 4096,4,2,1	CONVT 1,4,2,1
				CONV 2,2,2,1	

Table 4: High dimensional visualization experiment architectures.

Training details. Training was done using ADAM optimizer [Kingma and Ba, 2014], with a fixed learning rate of 0.001 and batch size of 128. MNIST, and COIL100 dataset were trained for 1000 epochs on all autoencoders, and FMNIST was trained for 500 epochs. I-AE parameters, were set to: $\lambda_{\text{rec}} = 10$, $\lambda_{\text{inv}} = 0.1$, $\lambda_{\text{iso}} = 1$, $\lambda_{\text{piso}} = 1$.

Baselines. The following regularizers were used as baselines: Contractive autoencoder (CAE) [Rifai et al., 2011b]; Gradient penalty on the decoder (RAE-GP) [Ghosh et al., 2020]; Denoising autoencoder with gaussian noise (DAE) [Vincent et al., 2010]. For CAE the regularization term is $\|dg(\boldsymbol{x})\|^2$. For RAE-GP the regularization term is $\|df(\boldsymbol{z})\|^2$. We used U-MAP [McInnes et al., 2018] official implementation with random_state $= 42$, and [Ulyanov, 2016] multicore implementation for t-SNE [Maaten and Hinton, 2008] with default parameters.

6.1.4 Generalization in high dimensional space

Architecture. For all methods, we used an autoencoder with Convolutional and Convolutional transpose layers. Table 5 lists the complete details.

MNIST		CIFAR-10		CelebA	
Encoder	Decoder	Encoder	Decoder	Encoder	Decoder
CONV_B 128, 4, 2, 1	FC 16384	CONV_B 128, 4, 2, 1	FC 16384	CONV_B 128, 5, 2, 1	FC 65536
CONV_B 256, 4, 2, 1	CONVT_B 512, 4, 2, 1	CONV_B 256, 4, 2, 1	CONVT_B 512, 4, 2, 1	CONV_B 256, 5, 2, 1	CONVT_B 512, 4, 2, 1
CONV_B 512, 4, 2, 1	CONVT_B 256, 4, 2, 1	CONV_B 512, 4, 2, 1	CONVT_B 256, 4, 2, 1	CONV_B 512, 5, 2, 1	CONVT_B 256, 4, 2, 1
CONV_B 1024, 4, 2, 1	CONVT_B 128, 4, 2, 1	CONV_B 1024, 4, 2, 1	CONVT_B 128, 4, 2, 1	CONV_B 1024, 5, 2, 1	CONVT_B 128, 4, 2, 1
LIN 16	CONVT 1, 1, 0, 0	LIN 128	CONVT 3, 1, 0, 0	LIN 128	CONVT 3, 1, 0, 0

Table 5: High dimensional generalization experiment architectures.

Training details. Training was done with the ADAM optimizer [Kingma and Ba, 2014], setting a learning rate of 0.0005 and batch size 100. I-AE parameters, were set to: $\lambda_{\text{rec}} = 10$, $\lambda_{\text{inv}} = 0.01$, $\lambda_{\text{iso}} = 0.1$, $\lambda_{\text{piso}} = 0.1$.

Qualitative comparison. In figures 6,7 and 8 we provide test image reconstructions for all the methods listed in the experiment.

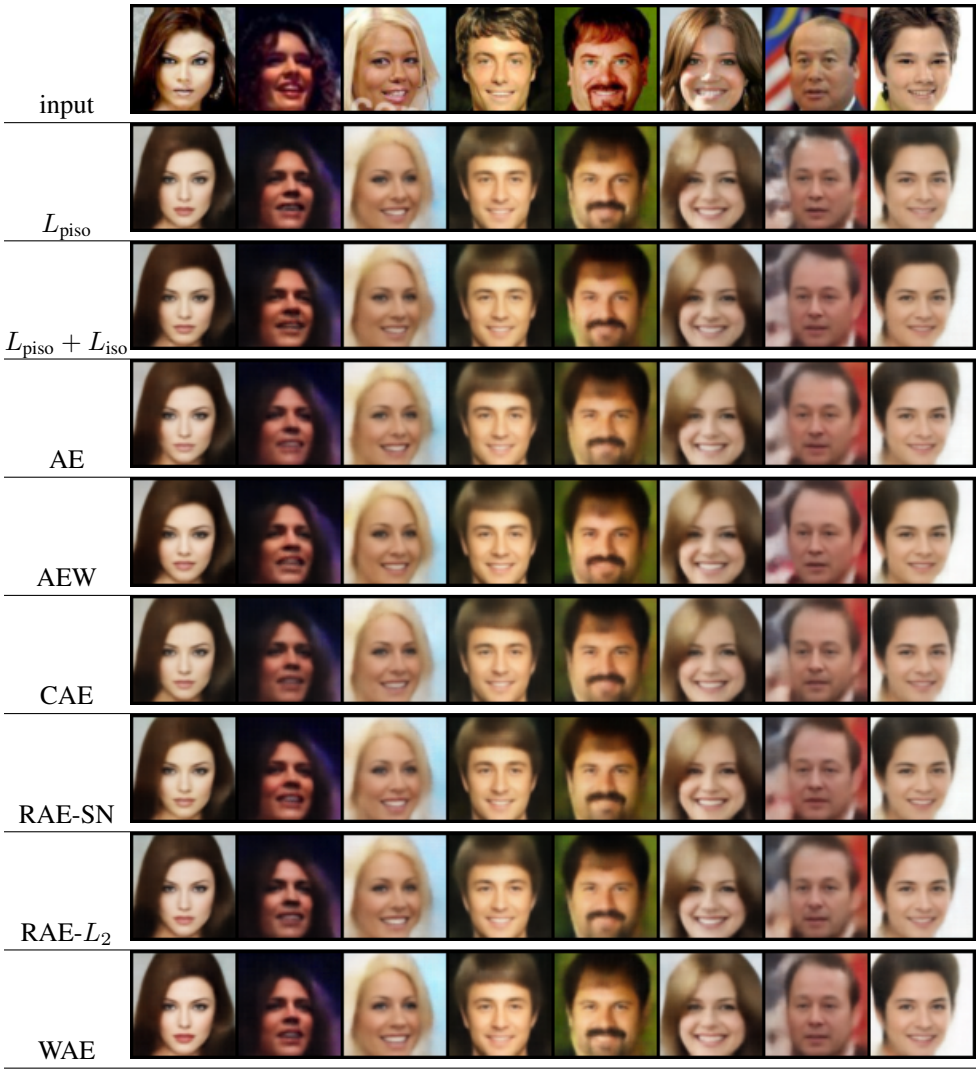

Figure 6: CelebA reconstructions.

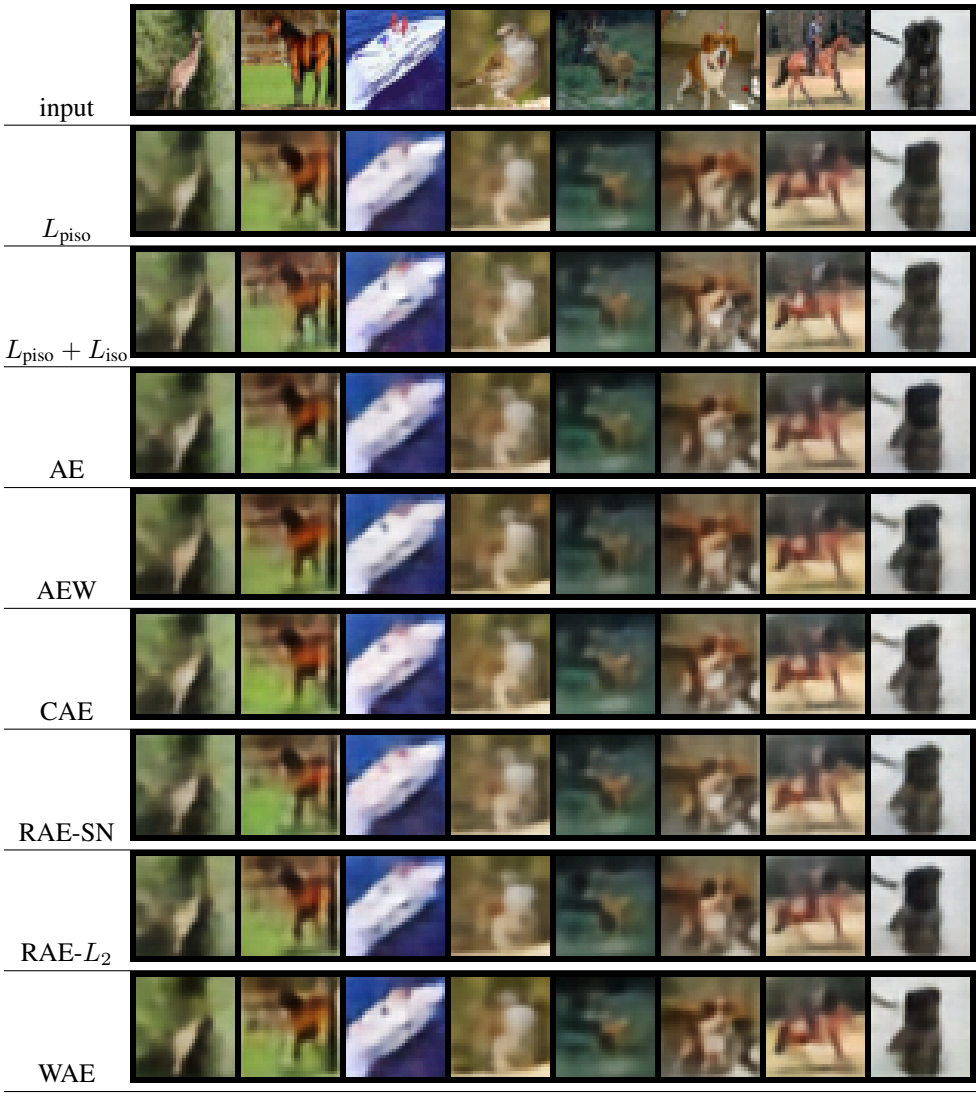

Figure 7: CIFAR-10 reconstructions.

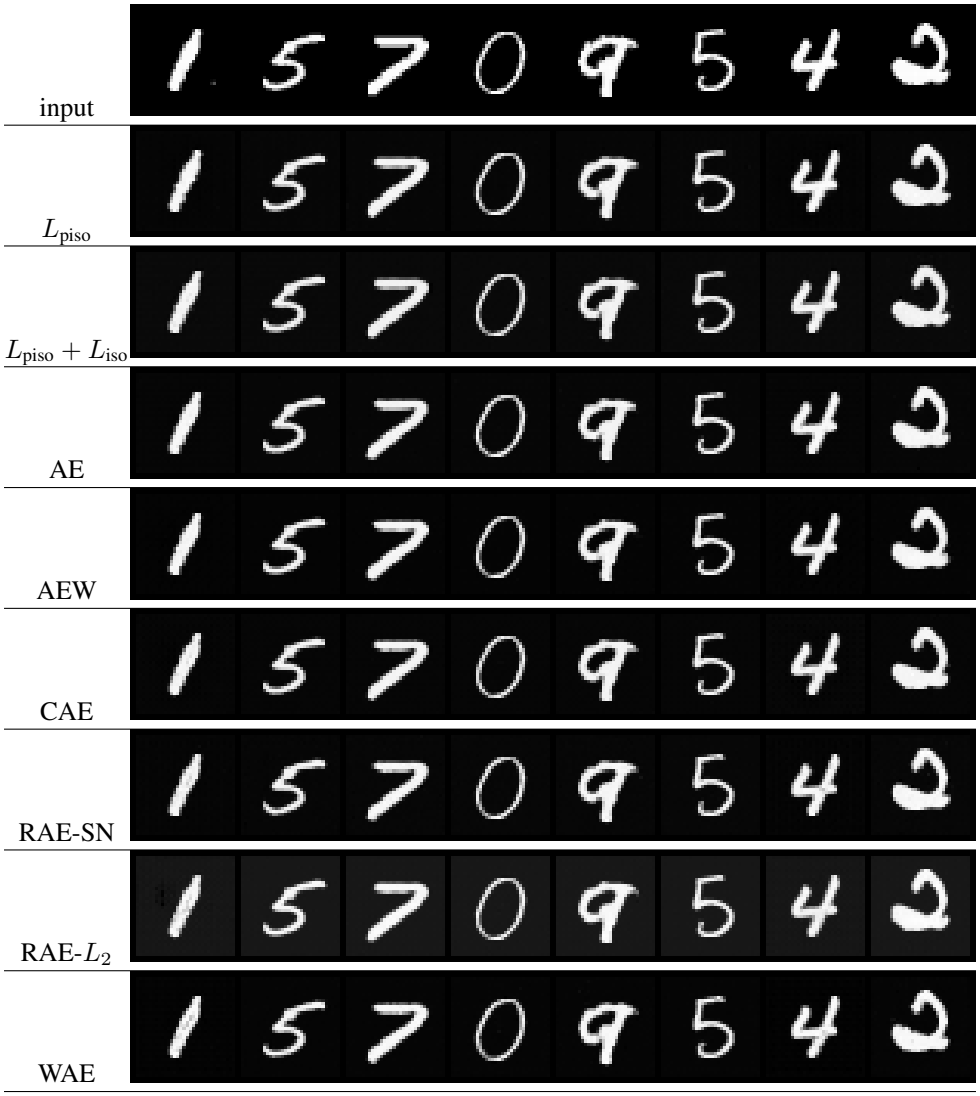

Figure 8: MNIST reconstructions.