GLOBALLY INJECTIVE RELU NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Injectivity plays an important role in generative models where it enables inference; in inverse problems and compressed sensing with generative priors it is a precursor to well posedness. We establish sharp characterizations of injectivity of fully-connected and convolutional ReLU layers and networks. First, through a layerwise analysis, we show that an expansivity factor of two is necessary and sufficient for injectivity by constructing appropriate weight matrices. We show that global injectivity with iid Gaussian matrices, a commonly used tractable model, requires larger expansivity between 3.4 and 5.7. We also characterize the stability of inverting an injective network via worst-case Lipschitz constants of the inverse. We then use arguments from differential topology to study injectivity of deep networks and prove that any Lipschitz map can be approximated by an injective ReLU network. Finally, using an argument based on random projections, we show that an end-to-end—rather than layerwise—doubling of the dimension suffices for injectivity. Our results establish a theoretical basis for the study of nonlinear inverse and inference problems using neural networks.

1 INTRODUCTION

Many applications of deep neural networks require inverting them on their range. Given a neural network $N : \mathcal{Z} \to \mathcal{X}$, where $\mathcal{X}$ is often the Euclidean space $\mathbb{R}^m$ and $\mathcal{Z}$ is a lower-dimensional space, the map $N^{-1} : N(\mathcal{Z}) \to \mathcal{Z}$ is only well-defined when N is injective. The issue of injectivity is particularly salient in two applications: generative models and (nonlinear) inverse problems.

Generative networks model a complicated distribution p_X over $\mathcal{X}$ as a pushforward of a simple distribution p_Z through N. Given an x in the range of N, inference requires computing $p_Z(N^{-1}(x))$ which is well-posed only when N is injective. In the analysis of inverse problems (Arridge et al., 2019), uniqueness of a solution is a key concern; it is tantamount to injectivity of the forward operator. Given a forward model that is known to yield uniqueness, a natural question is whether we can design a neural network that approximates it arbitrarily well while preserving uniqueness. Similarly, in compressed sensing with a generative prior N and a possibly nonlinear forward operator A injective on the range of N, we seek a latent code z such that $A(N(z))$ is close to some measured $y = A(x)$. This is again only well-posed when N can be inverted on its range (Balestriero et al., 2020). Beyond these motivations, injectivity is a fundamental mathematical property with numerous implications. We mention a notable example: certain injective generators can be trained with sample complexity that is polynomial in the image dimension (Bai et al., 2018).

1.1 OUR RESULTS

In this paper we study injectivity of neural networks with ReLU activations. Our contributions can be divided into layerwise results and multilayer results.

Layerwise results. For a ReLU layer $f : \mathbb{R}^n \to \mathbb{R}^m$ we derive sufficient and necessary conditions for invertibility on the range. For the first time, we construct deterministic injective ReLU layers with minimal expansivity $m = 2n$. We then derive specialized results for convolutional layers which are given in terms of filter kernels instead of weight matrices. We also prove upper and lower bounds on minimal expansivity of globally injective layers with iid Gaussian weights. This generalizes certain existing pointwise results (Theorem 2 and Appendix A.2). We finally derive the worst-case inverse

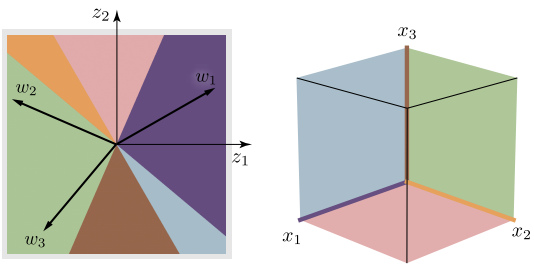
Lipschitz constant for an injective ReLU layer which yields stability estimates in applications to inverse problems.

Multilayer results. A natural question is whether injective models are sufficiently expressive. Using techniques from differential topology we prove that injective networks are *universal* in the following sense: if a neural network $N_1 : \mathcal{Z} \to \mathbb{R}^{2n+1}$ models the data, $\mathcal{Z} \subset \mathbb{R}^n$, then we can approximate N_1 by an injective neural network $N_2 : \mathcal{Z} \to \mathbb{R}^{2n+1}$. As N_2 is injective, the image set $N_2(\mathcal{Z})$ is a Lipschitz manifold. We then use an argument based on random projections to show that an *end-to-end* expansivity by a factor of ≈ 2 is enough for injectivity in ReLU networks, as opposed to layerwise 2-expansivity implied by the layerwise analysis.

We conclude with preliminary numerical experiments to show that imposing injectivity improves inference in GANs while preserving expressivity.

1.2 Why Global Injectivity?

The attribute "global" relates to global injectivity of the map $N : \mathcal{Z} \to \mathbb{R}^m$ on the low-dimensional latent space $\mathcal{Z}$, but it does not imply global invertibility over $\mathbb{R}^m$, only on the range $N(\mathcal{Z}) \subset \mathbb{R}^m$. If we train a GAN generator to map iid normal latent vectors to real images from a given distribution, we expect that any sampled latent vector generates a plausible image. We thus desire that any $N(z)$ be produced by a unique latent code $z \in \mathcal{Z}$. This is equivalent to global injectivity, or invertibility on the range. Our results relate to the growing literature on using neural generative models for compressed sensing (Bora et al., 2017). They parallel the related guarantees for sparse recovery where the role of the low-dimensional latent space is played by the set of all k-sparse vectors. One then looks for matrices which map *all* k-sparse vectors to distinct measurements (Foucart & Rauhut, 2013). As an example, in the illustration in Figure 1 images coresponding to latent codes in orange, brown, and violet wedges cannot be compressively sensed. Finally, a neural network is often trained to directly reconstruct an image x from its (compressive) low-dimensional measurements $y = A(x)$ without introducing any generative models. In this case, whenever A is Lipschitz, it is immediate that the learned inverse must be injective.



$[x_1, x_2, x_3]^T = \mathrm{ReLU}([w_1, w_2, w_3]^T [z_1, z_2]^T)$

Figure 1: An illustration of a ReLU layer $N : \mathbb{R}^2 \to \mathbb{R}^3$, $x = N(z)$, that is not globally injective. Differently colored regions in the z-space are mapped to regions of the same color in the x-space. While N is locally injective in the pink, blue and green wedges in z-space, the orange, brown, and violet wedges are mapped to coordinate axes. N is thus *not* injective on these wedges. This prevents construction of an inverse in the range of N.

1.3 Related Work

Closest to our work are the papers of Bruna et al. (2013), Hand et al. (2018) and Lei et al. (2019). Bruna et al. (2013) study injectivity of pooling motivated by the problem of signal recovery from feature representations. They focus on ℓ^p pooling layers; their Proposition 2.2 gives a criterion similar to the DSS (Definition 1) and bi-Lipschitz bounds for a ReLU layer (similar to our Theorem 3). Unlike Theorems 1 and 3, their criterion and Lipschitz bound are in some cases not precisely aligned with injectivity; see Appendix E.1.

Compressed sensing with GAN priors requires inverting the generator on its range (Bora et al., 2017; Shah & Hegde, 2018; Wu et al., 2019; Mardani et al., 2018; Hand et al., 2018). Lei et al. (2019) replace the end-to-end inversion by the faster and more accurate layerwise inversion when each layer is injective. They show that with high probability a ReLU layer with an iid normal weight matrix can be inverted about a fixed point if the layer expands at least by a factor of 2.1. This result is related to our Theorem 2 which gives conditions for global injectivity or layers with random matrices. Hand & Voroninski (2017) show that when the weights of a ReLU network obey a certain weighted distribution condition, the loss function for the inversion has a strict descent direction everywhere

except in a small set. The condition is in particular satisfied by random matrices with expansivity $n_j = \Omega(n_{j-1} \log n_{j-1})$, where n_j is the output dimension of layer j.

A continuous analogy of our convolutional construction (Definition 3) was considered by Mallat et al. (2018). They show that ReLU acts as a phase filter and that the layer is bi-Lipschitz and hence injective when the filters have a diverse phase and form a frame. Discretizing their model gives a statement related to Corollary 2 and Theorem 4.

Injectivity is automatic in invertible neural networks such as normalizing flows (Grover et al., 2018; Kingma & Dhariwal, 2018; Grathwohl et al., 2018). Specialized architectures with simple Jacobians give easy access to the likelihood (Dinh et al., 2014; 2016; Gomez et al., 2017), which facilitates application to inverse problems (Ardizzone et al., 2018). Inference with GANs can be achieved by jointly training a generative network and its inverse (Donahue et al., 2016; Dumoulin et al., 2016), which is well-defined when the generator is injective. Relaxed injective probability flows resemble GAN generators but are trained via approximate maximum likelihood (Kumar et al., 2020). Injectivity is promoted by keeping the smallest singular value of the Jacobian away from zero at the training examples, a necessary but not sufficient condition. In general, Jacobian conditioning improves GAN performance (Heusel et al., 2017; Odena et al., 2018). Finally, lack of injectivity interferes with disentanglement (Chen et al., 2016; Lin et al., 2019). In this context, injectivity seems to be a natural heuristic to increase latent space capacity without increasing its dimension (Brock et al., 2016).

Notation. Given a matrix $W \in \mathbb{R}^{m\times n}$ we define the notation $w \in W$ to mean that $w \in \mathbb{R}^n$ is a row vector of W. For a matrix $W \in \mathbb{R}^{m\times n}$ with rows $\{w_j\}_{j=1}^m$ and $x \in \mathbb{R}^n$, we write

$$S(x, W) \coloneqq \{j \in [[m]] \colon \langle w_j, x\rangle \geq 0\} \tag{1}$$

and $S^c(x, W)$ for its complement, with $[[m]] = \{1, \dots, m\}$. We let $\mathcal{NN}(n, m, L, \boldsymbol{m})$ be the family of functions $N_\theta : \mathbb{R}^n \to \mathbb{R}^m$ of the form

$$N(z) = W_{L+1}\phi_L(W_L \cdots \phi_2(W_2\phi_1(W_1 z + b_1) + b_2) \cdots + b_L) \tag{2}$$

Indices $\ell = 1, \dots, L$ index the network layers, $b_\ell \in \mathbb{R}^{n_{\ell+1}}$ are the bias vectors, $W_\ell \in \mathbb{R}^{n_{\ell+1}\times n_\ell}$ are the weight matrices with $n_1 = n$, $n_L = m$, and ϕ_ℓ are the nonlinear activation functions. We will denote $\mathrm{ReLU}(x) = \max(x, 0)$. We write $\boldsymbol{n} = (n_1, n_2, \dots n_{L-1})$ and $\theta = (W_1, b_1, \dots, W_L, b_L)$ for the parameters that determine the function N_θ. We also write $\mathcal{NN}(n, m, L) = \bigcup_{\boldsymbol{n}\in\mathbb{Z}^{L-1}} \mathcal{NN}(n, m, L, \boldsymbol{n})$ and $\mathcal{NN}(n, m) = \bigcup_{L\in\mathbb{Z}} \mathcal{NN}(n, m, L)$.

2 Layerwise Injectivity of ReLU Networks

For a one-to-one activation function such as a leaky ReLU or a $\tanh$, it is easy to see that injectivity of $x \mapsto W_i x$ implies the injectivity of the layer. We therefore focus on non-injective ReLU activations.

2.1 Directed Spanning Set

Unlike in the case of a one-to-one activation, when $\phi(x) = \mathrm{ReLU}(x)$, $x \mapsto \phi(Wx)$ cannot be injective for all $x \in \mathbb{R}^n$ if W is a square matrix. Noninjectivity of $\mathrm{ReLU}(Wx)$ occurs in some simple situations, for example if Wx is not full rank or if $Wx \leq 0$ element-wise for a nonzero x, but these two conditions on their own are not sufficient to characterize injectivity. Figure 2 gives an example of a weight matrix that yields a noninjective layer that passes these two tests. In order to facilitate the full characterization of injectivity of ReLU layers, we define a useful theoretical device:

Definition 1 (Directed Spanning Set)**.** Let $W \in \mathbb{R}^{n\times m}$. We say that W has a directed spanning set (DSS) of $\Omega \subset \mathbb{R}^n$ with respect to a vector $x \in \mathbb{R}^n$ if there exists a $\hat{W}_x$ such that each row vector of $\hat{W}_x$ is a row vector of W,

$$\langle x, w_i\rangle \geq 0 \quad \text{for all } w_i \in \hat{W}_x, \tag{3}$$

and $\Omega \subset \mathrm{span}(\hat{W}_x)$. When omitted Ω is understood to be $\mathbb{R}^n$.

For a particular $x \in \mathbb{R}^n$, it is not hard to verify if a given $W = \{w_i\}_{i=1,\dots,m}$ has a DSS of Ω. One can simply let $\hat{W}_x = \{w_i \in \mathbb{R}^n \colon \langle x, w_i\rangle \geq 0\}$. Then, W has a DSS for Ω with respect to x if and only if $\hat{W}_x$ is a basis of Ω, which can be checked efficiently. Note that having full rank is necessary for it

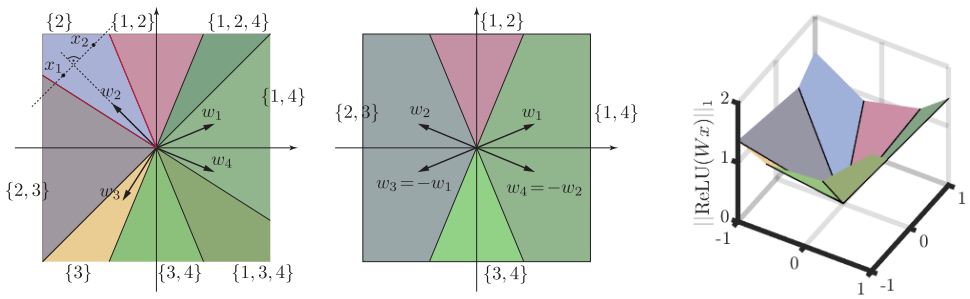


Figure 2: Illustration of the DSS definition. *Left:* A configuration of 4 vectors in $\mathbb{R}^2$ that do not have a DSS w.r.t. all $x \in \mathbb{R}^2$. In this case, the vectors do not generate an injective layer. The set of labels indicate which w_j have positive inner product with vectors in the wedge; there are two wedges with only one such $\{w_j\}$; we have $\mathrm{ReLU}(Wx_1) = \mathrm{ReLU}(Wx_2)$. *Center:* A configuration where four vectors have a DSS for all $x \in \mathbb{R}^2$. These vectors correspond to a minimally-expansive injective layer; see Corollary 2. *Right:* A plot of $\|\mathrm{ReLU}(Wx)\|_1$ where W is given as in the left figure. Note that $x \mapsto \mathrm{ReLU}(Wx)$ is linear in every wedge.

to have a DSS for *any* $x \in \mathbb{R}^n$. It is however not sufficient. For example, $\mathrm{Id}_n \in \mathbb{R}^{n\times n}$, the identity matrix in $\mathbb{R}^n$, is clearly full rank, but it doesn't have a DSS of $\mathbb{R}^n$ w.r.t. $x = (-1, 0, \ldots, 0) \in \mathbb{R}^n$.

To check whether W has a DSS for all $x \in \mathbb{R}^n$, note that W partitions $\mathbb{R}^n$ into open wedges S_k, $\mathbb{R}^n = \bigcup_k S_k$, with constant sign patterns. That is, for $x_1, x_2 \in S_k$, $\mathrm{sign}(Wx_1) = \mathrm{sign}(Wx_2)$. (See also proof of Theorem 2 in Appendix A.2.2.) Checking whether W has a DSS for all x is equivalent to checking that for every wedge there are at least n vectors $W_k \subset W$ with positive sign, $\langle w, x\rangle > 0$ for $x \in S_k$, and that W_k spans $\mathbb{R}^n$. Since the number of wedges can be exponential in m and n (Winder, 1966) this suggests an exponential time algorithm. We also note the connection between DSS and spark in compressed sensing (Foucart & Rauhut, 2013), defined as the size of the smallest set of linearly dependent vectors in W. If every wedge has n or more positive signs, then full spark $n + 1$ is sufficient for W to have a DSS w.r.t all $x \in \mathbb{R}^n$. Computing spark is known to be NP-hard (Tillmann & Pfetsch, 2013); whether one can do better for DSS remains an open question.

2.2 Fully Connected Layer

The notion of a DSS immediately leads to our main result for fully connected layers.

Theorem 1 (Conditions for Injectivity of $\mathrm{ReLU}(Wx)$)**.** *Let $W \in \mathbb{R}^{m\times n}$ where $n > 1$ be a matrix with row vectors $\{w_j\}_{j=1}^m$, and $\mathrm{ReLU}(y) = \max(y, 0)$. The function $\mathrm{ReLU}(W(\cdot))\colon \mathbb{R}^n \to \mathbb{R}^m$ is injective if and only if W has a DSS w.r.t every $x \in \mathbb{R}^n$.*

The question of injectivity in the case when $b = 0$ and when $b \neq 0$ are very similar and, as Lemma 1 shows, the latter question is equivalent to the former on a restricted weight matrix.

Lemma 1 (Injectivity of $\mathrm{ReLU}(Wx+b)$)**.** *Let $W \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. The function $\mathrm{ReLU}(W(\cdot)+b)\colon \mathbb{R}^n \to \mathbb{R}^m$ is injective if and only if $\mathrm{ReLU}(W|_{b\geq 0}\cdot)$ is injective, where $W|_{b\geq 0} \in \mathbb{R}^{m\times n}$ is row-wise the same as W where $b_i \geq 0$, and is a row of zeroes when $b_i < 0$.*

Remark 1 (Injectivity of $\mathrm{ReLU}(Wx)$, Positive x). Without normalization strategies, inputs of all but the first layer are element-wise non-negative. One can ask whether $\mathrm{ReLU}(Wx)$ is injective when x is restricted to be element-wise non-negative. Following the same argument as in Theorem 1, we find that W must have a DSS w.r.t. $x \in \mathbb{R}^n$ for every x that is element-wise non-negative. With normalizations strategies however (for example batch renormalization (Ioffe, 2017)) the full power of Theorem 1 and Lemma 1 may be necessary. In Appendix B we show that common normalization strategies such as batch, layer, or group normalization do not interfere with injectivity.

Remark 2 (Global vs Restricted Injectivity). Even when the conditions of Theorem 1 and Lemma 1 are not satisfied the network might be injective in some $\mathcal{X} \subset \mathbb{R}^n$. Indeed, $N(x) = \mathrm{ReLU}(\mathrm{Id}_n\cdot)$ is in general not injective, but it is injective in $\mathcal{X} = \{x \in \mathbb{R}^n\colon x_i > 0 \text{ for } i = 1, \ldots, n\}$, a convex and open set. Theorem 1 and Lemma 1 are the precise criteria for single-layer injectivity for all $x \in \mathbb{R}^n$.

The above layerwise results imply a sufficient condition for injectivity of deep neural networks.

Corollary 1 (layerwise Injectivity Implies End-to-End Injectivity)**.** *Let $N\colon \mathbb{R}^n \to \mathbb{R}^m$ be a deep neural network of the form (2). If each layer $\phi_\ell(W_\ell \cdot + b_\ell)\colon \mathbb{R}^{n_i} \to \mathbb{R}^{n_{i+1}}$ is injective, then so is N.*

Note that Corollary 1 is sufficient but not necessary for injectivity of the entire network. Consider for example an injective layer $\mathrm{ReLU}(W)$, and $N(x) = \mathrm{ReLU}(\mathrm{Id}_m \cdot \mathrm{ReLU}(W(x)))$, where I is the identity matrix. Clearly,

$$\mathrm{ReLU}(\mathrm{Id}_m \cdot \mathrm{ReLU}(W(x))) = \mathrm{ReLU}(\mathrm{ReLU}(W(x))) = \mathrm{ReLU}(W(x)) \tag{4}$$

so N is injective, but it fails the criterion of Corollary 1 (at the $\mathrm{ReLU}(I\cdot)$ layer).

An important implication of Theorem 1 is the following result on minimal expansivity.

Corollary 2 (Minimal Expansivity)**.** *For any $W \in \mathbb{R}^{m\times n}$, $\mathrm{ReLU}(W\cdot)$ is non-injective if $m < 2 \cdot n$. If $W \in \mathbb{R}^{2n\times n}$ satisfies Theorem 1 and Lemma 1, than up to row rearrangement W can be written as*

$$W = \begin{bmatrix} B \\ -DB \end{bmatrix} \tag{5}$$

where $B, D \in \mathbb{R}^{n\times n}$ and B is a basis and D a diagonal matrix with strictly positive diagonal entries.

While $m < 2n$ immediately precludes injectivity, Corollary 2 gives a simple recipe for the construction of minimally-expansive injective layers with $m = 2n$. To the best of our knowledge this is the first such result. Further, we can also use Corollary 2 to build weight matrices for which $m > 2n$. For example, the matrix

$$W = \begin{bmatrix} B \\ -DB \\ M \end{bmatrix},$$

where $B \in \mathbb{R}^{n\times n}$ has full rank, $D \in \mathbb{R}^{n\times n}$ is a positive diagonal matrix, and $M \in \mathbb{R}^{m-2n\times n}$ is arbitrary, has a DSS w.r.t. every $x \in \mathbb{R}^n$. This method can be used in practice to generate injective layers by using standard spectral regularizers to ensure that B is a basis (Cisse et al., 2017). In Section 3 we will show that, in fact, an *end-to-end* rather than layerwise doubling of dimension is sufficient for injectivity.

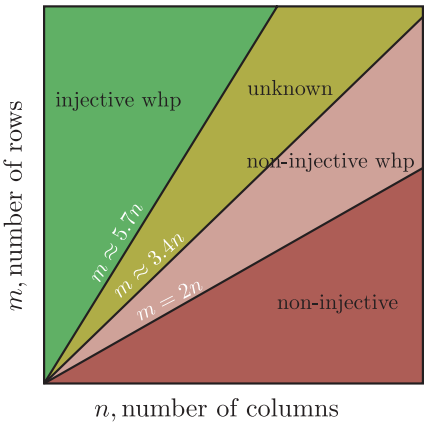


Figure 3: A visualization of regions where asymptotics of $\mathcal{I}(m, n)$ in Theorem 2 are valid.

Previous work that sought injectivity uses random-weight models. Lei et al. (2019) show that a layer is invertible *about a point* in the range provided that $m \geq 2.1n$ and W is iid Gaussian. In Appendix A.3 we show that $m = 2.1n$ is not enought to guarantee injectivity. In fact, we show that with high probability an iid Gaussian weight matrix W yields a *globally* invertible ReLU layer, when W is sufficiently expansive, and conversely does not satisfy Theorem 1 if it is not expansive by at least a factor of 3.4:

Theorem 2 (Injectivity for Gaussian Weights)**.** *Let $\mathcal{I}(m, n) = \mathbb{P}\{x \mapsto \mathrm{ReLU}(Wx)$ is injective$\}$ with the entries of W iid standard normal and $c = m/n$ fixed. Then as $n \to \infty$,*

$$\mathcal{I}(m, n) \to 1 \text{ if } c \gtrapprox 5.7 \text{ and } \mathcal{I}(m, n) \to 0 \text{ if } c \lessapprox 3.4.$$

The parameter regions defined in Theorem 2 are illustrated in Figure 3.

We note that the study of neural networks with random weights is an important recent research direction (Pennington & Worah, 2017; Benigni & Péché, 2019). Pennington & Worah (2017) articulate this clearly, arguing that large complex systems, of which deep networks are certainly an example, can be profitably studied by approximating their constituent parts as random variables.

2.3 Inverse Lipschitz Constant for Networks

Because $\|\mathrm{ReLU}(Wx) - \mathrm{ReLU}(Wy)\| \leq \|W\|\,\|x - y\|$, it is clear that $\mathrm{ReLU}(W\cdot)$ is Lipschitz with constant $\|W\|$; whether the inverse is Lipschitz, and if so, with what Lipschitz constant, is less obvious. We can prove the following result (see Appendix C.1):

Theorem 3 (Global Inverse Lipschitz Constant)**.** *Let $W \in \mathbb{R}^{m\times n}$ have a DSS w.r.t. every $x \in \mathbb{R}^n$. There exists a $C(W) > 0$ such that for any $x_0, x_1 \in \mathbb{R}^n$,*

$$\|\mathrm{ReLU}(Wx_0) - \mathrm{ReLU}(Wx_1)\|_2 \geq C(W)\,\|x_0 - x_1\|_2 \tag{6}$$

where $C(W) = \frac{1}{\sqrt{2m}} \min_{x\in\mathbb{R}^n} \sigma(W|_{S(x,W)})$, and σ denotes the smallest singular value.

This result immediately yields stability estimates when solving inverse problems by inverting (injective) generative networks. We note that the $(2m)^{-1/2}$ factor is essential; the naive "smallest singular value" estimate is too optimistic.

2.4 Convolutional Layer

Since convolution is a linear operator we could simply apply Theorem 1 and Lemma 1 to the convolution matrix. It turns out, however, that there exists a result specialized to convolution that is much simpler to verify. In this section, capital letters such as I, J and K denote multi-indexes; the same holds for their element-wise limits such as N and M. N_j refers to the jth index in the multi-index N. We use the shortcuts $c \in \mathbb{R}^N$ for $c \in \mathbb{R}^{N_1} \times \cdots \times \mathbb{R}^{N_p}$, and $\sum_{I=1}^{N} c_I = \sum_{I_1=1}^{N_1} \cdots \sum_{I_p=1}^{N_p} c_{I_1,\ldots,I_p}$, for $I = (I_1, \ldots, I_p)$. The symbol 1 can refer to number 1 or a p-tuple $(1, \ldots, 1)$. Further, the notation $I \leq J$ means that $I_k \leq J_k$ for all $k = 1, \ldots, p$ and likewise for $\geq, <$ and $>$. If $I \not\leq J$, then there is at least one k such that $I_k > J_k$.

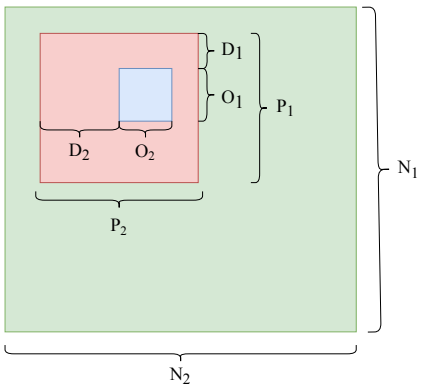


Figure 4: A visualization of the indices in (8) in two dimensions. The blue region is a kernel c of width O, the pink region is the zero-padded box of width P, and D is the offset of the kernel c in the pink box. The entire signal is the green box of width N.

Definition 2 (Convolution Operator)**.** Let $c \in \mathbb{R}^O$. We say that c is a convolution kernel of width O. Given c and $x \in \mathbb{R}^N$, we define the convolution operator $C \in \mathbb{R}^{N\times N}$ with stride 1 as,

$$(Cx)_J = \sum_{I=1}^{O} c_{O-I+1} x_{J+I} = \sum_{I'=1+J}^{O+J} c_{O+J-I'+1} x_{I'}. \tag{7}$$

When $1 \not\leq K$ or $K \not\leq O$ we will set $c_K = 0$. We do not specify the boundary condition on x (zero-padded, periodic, or otherwise), as our results hold generally.

Definition 3 (Convolutional Layer)**.** We say that a matrix $W \in \mathbb{R}^{M\times N}$ is a convolution operator, with n_Q convolutions, provided that W can be written (up to row rearrangement) in the form $W = \begin{bmatrix} C_1^T & C_2^T & \cdots & C_{n_Q}^T \end{bmatrix}^T$ where each C_k is a convolution operator for $k = 1, \ldots, n_Q$. A neural network layer for which W is a convolution operator is called a convolution layer.

Definitions 2 and 3 automatically model the standard multi-channel convolution used in practice. For 2D images of size 512×512, n_c input channels, and 3×3 convolutions, we simply let $x \in \mathbb{R}^N$ with $N = (512, 512, n_c)$ and $c \in \mathbb{R}^O$ with $O = (3, 3, n_c)$.

To state our main result for convolutions we also need to define the set of zero-padded kernels for $c \in \mathbb{R}^O$. Think of a multi-index as a box (or a hyperrectangle). Let P be a multi-index such that O "fits" in P. Then we define

$$\mathcal{Z}_P(c) = \{D \in \mathbb{R}^P \ : \ d \text{ is a shift of } c \text{ within the box } P\}. \tag{8}$$

Theorem 4 (Sufficient Condition for Injectivity of Convolutional Layer)**.** *Suppose that $W \in \mathbb{R}^{M\times N}$ is a convolution layer with convolutions $\{C_k\}_{k=1}^q$, and corresponding kernels $\{c_k\}_{k=1}^q$. If for any P,*

$$W|_{\mathcal{Z}_P} := \bigcup_{k=1}^{q} \mathcal{Z}_P(c_k) \tag{9}$$

has a DSS for $\mathbb{R}^P$ with respect to all $x \in \mathbb{R}^P$, then $\mathrm{ReLU}(W\cdot)$ satisfies Theorem 1.

Theorem 4 applies to multi-channel convolution of width $(O, n_c) = (O_1, \ldots, O_p, n_c)$ provided that we choose a (P, n_c) such that $O \leq P$. This theorem shows that if a convolution operator W has a DSS w.r.t. vectors with support in the pink region in Figure 4, then it has a DSS w.r.t. vectors supported on the green region as well.

3 Universality and Expansivity of Deep Injective Networks

We now consider general properties of deep injective networks. Note that a neural network $N_\theta \in \mathcal{NN}(n, m)$ is Lipschitz smooth: there is $L_0 > 0$ such that $|N_\theta(x) - N_\theta(y)| \leq L_0|x - y|$ for all $x, y \in \mathbb{R}^n$. If $N_\theta : \mathbb{R}^n \to \mathbb{R}^m$ is also injective, results of basic topology imply that for any bounded and closed set $B \subset \mathbb{R}^n$ the map $N_\theta : B \to N_\theta(B)$ has a continuous inverse $N_\theta^{-1} : N_\theta(B) \to B$ and the sets B and its image $N_\theta(B)$ are homeomorphic. Thus, for example, if Z is a random variable supported on the cube $[0, 1]^n \subset \mathbb{R}^n$, the sphere $\mathbb{S}^{n-1} \subset \mathbb{R}^n$, or the torus $\mathbb{T}^{n-1} \subset \mathbb{R}^n$, we see that $N_\theta(Z)$ is a random variable supported on a set in $\mathbb{R}^m$ that is homeomorphic to an n-dimensional cube or an $n - 1$ dimensional sphere or a torus, respectively. This means that injective neural networks can model random distributions on surfaces with prescribed topology. Moreover, if $N_\theta : \mathbb{R}^n \to \mathbb{R}^m$ is a decoder, all objects $x \in N_\theta(\mathbb{R}^n)$ correspond to the unique code z such that $x = N_\theta(z)$.

A fundamental property of neural networks is that they can uniformly approximate any continuous function $f : \mathcal{Z} \to \mathbb{R}^m$ defined in a bounded set $\mathcal{Z} \subset \mathbb{R}^n$. For general dimensions n and m, the injective neural networks do not have an analogous property. For instance, if $f : [-\pi, \pi] \to \mathbb{R}$ is the trigonometric function $f(x) = \sin(x)$, it is easy to see that there is no injective neural network (or any other injective function) $N_\theta : [-\pi, \pi] \to \mathbb{R}$ such that $|f(x) - N_\theta(x)| < 1$. Consider, however, the following trick: add two dimensions in the output vector and consider the map $F : [-\pi, \pi] \to \mathbb{R}^3$ given by $F(x) = (0, 0, \sin(x)) \in \mathbb{R}^3$. When f is approximated by a one-dimensional, non-injective ReLU network $f_\theta : [-\pi, \pi] \to \mathbb{R}$ and $\alpha > 0$ is small, then the neural network $N_\theta(x) = (\alpha\phi(x), \alpha\phi(-x), f_\theta(x))$ is an injective map that approximates F.

In general, as ReLU networks are piecewise affine maps, it follows in the case $m = n$ that if a ReLU network $N_\theta : \mathbb{R}^n \to \mathbb{R}^n$ is injective, it has to be surjective (Scholtes, 1996, Thm. 2.1). This is a limitation of injective neural networks when $m = n$. Building on these ideas we show that when the dimension of the range space is sufficiently large, $m \geq 2n + 1$, injective neural networks become sufficiently expressive to universally model arbitrary continuous maps.

Theorem 5 (Universal Approximation with Injective Neural Networks)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a continuous function, where $m \geq 2n + 1$, and $L \geq 1$. Then for any $\varepsilon > 0$ and compact subset $\mathcal{Z} \subset \mathbb{R}^n$ there exists a neural network $N_\theta \in \mathcal{NN}(n, m)$ of depth L such that $N_\theta : \mathbb{R}^n \to \mathbb{R}^m$ is injective and*

$$|f(x) - N_\theta(x)| \leq \varepsilon, \quad \textit{for all } x \in \mathcal{Z}. \tag{10}$$

Before describing the proof, we note that this result also holds for leaky ReLU networks with no modification. The key fact used in the proof is that the network is a piecewise affine function.

To prove this result, we combine the approximation results for neural networks (e.g., Pinkus's density result for shallow networks (Pinkus, 1999, Theorem 3.1) or Yarotsky's result for deep neural networks (Yarotsky, 2017)), with the Lipschitz-smooth version of the generic projector technique. This technique from differential topology is used for example to prove the easy version of the Whitney's embedding theorem (Hirsch, 2012, Chapter 2, Theorem 3.5). Related random projection techniques have been used earlier in machine learning and compressed sensing (Broomhead & Kirby, 2000; 2001; Baraniuk & Wakin, 2009; Hegde et al., 2008; Iwen & Maggioni, 2013).

The proof of Theorem 5 is based on applying the above classical results to locally approximate the function $f : \mathbb{R}^n \to \mathbb{R}^m$ by some (possibly non-injective) ReLU-based neural network $F_\theta : \mathbb{R}^n \to \mathbb{R}^m$, and augment it by adding additional variables, so that $H_\theta(x) = (x, F_\theta(x))$ is an injective map $H_\theta : \mathbb{R}^n \to \mathbb{R}^{n+m}$. The image of this map, $M = H_\theta(\mathbb{R}^n)$, is an n-dimensional, Lipschitz-smooth submanifold of $\mathbb{R}^{n+m}$. The dimension of $m + n$ is larger than $2n + 1$, which implies that for a randomly chosen projector P_1 that maps $\mathbb{R}^{n+m}$ to a subspace of dimension $m + n - 1$, the restriction of P_1 on the submanifold M, that is, $P_1 : M \to P_1(M)$, is injective. By applying n random projectors, $P_1, \ldots, P_n$, we have that $N_\theta = P_n \circ P_{n-1} \circ \cdots \circ P_1 \circ H_\theta$ is an injective map whose image is in a m dimensional linear space. These projectors can be multiplied together, but are

generated sequentially. By choosing the projectors P_j in a suitable way, the obtained neural network N_θ approximates the map f.

We point out that in the proof of Theorem 5 it is crucial that we first approximate function f by a neural network and then apply to it a generic projection to make the neural network injective. Indeed, doing this in the opposite order may fail as an arbitrarily small deformation (in $C(\mathbb{R}^n)$) of an injective map may produce a map that is non-injective.

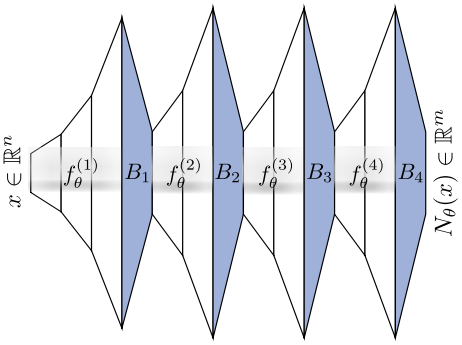


Figure 5: An illustration of an injective deep neural network that avoids expansivity as described by Corollary 3. White trapezoids are expansive weight matrices satisfying Theorem 1, and the blue trapezoids are random projectors that reduce dimension while preserving injectivity.

The proof of Theorem 5 implies the following corollary on cascaded neural networks where the dimensions of the hidden layers can both increase and decrease (see Figure 5):

Corollary 3. *Let $n, m, d_j \in \mathbb{Z}_+$, $j = 0, 1, \ldots, 2k$ be such that $d_0 = n$, $d_{2k} = m \geq 2n+1$ and $d_j \geq 2n+1$ for even indexes $j \geq 2$. Let*

$$F_k = B_k \circ f_\theta^{(k)} \circ B_{k-1} \circ f_\theta^{(k-1)} \circ \cdots \circ B_1 \circ f_\theta^{(1)}$$

where $f_\theta^{(j)} : \mathbb{R}^{d_{2j-2}} \to \mathbb{R}^{d_{2j-1}}$ are injective neural networks and $B_j \in \mathbb{R}^{d_{2j-1}} \to \mathbb{R}^{d_{2j}}$ are random matrices whose joint distribution is absolutely continuous with respect to the Lebesgue measure of $\prod_{j=1}^{k} (\mathbb{R}^{d_{2j} \times d_{2j-1}})$. Then the neural network $F_k : \mathbb{R}^n \to \mathbb{R}^m$ is injective almost surely.

Observe that in Corollary 3 the weight matrices B_j and $B_{j'}$ may not be independent. Moreover, the distribution of the matrices B_j may be supported in an arbitrarily small set, that is, a small random perturbation of the weight matrices make the function F_k injective. Corollary 3 characterizes deep globally injective networks, while avoiding the exponential growth in dimension as indicated by Corollary 2.

4 Numerical Experiment: Injective GANs Improve Inference

We devise a simple experiment to demonstrate that 1) one can construct injective networks using Corollary 2 that 2) perform as well as the corresponding non-injective networks, while 3) being better at inference problems. One way to do inference with GANs is to train a so-called inference network $I_{\text{net}} : \mathcal{X} \to \mathcal{Z}$ jointly with the generator $G : \mathcal{Z} \to \mathcal{X}$ so that $I(G(z)) \approx z$ (Donahue et al., 2016; Dumoulin et al., 2016); I_{net} is trained to invert G on its range. This is used to evaluate the likelihood of an image x in the range of G as $p_Z(I_{\text{net}}(x))$. However, for $p_Z(I_{\text{net}}(G(z)) = p_Z(z)$ to hold, G must be injective—otherwise the distribution of $I_{\text{net}}(G(z))$ will be different from that of z. One would thus hope that the distribution of $I_{\text{net}}(G(z))$ will be closer to that of z if we enforce injectivity. If that is the case and the sample quality is comparable, then we have a practical tool to improve inference.

We use the DCGAN (Radford et al., 2015) architecture with the same hyperparameters, layer sizes and normalization strategies for the regular GAN and the injective GAN; see Appendix F. Note that the original DCGAN uses several ReLU conv layers with an expansivity of 2 per layer. By Corollary 2 these layers are non-injective. Therefore, the filters of the DCGAN are modified to be of the form $[C; -s^2C]$ in order to build an injective GAN. We train the networks to generate $64 \times 64 \times 3$ images and draw the latent code $z \in \mathbb{R}^{256}$ iid from a standard normal distribution. We test on CelebA (Liu et al., 2015) and FFHQ (Karras et al., 2019) datasets. To get a performance metric, we fit Gaussian distributions $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(\mu_{\text{inj}}, \Sigma_{\text{inj}})$ to $G(z)$ and $G_{\text{inj}}(z)$. We then compute the Wasserstein-2 distance $\mathcal{W}_2$ between the distribution of $z \sim \mathcal{N}(0, I_{256})$ and the two fitted Gaussians using the closed-form expression for $\mathcal{W}_2$ between Gaussians, $\mathcal{W}_2^2(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)) = \|\mu_1 - \mu_2\|^2 + \text{Tr}(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{1/2})$. We summarize the results in Table 1. Despite the restrictions on the weights of injective generators, their performance on popular GAN metrics—Fréchet inception distance (FID) (Heusel et al., 2017) and inception score (IS) (Salimans et al., 2016)—is comparable to the standard GAN while inference improves. That the generated samples are indeed comparable to the standard GAN can be also gleaned from the from the figure in Appendix F.

Table 1: Injectivity improves inference without sacrificing performance.

Dataset	Type of G	Inception Score↑	Fréchet Inception Distance↓	$\mathcal{W}_2^2(\mathbb{P}_{\hat{z}}, \mathbb{P}_z) \downarrow$
CelebA	Injective	2.24 ± 0.09	39.33 ± 0.41	$\mathbf{18.59}$
	Regular	2.22 ± 0.16	50.56 ± 0.52	33.85
FFHQ	Injective	2.56 ± 0.15	61.22 ± 0.51	$\mathbf{9.87}$
	Regular	2.57 ± 0.16	47.23 ± 0.90	19.63

5 Conclusion

We derived and explored conditions for injectivity of ReLU neural networks. In contrast to prior work which looks at random weight matrices, our characterizations are deterministic and derived from first principles. They are also sharp in that they give sufficient and necessary conditions for layerwise injectivity. Our results apply to any network of the form (2)—they only involve weight matrices but make no assumptions about the architecture. We included explicit constructions for minimally expansive networks that are injective; interestingly, this simple criterion already improves inference in our preliminary experiments. The results on universality of injective neural networks further justify their use in applications; they also implicitly justify the various Jacobian conditioning strategies when learning to generate real-world data. Further, injective neural networks are topology-preserving homeomorphisms which opens applications in computational topology and establishes connections to tools such as self-organizing maps. Analysis of deep neural networks has an analogue in the analysis of inverse problems where one studies uniqueness, stability and reconstruction. Uniqueness coincides with injectivity, quantitative stability with the Lipschitz constant of the inverse, and, following Lei et al. (2019), from a linear program we get a reconstruction in the range.

REFERENCES

Robert A. Adams. *Sobolev spaces*. Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers], New York-London, 1975. Pure and Applied Mathematics, Vol. 65.

Lynton Ardizzone, Jakob Kruse, Sebastian Wirkert, Daniel Rahner, Eric W Pellegrini, Ralf S Klessen, Lena Maier-Hein, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. *arXiv preprint arXiv:1808.04730*, 2018.

Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Yu Bai, Tengyu Ma, and Andrej Risteski. Approximability of discriminators implies diversity in gans. *arXiv preprint arXiv:1806.10586*, 2018.

Randall Balestriero, Sebastien Paris, and Richard Baraniuk. Max-affine spline insights into deep generative networks. *arXiv preprint arXiv:2002.11912*, 2020.

Richard G. Baraniuk and Michael B. Wakin. Random projections of smooth manifolds. *Found. Comput. Math.*, 9(1):51–77, 2009. ISSN 1615-3375. doi: 10.1007/s10208-007-9011-z. URL https://doi.org/10.1007/s10208-007-9011-z.

Lucas Benigni and Sandrine Péché. Eigenvalue distribution of nonlinear models of random matrices. *arXiv preprint arXiv:1904.03090*, 2019.

Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 537–546. JMLR. org, 2017.

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

D. S. Broomhead and M. Kirby. A new approach to dimensionality reduction: theory and algorithms. *SIAM J. Appl. Math.*, 60(6):2114–2142, 2000. ISSN 0036-1399. doi: 10.1137/S0036139998338583. URL https://doi.org/10.1137/S0036139998338583.

D. S. Broomhead and M. J. Kirby. The whitney reduction network: A method for computing autoassociative graphs. 13(11), 2001. ISSN 0899-7667. doi: 10.1162/089976601753196049. URL https://doi.org/10.1162/089976601753196049.

Joan Bruna, Arthur Szlam, and Yann LeCun. Signal recovery from pooling representations. *arXiv preprint arXiv:1311.4025*, 2013.

Peter Bürgisser and Felipe Cucker. *Condition: The geometry of numerical algorithms*, volume 349. Springer Science & Business Media, 2013.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. *arXiv preprint arXiv:1704.08847*, 2017.

Ştefan Cobzaş, Radu Miculescu, and Adriana Nicolae. *Lipschitz functions*, volume 2241 of *Lecture Notes in Mathematics*. Springer, Cham, 2019. ISBN 978-3-030-16488-1; 978-3-030-16489-8. doi: 10.1007/978-3-030-16489-8. URL https://doi.org/10.1007/978-3-030-16489-8.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Birkhäuser Basel, 2013. ISBN 0817649476.

Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. The reversible residual network: Backpropagation without storing activations. In *Advances in neural information processing systems*, pp. 2214–2224, 2017.

Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Aditya Grover, Manik Dhar, and Stefano Ermon. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pp. 5767–5777, 2017.

Paul Hand and Vladislav Voroninski. Global guarantees for enforcing deep generative priors by empirical risk. *arXiv preprint arXiv:1705.07576*, 2017.

Paul Hand, Oscar Leong, and Vlad Voroninski. Phase retrieval under a generative prior. In *Advances in Neural Information Processing Systems*, pp. 9136–9146, 2018.

Chinmay Hegde, Michael Wakin, and Richard Baraniuk. Random projections for manifold learning. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis (eds.), *Advances in Neural Information Processing Systems 20*, pp. 641–648. Curran Associates, Inc., 2008. URL http://papers.nips.cc/paper/3191-random-projections-for-manifold-learning.pdf.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pp. 6626–6637, 2017.

Morris W Hirsch. *Differential topology*, volume 33. Springer Science & Business Media, 2012.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2): 251–257, 1991.

Sergey Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In *Advances in neural information processing systems*, pp. 1945–1953, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Mark A. Iwen and Mauro Maggioni. Approximation of points on low-dimensional manifolds via random linear projections. *Inf. Inference*, 2(1):1–31, 2013. ISSN 2049-8764. doi: 10.1093/imaiai/iat001. URL https://doi.org/10.1093/imaiai/iat001.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 10215–10224, 2018.

Abhishek Kumar, Ben Poole, and Kevin Murphy. Regularized autoencoders via relaxed injective probability flow. *arXiv preprint arXiv:2002.08927*, 2020.

Qi Lei, Ajil Jalal, Inderjit S Dhillon, and Alexandros G Dimakis. Inverting deep generative models, one layer at a time. In *Advances in Neural Information Processing Systems*, pp. 13910–13919, 2019.

Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

Zinan Lin, Kiran Koshy Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr: Disentangling generative adversarial networks with contrastive regularizers. *arXiv preprint arXiv:1906.06034*, 2019.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 12 2015.

J. Luukkainen and J. Väisälä. Elements of Lipschitz topology. *Ann. Acad. Sci. Fenn. Ser. A I Math.*, 3(1):85–122, 1977. ISSN 0066-1953. doi: 10.5186/aasfm.1977.0315. URL https://doi.org/10.5186/aasfm.1977.0315.

Stéphane Mallat, Sixin Zhang, and Gaspar Rochette. Phase harmonic correlations and convolutional neural networks. *arXiv preprint arXiv:1810.12136*, 2018.

Morteza Mardani, Enhao Gong, Joseph Y Cheng, Shreyas S Vasanawala, Greg Zaharchuk, Lei Xing, and John M Pauly. Deep generative adversarial neural networks for compressive sensing mri. *IEEE transactions on medical imaging*, 38(1):167–179, 2018.

Pertti Mattila. *Geometry of sets and measures in Euclidean spaces*, volume 44 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1995. ISBN 0-521-46576-1; 0-521-65595-1. doi: 10.1017/CBO9780511623813. URL https://doi.org/10.1017/CBO9780511623813. Fractals and rectifiability.

John W. Milnor and James D. Stasheff. *Characteristic classes*. Princeton University Press, Princeton, N. J.; University of Tokyo Press, Tokyo, 1974. Annals of Mathematics Studies, No. 76.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Frank Morgan. *Geometric measure theory: a beginner's guide*. Academic press, 2016.

Augustus Odena, Jacob Buckman, Catherine Olsson, Tom B Brown, Christopher Olah, Colin Raffel, and Ian Goodfellow. Is generator conditioning causally related to gan performance? *arXiv preprint arXiv:1802.08768*, 2018.

Jeffrey Pennington and Pratik Worah. Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems*, pp. 2637–2646, 2017.

Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta numerica*, 8:143–195, 1999.

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

Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in neural information processing systems*, pp. 901–909, 2016.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pp. 2234–2242, 2016.

Stefan Scholtes. Homeomorphism conditions for coherently oriented piecewise affine mappings. *Mathematics of operations research*, 21(4):955–978, 1996.

Viraj Shah and Chinmay Hegde. Solving linear inverse problems using gan priors: An algorithm with provable guarantees. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4609–4613. IEEE, 2018.

Andreas M Tillmann and Marc E Pfetsch. The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. *IEEE Transactions on Information Theory*, 60(2):1248–1259, 2013.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

RO Winder. Partitions of n-space by hyperplanes. *SIAM Journal on Applied Mathematics*, 14(4): 811–818, 1966.

Yan Wu, Mihaela Rosca, and Timothy Lillicrap. Deep compressed sensing. *arXiv preprint arXiv:1905.06723*, 2019.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94: 103–114, 2017.

Globally Injective ReLU Networks: Supplementary Material

Appendix A Proofs from Section 2

A.1 Proofs from Subsection 2.2

Proof of Theorem 1. Suppose that W is such that the conditions of Theorem 1 hold, and that $\mathrm{ReLU}(Wx_1) = \mathrm{ReLU}(Wx_2) = y$. If for $j \in [[m]]$, $y|_j > 0$ then both $\langle w_j, x_1\rangle > 0$ and $\langle w_j, x_2\rangle > 0$. Similarly, if $y|_j \leq 0$ then both $\langle w_j, x_1\rangle \leq 0$ and $\langle w_j, x_2\rangle \leq 0$. In particular, this implies that

$$\langle w_j, x_1\rangle > 0 \iff \langle w_j, x_2\rangle > 0 \text{ and } \langle w_j, x_1\rangle \leq 0 \iff \langle w_j, x_2\rangle \leq 0. \tag{11}$$

If we then consider $x_\alpha := (1-\alpha)x_1 + \alpha x_2$ where $\alpha \in (0,1)$, then

$$\mathrm{ReLU}(Wx_\alpha) = y = \mathrm{ReLU}(Wx_1) = \mathrm{ReLU}(Wx_2). \tag{12}$$

If $\langle w_j, x_\alpha\rangle > 0$ then at least one of $\langle w_j, x_1\rangle > 0$ or $\langle w_j, x_2\rangle > 0$. (11) implies that both must hold, therefore $\langle w_j, x_1\rangle = \langle w_j, x_2\rangle > 0$. If $\langle w_j, x_\alpha\rangle = 0$ then $\langle w_j, x_1\rangle = \langle w_j, x_2\rangle = 0$ (otherwise (11) is violated), thus

$$\mathrm{ReLU}(W|_{S(x_\alpha,W)}x_1) = \mathrm{ReLU}(W|_{S(x_\alpha,W)}x_2) \implies W|_{S(x_\alpha,W)}x_1 = W|_{S(x_\alpha,W)}x_2 \tag{13}$$

and so because $W|_{S(x_\alpha,W)}$ is full rank, this implies that $x_1 = x_2$. This proves one direction.

The other direction follows from the following. Suppose that there exists a x such that $W|_{S(x,W)}$ don't span $\mathbb{R}^n$. If $S(x,W) = \emptyset$ non-injectivity trivially follows, so suppose w.l.o.g. that $S(x,W) \neq \emptyset$. Let $x^\perp \in \ker(W|_{S(x,W)})$ and $\alpha \in \mathbb{R}^+$ such that $\alpha < \min_{j\in S^c(x,W)} \frac{-\langle x, w_j\rangle}{|\langle x^\perp, w_j\rangle|}$[1]. Then for $j = 1,\dots,m$ one of the following two hold

$$\text{if } j \in S(x,W) \text{ then } \left\langle w_j, x + \alpha x^\perp\right\rangle = \langle w_j, x\rangle + \alpha\left\langle w_j, x^\perp\right\rangle = \langle w_j, x\rangle \tag{14}$$

$$\text{if } j \in S^c(x,W) \text{ then } \left\langle w_j, x + \alpha x^\perp\right\rangle = \langle w_j, x\rangle + \alpha\left\langle w_j, x^\perp\right\rangle < 0. \tag{15}$$

Thus, as ReLU acts pointwise (row-wise in W), we have that

$$\mathrm{ReLU}(W(x + \alpha x^\perp)) = \mathrm{ReLU}(Wx) \tag{16}$$

and, hence, $\mathrm{ReLU}(W\cdot)$ is not injective. □

Proof of Lemma 1. First, we show that if $\mathrm{ReLU}(W|_{b\geq 0}\cdot)$ is injective, then so is $\mathrm{ReLU}(W\cdot + b)$. Clearly if $\mathrm{ReLU}(Wx_1+b) = \mathrm{ReLU}(Wx_2+b)$ then $\mathrm{ReLU}(W|_{b\geq 0}x_1+b|_{b\geq 0}) = \mathrm{ReLU}(W|_{b\geq 0}x_2+b|_{b\geq 0})$ as well. If we apply Lemma 8 to each component of the above equation, then we obtain that $\mathrm{ReLU}(W|_{b\geq 0}x_1) = \mathrm{ReLU}(W|_{b\geq 0}x_2)$ which, given the injectivity of $\mathrm{ReLU}(W|_{b\geq 0}\cdot)$, implies that $x_1 = x_2$.

Now suppose that $\mathrm{ReLU}(W|_{b\geq 0})$ is not injective. Let $x \in \mathbb{R}^n$ be such that $W|_{b\geq 0}$ doesn't have a DSS of $\mathbb{R}^n$ w.r.t. x. Let $\beta > 0$ be small enough so that $W|_{b<0}(\beta x) + b|_{b<0} < 0$ component-wise, and let $x^\perp \in \mathbb{R}^n$ such that (as in (16)) $\mathrm{ReLU}(W|_{b\geq 0}(\beta x + x^\perp)) = \mathrm{ReLU}(W|_{b\geq 0}\beta x)$. Further let $\alpha < 1$ be small enough such that $W|_{b<0}(\beta x + \alpha x^\perp) + b|_{b<0} < 0$. By a component-wise analysis, we have that

$$\mathrm{ReLU}(W(\beta x + \alpha x^\perp) + b) = \mathrm{ReLU}(W\beta x + b)|_{b\geq 0} = \mathrm{ReLU}(W\beta x + b); \tag{17}$$

thus, $\mathrm{ReLU}(W\cdot + b)$ is not injective. □

Proof of Corollary 2. If $W \in \mathbb{R}^{m\times n}$ is injective then consider a plane p in $\mathbb{R}^n$ that none of the rows of W lie in. Apply Theorem 1 to both normals of the plane. The corresponding DSS' for each normal are disjoint, thus there must be at least $2n \geq m$, so $m < 2\cdot n$ implies non-injectivity.

Now we show that if W satisfies Theorem 1, then W is of the form given by (5). Suppose that there is a row vector w_i such that there are no row vectors pointing in the $-w_i$ direction. Let p be a plane

[1] If $\left\langle x^\perp, w_j\right\rangle = 0$ for all $j \in S^c(x,W)$, then any $\alpha > 0$ will do.

through the origin such that $w_i \in p$, but $w_{i'} \notin p$ for $i \neq i'$. By Theorem 1, there must be at least n columns that lie on each (closed) half of p. Indeed one of the sides must have exactly n vectors on it (including w_i). By considering a small rotation of p, we can construct a plane with only $n-1$ vectors on one side, hence there is no DSS for that rotated plane's normal. Thus for $\mathrm{ReLU}\, W$ to be injective, for every $i \in [[2n]]$ there must be a different $i' \in [[2n]]$ such that w_i and $w_{i'}$ are anti-parallel. This can only happen if for every $w \in W$ there is a corresponding $-dw \in W$, so W must have the form of (5). □

A.2 Proof of Theorem 2

A.2.1 Upper Bound on Minimal Expansivity

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let $w_i : \Omega \to \mathbb{R}^n$, $1 \leq i \leq m$ be m iid Gaussian vectors on Ω stacked in a matrix $W : \Omega \to \mathbb{R}^{m \times n}$.

We aim to find conditions for a ReLU layer with the matrix W to be injective. Injectivity fails at p the half-space $\{x \,:\, \langle x, p \rangle \geq 0\}$ contains fewer than n vectors w_i. Equivalently, injectivity fails if there is a half-space which contains more than $m-n$ vectors w_i (which implies that its opposite half-space has fewer than n).

Let $k = m - n + 1$ and denote by $A \in \mathbb{R}^{k \times n}$ some $k \times n$ submatrix of W (for example, the first k rows). Our strategy is to first bound the probability that for a fixed subset of k rows of W, there exists an x having positive inner products will all k rows (which signals non-injectivity per above discussion). Second, since there are $\binom{m}{k}$ subsets of k rows, we use the union bound to get an upper bound on the probability of non-injectivity.

For the first part we follow the proof of Bürgisser & Cucker (2013, Theorem 13.6), parts of which we reproduce for the reader's convenience. For a sign pattern $\sigma \in \{-1, 0, 1\}^k$, we denote by $R_A(\sigma)$ the set of all $x \in \mathbb{R}^n$ which produce the sign pattern σ (they belong to the σ-"wedge", possibly empty),

$$R_A(\sigma) = \{x \in \mathbb{R}^n \,:\, \mathrm{sign}(\langle x, a_i \rangle) = \sigma_i,\ i \in \{1, \ldots, k\}\}\,. \tag{18}$$

For $\sigma \in \{-1, +1\}^k = \Sigma$, we define the event $\mathcal{E}_\sigma$ as

$$\mathcal{E}_\sigma = \left\{\omega : R_{A(\omega)}(\sigma) \neq \emptyset\right\}. \tag{19}$$

We are interested in $\sigma_0 = (1, \ldots, 1)$, meaning that all the inner products are positive. Note that the probability of $\mathcal{E}_\sigma$ is the same for all σ due to the symmetry of the Gaussian measure. Further, note that $\sum_{\sigma \in \Sigma} \mathbf{1}_{\mathcal{E}_\sigma}(\omega) = |\left\{\sigma \,:\, R_{A(\omega)}(\sigma) \neq \emptyset\right\}|$ is the number of wedges defined by A. Then

$$\mathbb{P}(\mathcal{E}_{\sigma_0}) = \frac{1}{2^k} \sum_{\sigma \in \Sigma} \mathbb{P}(\mathcal{E}_\sigma) = \frac{1}{2^k} \sum_{\sigma \in \Sigma} \mathbf{E}(\mathbf{1}_{\mathcal{E}_\sigma}) = \frac{1}{2^k} \mathbf{E}\left(\sum_{\sigma \in \Sigma} \mathbf{1}_{\mathcal{E}_\sigma}\right) = \frac{1}{2^{k-1}} \sum_{i=0}^{n-1} \binom{k-1}{i}, \tag{20}$$

by Windner's bound (Winder, 1966). (The hyperplane arrangement is generic almost surely so we use equality.)

We now have the probability that for a subset of k vectors w_i, there exists an $x \in \mathbb{R}^n$ which has positive inner products with all k vectors. We are interested in the following event which implies non-injectivity,

$$\mathcal{E}_{\mathrm{NI}} = \left\{\omega \,:\, W(\omega) \text{ has a subset of } k \text{ rows } B(\omega) \text{ such that } R_{B(\omega)}(\sigma_0) \neq \emptyset\right\}. \tag{21}$$

Conversely, $\omega \in \mathcal{E}_{\mathrm{NI}}^c$ implies almost sure injectivity.

Since there are $\binom{m}{k} = \binom{m}{m-n+1} = \binom{m}{n-1}$ different subsets of k rows, we can bound the probability of $\mathcal{E}_{\mathrm{NI}}$ as

$$\mathbb{P}(\mathcal{E}_{\mathrm{NI}}) \leq \binom{m}{n-1} \mathbb{P}(\mathcal{E}_{\sigma_0}) \leq \left(\frac{me}{n-1}\right)^n 2^{-(m-n)} 2^{nH\left(\frac{n}{m-n-1}\right)} \tag{22}$$

$$\lessapprox (ce)^n 2^{-n(c-1)} 2^{nH((c-1)^{-1})} \tag{23}$$

$$= 2^{-n[-\log_2(ce) + c - 1 - H(\epsilon)]}, \tag{24}$$

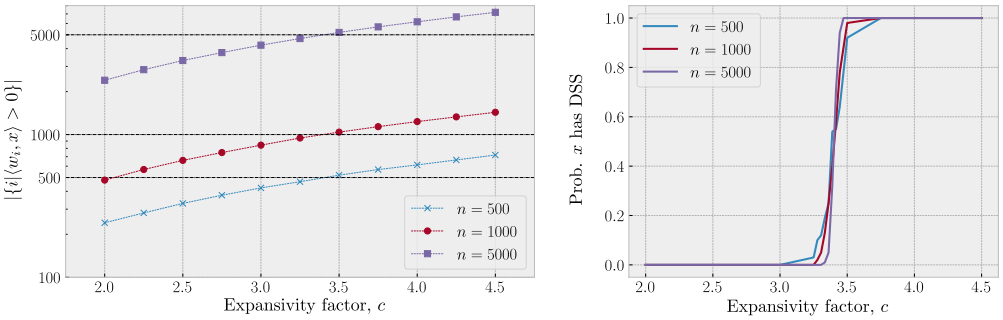


Figure 6: We empirically show that for random Gaussian matrices $\mathbf{W} \in \mathbb{R}^{m\times n}$ $m = cn$, a critical oversampling factor of at least $c > 3.5$ is required for injectivity of ReLU. We choose $x = -\frac{1}{m}\sum_j w_j / \|w_j\|$. *Left:* a plot of the number of elements of W that point in the direction of x as a function of expansivity for several different choices of n. If this quantity is less than n, then W cannot contain a DSS of $\mathbb{R}^n$ w.r.t. x. *Right:* a plot of the empirical probability that a Gaussian matrix W contains a DSS of $\mathbb{R}^n$ w.r.t. x as a function of expansivity for several different values of n.

with $\epsilon = (c-1)^{-1}$. To get the bound note that the bracket in the exponent is positive for $c \geq 5.75$, where $H(\epsilon) = -\epsilon\log_2(\epsilon) - (1-\epsilon)H(1-\epsilon)$ is the binary entropy function and we used a related bound on the sums of binomial coefficients.[2]

A.2.2 Lower Bound on Minimal Expansivity of Layers with Gaussian Weights

In this appendix we prove that large random Gaussian weight matrices $W \in \mathbb{R}^{m\times n}$ yield non-injective ReLU layers with high probability if $m < c^* n$, where $c^* \approx 3.4$. Note that injectivity fails if there exists a half-space in $\mathbb{R}^n$ which contains less than n rows of W. Equivalently, injectivity fails if there is a half-space with more than $m-n$ rows of W, since then the opposite halfspace has less than n. The core idea of the proof is to make an educated guess for a half-space which has many vectors, and compute the probability that it has more than $m-n$. A good such guess is to take the halfspace defined by the average direction of a row of W. Equivalently, we study of the size of $S(\overline{w}, W)$ where $\overline{w} = \frac{1}{m}\sum_{k=1}^m w_k$, the row average of a matrix W. If $|S(\overline{w}, W)| > m-n$, then $|S(-\overline{w}, W)| < n$, and W cannot have a DSS w.r.t. $-\overline{w}$. As the Figure 6 shows, when $m < c^* n$, W does not contain a DSS w.r.t. $-\overline{w}$ with high probability.

Let $W \in \mathbb{R}^{m\times n}$ be a matrix such that the rows of W are i.i.d. random vectors distributed as $\mathcal{N}(0, I_n)$. Define the event E_i as

$$E_i := \left\{\omega \ : \ \left\langle w_i(\omega), \sum_{k=1}^m w_k(\omega)\right\rangle \geq 0\right\}. \tag{25}$$

Lemma A. *Let $m, n \to \infty$ so that $\frac{m}{n} \to c$. Then $\mathbb{P}(E_i) \to \frac{1}{2}\mathrm{erfc}\left(-\frac{1}{\sqrt{2c}}\right)$.*

Proof. We have

$$\mathbb{P}\left(E_i\right) = \mathbb{P}\left(\|w_i\|^2 + \left\langle w_i, \sum_{i\neq j} w_j\right\rangle \geq 0\right) = \mathbb{P}\left(\|w_i\|^2 + \|w_i\| Y \geq 0\right)$$

where $Y \sim \mathcal{N}(0, m-1)$. We now use the fact that $\|w_i\|^2$ concentrates around n. Define the event

$$D_i = \left\{\omega \ : \ \|w_i(\omega)\|^2 \in [(1-\epsilon)n, n/(1-\epsilon)]\right\}. \tag{26}$$

[2] https://en.wikipedia.org/wiki/Binomial_coefficient#Sums_of_binomial_coefficients

One can show using standard concentration arguments that $\mathbb{P}(D_i) \geq 1 - 2\exp(-\epsilon^2 n/4)$, and so by the law of total probability,

$$\mathbb{P}\left\{\|w_i\|^2 + \|w_i\| Y \geq 0\right\} = \mathbb{P}\left\{\|w_i\| + Y \geq 0 \,|\, D_i\right\} \mathbb{P}\{D_i\} + \mathbb{P}\left\{\|w_i\| + Y \geq 0 \,|\, D_i^c\right\} \mathbb{P}\left\{D_i^c\right\}. \tag{27}$$

Choosing $\epsilon = \epsilon(n) = n^{-1/4}$ yields

$$\mathbb{P}\left\{\sqrt{n - n^{3/4}} + Y \geq 0\right\} \leq \mathbb{P}\left\{\|w_i\| + Y \geq 0 \,|\, D_i\right\} \leq \mathbb{P}\left\{\sqrt{n/(1 - n^{-1/4})} + Y \geq 0\right\}.$$

Finally, substituting $Z = \frac{Y}{\sqrt{m-1}} \sim \mathcal{N}(0,1)$ yields

$$\mathbb{P}\left\{Z \geq -\sqrt{\frac{n - n^{3/4}}{m-1}}\right\} \leq \mathbb{P}\left\{\|w_i\| + Y \geq 0 \,|\, D_i\right\} \leq \mathbb{P}\left\{Z \geq -\sqrt{\frac{n/(1 - n^{-1/4})}{m-1}}\right\}$$

Both sandwiching probabilities converge to $\mathbb{P}\left\{Z \geq -\frac{1}{\sqrt{c}}\right\} = \frac{1}{2}\mathrm{erfc}\left(-\frac{1}{\sqrt{2c}}\right)$. Noting that $\mathbb{P}\{D_i\} \to 1$ and $\mathbb{P}\{D_i^c\} \to 0$ we finally have from (27) that

$$\mathbb{P}\{E_i\} \to \frac{1}{2}\mathrm{erfc}\left(-\frac{1}{\sqrt{2c}}\right).$$

□

Lemma B. *Under the same conditions as Lemma A when $i \neq j$,*

$$\mathbb{P}\{E_i \cap E_j\} \to \frac{1}{4}\mathrm{erfc}\left(-\frac{1}{\sqrt{2c}}\right)^2.$$

Proof. The proof is similar to that of Lemma A. In addition to concentration of norm of iid Gaussian vectors, we use the fact that $\langle w_i, w_j \rangle / m$ is of order $n^{-1/2}$ and that if $\mathbb{P}\{D\} \to 1$ as $n \to \infty$, then for a fixed c and some event A, $\mathbb{P}\{A|D\} - \mathbb{P}\{A\} \to 0$. Here D is the event that the various quantities are close to the value they concentrate about, and A asymptotically has the same probability as $E_i \cap E_j$. □

Theorem 6. *Given a Gaussian weight matrix $W \in \mathbb{R}^{m \times n}$, the layer $\mathrm{ReLU}(Wx)$ with $m/n \to c$ is not injective with probability $\to 1$ as $n \to \infty$ when $c < c^*$, where c^* is the unique positive real solution to*

$$\frac{1}{2}\mathrm{erfc}\left(\frac{1}{\sqrt{2c}}\right) = \frac{1}{c}.$$

The numerical value of c^* is ≈ 3.4.

Proof. Let $X_i = \mathbf{1}_{\left\langle w_i, \frac{1}{m}\sum_{j=1}^m w_j \right\rangle \geq 0}$, the indicator function of the event E_i. The expected number of w_i with a positive inner product with $\sum_{k=1}^m w_i$ is by the linearity of expectation equal to

$$\mathbb{E}\left(\sum_{i=1}^m X_i\right) = m \cdot \mathbb{P}\{E_i\} =: mp.$$

By Chebyshev's inequality

$$\mathbb{P}\left(\left|\frac{1}{m}\sum_{i=1}^m X_i - p\right| \geq t\right) \leq \frac{\sigma^2}{t^2}$$

where $\sigma^2 = \mathbb{V}\left(\frac{1}{m}\sum_{i=1}^m X_i\right)$ is the variance of the sum. We compute

$$m^2\sigma^2 = \mathbb{V}\left(\sum_{i=1}^m X_i\right) = \mathbb{E}\left(\sum_{i=1}^m X_i\right)^2 - \left(\mathbb{E}\sum_{i=1}^m X_i\right)^2 = mp + \sum_{i \neq j} \mathbb{P}(E_i \cap E_j) - (mp)^2.$$

Since $\mathbb{P}(E_i \cap E_j) \to p^2$ by Lemmas A and B, we have for any i and $j \neq i$

$$\sigma^2 = \frac{mp + m(m-1)\mathbb{P}(E_i \cap E_j) - (mp)^2}{m^2} \to 0.$$

Thus indeed for any $t > 0$

$$\mathbb{P}\left(\left|\frac{1}{m}\sum_{i=1}^{m} X_i - p\right| \geq t\right) \to 0.$$

Finally,

$$\mathbb{P}(\text{noninjectivity}) \geq \mathbb{P}\left(\sum_{i=1}^{m} X_i > m - n\right)\mathbb{P}\left(\frac{1}{m}\sum_{i=1}^{m} X_i > 1 - \frac{1}{c}\right). \tag{28}$$

Combining with Lemma A we get that

$$\frac{1}{m}\sum_{i=1}^{m} X_i \xrightarrow{\mathbb{P}} \frac{1}{2}\text{erfc}\left(-\frac{1}{\sqrt{2c}}\right)$$

Thus

$$\mathbb{P}(\text{noninjectivity}) \geq \mathbb{P}\left(\frac{1}{m}\sum_{i=1}^{m} X_i > 1 - \frac{1}{c}\right) \to \begin{cases} 0 & c > c^* \\ 1 & c < c^*. \end{cases} \tag{29}$$

□

A.3 Experiments testing the minimal expansivity threshold

Figure 6 suggests that an expansivity factor of 2.1 as suggested by Lei et al. (2019) is not enough to ensure global injectivity with Gaussian weights. This in turn leads to failure cases when inverting the network in applications to inverse problems, as we show in this section. We use a single ReLU layer, $f(x) = \text{ReLU}(Wx)$, and choose x to be an MNIST digit for ease of visualization of the "latent" space. We choose $W \in \mathbb{R}^{cn \times n}$ with iid Gaussian entries and set $c = m/n = 2.1$. Given $y = f(x)$, our goal is to reconstruct x using, for example, gradient descent. This requires at least n of the cn entries of $f(x)$ to be non-zero since $x \in \mathbb{R}^n$. As shown in Figure 6, $x_0 = -\frac{1}{m}\sum_j w_j / \|w_j\|$ violates this condition with high probability when $c = 2.1$.

We choose 2 data samples, x_1 and x_2 to be 2 "latent" codes. Note that W has a DSS with respect to x_1. This ensures that given $f(x_1)$ one can reconstruct x_1. Now, in order to show that injectivity around a few points is insufficient for global injectivity, we linearly interpolate between x_0 and x_1 and x_0 and x_2 in a 2D grid. We show the intermediate xs as images in Figure 7. The xs shown in red cannot be uniquely recovered given $f(x)$, i.e., there exist infinitely many samples close to the red samples that all map to the same output.

To further illustrate this, we also run a simple experiment where we have $y = f(x) + \eta$, with a 10dB signal-to-noise ratio. We invert f to estimate x for 3 different samples in the domain of f. We can easily see that non-injective points of the domain give poorer reconstructions due to the lower number of positive inner products with W. In order to avoid problems of non-convexity in optimization we report the best reconstruction out of 10 in 4 different trials with different random seeds, similar to experiments of Bora et al. (2017). While these results show reconstructions with single layer, the issues are only exacerbated with multiple layers.

A.4 Proof of Theorem 3

We divide the proof into parts for ease of understanding.

Lemma 2 (Inverse Lipschitz Constant: Face Adjacent Wedges)**.** *Let $W \in \mathbb{R}^{m \times n}$ have a DSS w.r.t. every $x \in \mathbb{R}^n$ and $x_0, x_1 \in \mathbb{R}^n$.Define*

$$\forall t \in [0,1], \quad \ell^{x_0,x_1}(t) = (1-t)x_0 + tx_1 \tag{30}$$

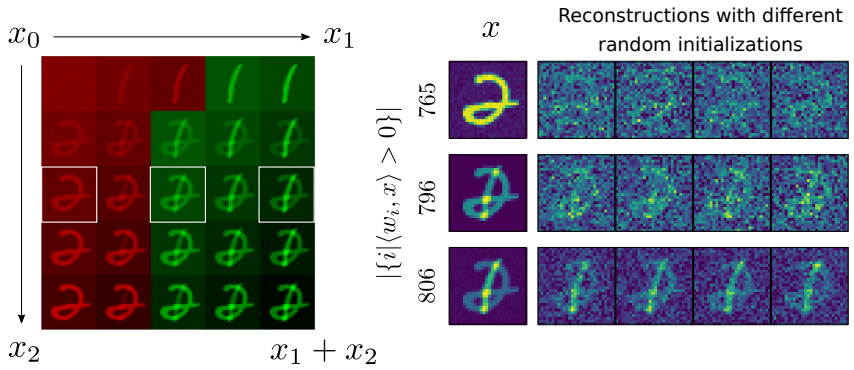
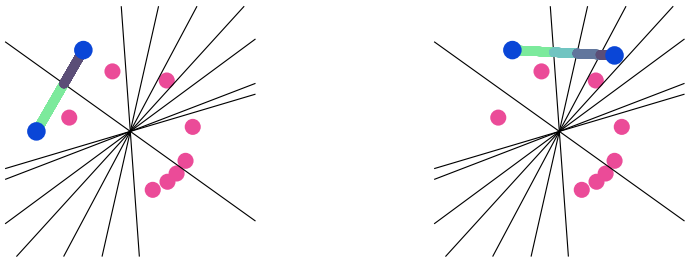


Figure 7: The mapping $\phi(x) := \mathrm{ReLU}(Wx),\ x \in \mathbb{R}^n$ when $W \in \mathbb{R}^{2.1n \times n}$ as suggested in Lei et al. (2019) is not injective as the samples shown in red do not have n positive inner products with the rows of W. This implies that around the red samples ϕ cannot be inverted. For the 3 samples delineated by white boxes on the left, we do a simple denoising test where $y = \phi(x) + \eta$ and η corresponds to 10dB noise. Each reconstruction reported is best out of 10 trials based on MSE error.



(a) (b)

Figure 8: A Proof aid of the DSS condition when $W \in \mathbb{R}^{8 \times 2}$. The blue dots are x_0 and x_1, the pink are the weight matrix rows, black lines denote the boundaries between adjacent wedges, and the multi-colored line is $\ell^{x_0,x_1}(t)$. This line changes color each time it crosses into a new wedge.

and suppose that x_0 and x_1 are such that there is a $t' \in (0,1)$ such that for all $t \in [0,1]$

$$W|_{S(\ell^{x_0,x_1}(t),W)} = \begin{cases} W|_{S(x_0,W)} & \text{if } t < t' \\ W|_{S(x_1,W)} & \text{if } t' < t \end{cases}. \tag{31}$$

Suppose further that x_0, x_1 are such that there is some $\delta > 0$ such that for all $\delta x \in \mathbb{R}^n$, if $\|\delta x\|_2 < \delta$ then there is a $t' + \delta t$ such that

$$W|_{S(\ell^{x_0,x_1+\delta x}(t),W)\geq 0} = \begin{cases} W|_{S(x_0,W)} & \text{if } t < t' + \delta t \\ W|_{S(x_1,W)} & \text{if } t' + \delta t < t \end{cases}. \tag{32}$$

Then

$$\|\mathrm{ReLU}(Wx_0) - \mathrm{ReLU}(Wx_1)\|_2 \geq \frac{1}{\sqrt{2}} \min(\sigma(W|_{S(x_0,W)}), \sigma(W|_{S(x_1,W)})) \|x_0 - x_1\|_2 , \tag{33}$$

where $\sigma(M)$ is the smallest singular value of the matrix M.

Remark 3. The conditions of Lemma 2 on x_0 and x_1 may look very odd, but they have a very natural geometric meaning. The DSS can be thought of as slicing $\mathbb{R}^n$ into wedges by a series of hyperplanes that have the rows of W as normals.

The condition in (31) is interpreted as that the line segment that connects x_0 to x_1 passes from x_0's wedge into x_1's wedge without passing through any wedges in between (see Figure 8a, as opposed to Figure 8b). The implies that x_0 and x_1 must be in wedges that share a boundary. The condition in (32) requires that the wedges of x_0 and x_1 share a face, and not just a corner.

Proof. We denote

$$W|_{S(x_0,W)} = W_0, W|_{S(x_1,W)} = W_1, W_{0\cap 1} = W_0 \cap W_1, W_0 = \begin{bmatrix} W_{0\cap 1} \\ W_{0\setminus 1} \end{bmatrix}, W_1 = \begin{bmatrix} W_{0\cap 1} \\ W_{1\setminus 0} \end{bmatrix}. \tag{34}$$

First we will show that if $w_i, w_{i'} \in W_{0\setminus 1}$, then w_i and $w_{i'}$ must be parallel. From equation (31) and the continuity of $\mathrm{ReLU}(W\ell^{x_0,x_1}(t))$ w.r.t. t, we have that there is a $t' \in (0,1)$ such that

$$\langle w_i, (1-t')x_0 + t'x_1 \rangle = 0 = \langle w_{i'}, (1-t')x_0 + t'x_1 \rangle . \tag{35}$$

If w_i and $w_{i'}$ are not parallel, then let δx be some vector $0 < \|\delta x\| < \delta$ that is perpendicular to w_i but not $w_{i'}$ then

$$\langle w_i, (1-t')x_0 - t'\delta x + t'x_1 \rangle = -t \langle w_i, \delta x \rangle = 0, \tag{36}$$

$$\langle w_{i'}, (1-t')x_0 - t'\delta x + t'x_1 \rangle = -t \langle w_{i'}, \delta x \rangle \neq 0, \tag{37}$$

which contradicts (32). W.l.o.g. the same argument applies to $W_{1\setminus 0}$ and also it is straight forward to see that all elements of $W_{0\setminus 1}$ must be anti-parallel to all elements of $W_{1\setminus 0}$. From $W_{1\setminus 0}$ and $W_{0\setminus 1}$ parallelism, there is a $c \geq 0$ such that for all $x \in \mathbb{R}^n$,

$$\left\|W_{1\setminus 0}x\right\| = c^2 \left\|W_{0\setminus 1}x\right\| . \tag{38}$$

Assume that $c \geq 1$, then

$$\begin{aligned}
\|W_0x_0 - W_1x_1\|_2^2 &= \|W_{0\cap 1}x_0 - W_{0\cap 1}x_1\|_2^2 + \left\|W_{0\setminus 1}x_0\right\|_2^2 + \left\|W_{1\setminus 0}x_1\right\|_2^2 && (39)\\
&= \|W_{0\cap 1}x_0 - W_{0\cap 1}x_1\|_2^2 + \left\|W_{0\setminus 1}x_0\right\|_2^2 + c^2 \left\|W_{0\setminus 1}x_1\right\|_2^2 && (40)\\
&\geq \|W_{0\cap 1}x_0 - W_{0\cap 1}x_1\|_2^2 + \left\|W_{0\setminus 1}x_0\right\|_2^2 + \left\|W_{0\setminus 1}x_1\right\|_2^2 && (41)\\
&\geq \|W_{0\cap 1}x_0 - W_{0\cap 1}x_1\|_2^2 + \frac{1}{2}\left\|W_{0\setminus 1}(x_0 - x_1)\right\|_2^2 && (42)\\
&\geq \frac{1}{2}\|W_0x_0 - W_0x_1\|_2^2 && (43)\\
&\geq \frac{\sigma(W_0)^2}{2}\|x_0 - x_1\|_2^2 . && (44)
\end{aligned}$$

The antepenultimate inequality comes as from the definition of $W_{0\setminus 1}$, we have that $W_{0\setminus 1}x_0$ and $-W_{0\setminus 1}x_1$ are the same sign, thus (98) applies. In the case that $c < 1$, then the rolls of $W_{0\setminus 1}$ and

$W_{1\setminus 0}$ can be switched, and the same result (with $\sigma(W_1)$ in place of $\sigma(W_0)$) is obtained. In either case,

$$\|W_0x_0 - W_1x_1\|_2^2 \geq \frac{1}{2}\min\left(\sigma(W|^2_{S(x_0,W)}), \sigma(W|_{S(x_1,W)})^2\right)\|x_0 - x_1\|_2^2 . \tag{45}$$

□

Lemma 3 (Inverse Lipschitz Constant: Connected through Faces)**.** *Let $W \in \mathbb{R}^{m\times n}$ have a DSS w.r.t. every $x \in \mathbb{R}^n$. Let x_0, x_1 be such that the line connecting passes through n_t wedges, and through their faces (in the sense of Lemma 3). Then*

$$\|\mathrm{ReLU}(Wx_0) - \mathrm{ReLU}(Wx_1)\|_2 \geq \frac{1}{\sqrt{2n_t}}\min_{t\in[0,1]}\sigma(W|_{S(\ell^{x_0,x_1}(t),W)})\|x_0 - x_1\|_2 . \tag{46}$$

Proof. Let $t_1 = 0$, $t_{n_t} = 1$, and let $T = \{t_k\}_{k=1}^{n_t}$ such that $t_k < t_{k+1}$ for $k < n_t$ and $x_k := \ell^{x_0,x_1}(t_k)$ are each in different wedges. Let $c = \frac{1}{\sqrt{2}}\min_{t\in[0,1]}\sigma(W|_{S(\ell^{x_0,x_1}(t),W)})$, then

$$c\|x_0 - x_1\|_2 \leq \sum_{k=1}^{n_t-1} c\left\|x_{t_k} - x_{t_{k+1}}\right\|_2 \tag{47}$$

then by Lemma 2,

$$\sum_{k=1}^{n_t-1} c\left\|x_{t_k} - x_{t_{k+1}}\right\|_2 \leq \sum_{k=1}^{n_t-1}\left\|\mathrm{ReLU}(Wx_{t_k}) - \mathrm{ReLU}(Wx_{t_{k+1}})\right\|_2 \tag{48}$$

and by Lemma 10 we have

$$\sum_{k=1}^{n_t-1}\left\|\mathrm{ReLU}(Wx_{t_k}) - \mathrm{ReLU}(Wx_{t_{k+1}})\right\|_2 \leq \sqrt{n_t}\,\|\mathrm{ReLU}(Wx_0) - \mathrm{ReLU}(Wx_1)\|_2 . \tag{49}$$

Combining (47) - (49) yields

$$\|\mathrm{ReLU}(Wx_0) - \mathrm{ReLU}(Wx_1)\|_2 \geq \frac{1}{\sqrt{2n_t}}\min_{t\in[0,1]}\sigma(W|_{S(\ell^{x_0,x_1}(t),W)})\|x_0 - x_1\|_2 . \tag{50}$$

□

Proof of Theorem 3. Let x_0 and x_1 be given. If the line connecting x_0 and x_1 does not pass through any wedge corners, then we can apply Lemma 3 directly, and get that

$$\|\mathrm{ReLU}(Wx_0) - \mathrm{ReLU}(Wx_1)\|_2 \geq \frac{1}{\sqrt{2n_t}}\min_{t\in[0,1]}\sigma(W|_{S(\ell^{x_0,x_1}(t),W)})\|x_0 - x_1\|_2 . \tag{51}$$

We now argue that the number of wedges that $\ell^{x_0,x_1}(t)$ passes through (and so n_t) is at most m. For each $j \in [[m]]$,

$$\mathrm{ReLU}(W\ell^{x_0,x_1}(t))|_j = \langle w_j, \ell^{x_0,x_1}(t)\rangle \tag{52}$$

is monotone increasing or decreasing. This implies that each $w_j \in W$ can enter or exit the DSS of W w.r.t. $\ell^{x_0,x_1}(t)$ at most once, therefore the total number of unique DSS w.r.t. $\ell^{x_0,x_1}(t)$ (i.e. wedges $\ell^{x_0,x_1}(t)$ pass through) is at most m. Hence, $n_t \leq m$. Clearly

$$\min_{t\in[0,1]}\sigma(W|_{S(\ell^{x_0,x_1}(t),W)}) \geq \min_{x\in\mathbb{R}^n}\sigma(W|_{S(x,W)}), \tag{53}$$

hence we have

$$\|\mathrm{ReLU}(Wx_0) - \mathrm{ReLU}(Wx_1)\|_2 \geq \frac{C}{\sqrt{m}}\|x_0 - x_1\|_2 . \tag{54}$$

Now we show that if Lemma 3 does not apply to two points (namely, the line $\ell^{x_0,x_1}(t)$ passes through a corner in the sense of Remark 3), then the two points can be perturbed an arbitrarily small amount,

so that the perturbed points do satisfy Lemma 3. The corners (again, as in Remark 3) describe the points which are orthogonal to at least two $w_{j_1}, w_{j_2} \in W$. w_{j_1} and w_{j_2} must not be parallel to each other (otherwise Lemma 2 would apply), thus the set of points orthogonal to both w_{j_1} and w_{j_2} constitute a $n-2$ dimensional linear space in $\mathbb{R}^n$.

Let x_0 and x_1 be such that $\ell^{x_0,x_1}(t)$ intersects one of these corners. By considering $\tilde{x}_0 = \delta x + x_0, \tilde{x}_1 = \delta x + x_1$ where δx is perpendicular to $x_1 - x_0$, we can obtained a line $\ell^{\tilde{x}_0,\tilde{x}_1}(t)$ so that $\ell^{\tilde{x}_0,\tilde{x}_1}(t)$ and $\ell^{x_0,x_1}(t)$ do not intersect. The choice of δx is $n-1$ dimensional, thus for every $\delta > 0$ and w_{j_1} and w_{j_2} (that are non-perpendicular) there is a δx so that $\|\delta x\|_2 < \delta$ and $\ell^{\tilde{x}_0,\tilde{x}_1}(t)$ does not intersect the corner of w_{j_1} and w_{j_2}.

Consider a sequence of $\tilde{x}_0^{(i)}, \tilde{x}_1^{(i)}, i = 1, \dots$ such that $\lim_{i\to\infty}(\tilde{x}_0^{(i)}, x_1^{(i)}) = (x_0, x_1)$ and $\ell^{\tilde{x}_0^{(i)},x_1^{(i)}}(t)$ does not pass through a corner for any i. Given that $\|\cdot\|_2$ and $\mathrm{ReLU}(W(\cdot))$ are continuous and so by Lemma 3

$$\left\|\mathrm{ReLU}(W\tilde{x}_0^{(i)}) - \mathrm{ReLU}(W\tilde{x}_1^{(i)})\right\|_2 - \frac{C}{\sqrt{2m}}\left\|\tilde{x}_0^{(i)} - \tilde{x}_1^{(i)}\right\|_2 \geq 0, \tag{55}$$

thus

$$\|\mathrm{ReLU}(Wx_0) - \mathrm{ReLU}(Wx_1)\|_2 - \frac{C}{\sqrt{2m}}\|x_0 - x_1\|_2, \tag{56}$$

$$= \lim_{i\to\infty}\left\|\mathrm{ReLU}(W\tilde{x}_0^{(i)}) - \mathrm{ReLU}(W\tilde{x}_1^{(i)})\right\|_2 - \frac{C}{\sqrt{2m}}\left\|\tilde{x}_0^{(i)} - x_1^{(i)}\right\|_2 \geq 0. \tag{57}$$

and so

$$\|\mathrm{ReLU}(Wx_0) - \mathrm{ReLU}(Wx_1)\|_2 \geq \frac{C}{\sqrt{2m}}\|x_0 - x_1\|_2. \tag{58}$$

□

Remark 4 (Factor of $\frac{1}{\sqrt{m}}$ in Theorem 3.). Throughout this paper we define the discrete norm of $y \in \mathbb{R}^d$ as

$$\|y\|_2 = \left(\sum_{j=1}^{d}[y]_j^2\right)^{\frac{1}{2}}. \tag{59}$$

This is to be contrasted with the norm that arise from the discretization of the L_2 function norm on a finite domain. For example, if we instead thought of y as a discrete sampling of a continuous function $\tilde{y} \in L_2([0,1])$, such that $\forall j = 1, \dots, d$

$$\tilde{y}\left(\frac{j-1}{m}\right) = [y]_j, \tag{60}$$

then we could approximate the $L_2([0,1])$ norm of $\tilde{y}$ by

$$\|\tilde{y}\|_{L_2([0,1])} \approx \|y\|_{l_2([0,1])} := \frac{1}{\sqrt{m}}\left(\sum_{j=1}^{d}[y]_j^2\right)^{\frac{1}{2}}. \tag{61}$$

If we express Theorem 3 in terms of $\|\cdot\|_{l_2([0,1])}$, then it would become

$$\|\mathrm{ReLU}(Wx_0) - \mathrm{ReLU}(Wx_1)\|_{l_2(0,1)} \geq \frac{C(W)}{m}\|x_0 - x_1\|_2. \tag{62}$$

A.5 Theorem 4

Example 1 (Applying Theorem 4, One Channel)**.** Consider a layer of the form $\mathrm{Reshape}(\mathrm{ReLU}(Wx))$ where $W = [C_1^T, \cdots, C_q^T]^T$ and each C_k is a convolution operator with kernel c_k. Suppose further that $W \in \mathbb{R}^{4\times4\times1024\times100} = \mathbb{R}^{16384\times100}$ (as in Radford et al. (2015)). The reshaping operator takes the 16384 single-channel output of W and transforms it into a multi-channel signal. This is necessary

for subsequent convolutions, but plays no role in injectivity. Let $q = 8$, and the 2×2 convolution kernels be given as

$$c_1 = \begin{bmatrix} 3 & -1 \\ -1 & -1 \end{bmatrix}, \quad c_2 = \begin{bmatrix} -1 & 3 \\ -1 & -1 \end{bmatrix}, \quad c_3 = \begin{bmatrix} -1 & -1 \\ 3 & -1 \end{bmatrix}, \quad c_4 = \begin{bmatrix} -1 & -1 \\ -1 & 3 \end{bmatrix} \tag{63}$$

$$c_5 = -c_1, c_6 = -c_2, c_7 = -c_3, c_8 = -c_4. \tag{64}$$

Directly proving that a 16384×100 dimension operator has a DSS w.r.t. every $x \in \mathbb{R}^{100}$ is daunting. However, since each layer of the operator is given by one of only 8 simple convolutions, we can leverage Theorem 4 to significantly simplify the problem. Choosing $P = (2, 2)$, implies that $\mathcal{Z}_{(2,2)}(c_k) = c_k$, and so $W|_{\mathcal{Z}_{(2,2)}} = \bigcup_{k=1}^{8} c_k$. Further, it is easy to see, that $\{c_1, c_2, c_3, c_4\}$ is a basis for $\mathbb{R}^{2\times 2}$, so Corollary 2 applies, $W|_{\mathcal{Z}_P}$ has a DSS for $\mathbb{R}^4$, and by Theorem 4, $\mathrm{ReLU}(Wx)$ is injective.

An example when a layer is injective but P in Theorem 4 must be greater than O is when W is a convolution of 4 kernels of width 3

$$c_1 = [1 \quad 0 \quad -1], \quad c_2 = [1 \quad 0 \quad 1], \quad c_3 = [-1 \quad 0 \quad 1], \quad c_4 = [-1 \quad 0 \quad -1]. \tag{65}$$

If we choose $P = (3)$, then $W|_{\mathcal{Z}_{(3)}} = \bigcup_{k=1}^{4} \{c_k\}$. only has four elements, and so cannot have a DSS w.r.t. every $x\mathbb{R}^3$ by Corollary 2. If we however choose $P = (4)$, then $\mathcal{Z}_{(4)}(c_k) = \{[c_k \quad 0], [0 \quad c_k]\}$ and $W|_{\mathcal{Z}_{(4)}}$ has a DSS of $\mathbb{R}^4$ w.r.t. all $x \in \mathbb{R}^4$ (from Corollary 2), so W has a DSS w.r.t. all $x \in \mathbb{R}^N$.

This example suggests that to apply Theorem 4 to a convolution layer with q kernels of width O, we must choose P so that $W|_{\mathcal{Z}_P}$ has at least the minimal number $2\prod_{j=1}^{p} P_j$ of vectors to have a DSS of a vector space of dimension $|P| = \prod_{j=1}^{p} P_j$. Some algebra gives that $q \geq 2\prod_{j=1}^{p} \frac{1}{1-O_j/P_j}$ and $P_j \geq \frac{O_j}{1-(2/q)^{1/p}}$, where the last inequality holds only when $\frac{P_j}{O_j}$ is independent of j.

Before we can commence with the proof of Theorem 4, we prove the following results.

Lemma 4 (Domain Decomposition)**.** *Suppose that* $\mathbb{R}^n = \mathrm{span}\{\Omega_1, \ldots, \Omega_K\}$[3] *where each* Ω_k *is a subspace and for each* $k = 1, \ldots, K$ *we have a*

$$W_k = \left[w_{k,1}^T, \ldots, w_{k,N_k}^T\right]^T \quad \textit{and} \quad W = \left[W_1^T, \ldots, W_K^T\right] \tag{66}$$

such that $w_{k,\ell} \in \Omega_k$ *and* W_k *has a DSS of* Ω_k *w.r.t. every* $x \in \Omega_k$*. Then* W *has a DSS of* $\mathbb{R}^n$ *w.r.t. every* $x \in \mathbb{R}^n$*.*

Proof. For every $k = 1, \ldots, K$ $W_k|_{S(x,W_k)}$ has a DSS of Ω_k with respect to $P_{\Omega_k}(x)$ where P_{Ω_k} is the orthogonal projection of $\mathbb{R}^n$ onto Ω_k. For every $w_{k,\ell}$,

$$\langle w_{k,\ell}, P_{\Omega_k}(x)\rangle = \langle w_{k,\ell}, x\rangle \tag{67}$$

thus $S(x, W_k) \subset S(x, W)$. From $\Omega_k \subset \mathrm{span}(W_k|_{S(x,W_k)})$ for each k we have a set spanning Ω_k that lie in $W|_{S(x,W)}$, hence

$$\mathbb{R}^n = \mathrm{span}\left(\Omega_1, \ldots, \Omega_K\right) \subset \mathrm{span}\left(\bigcup_{k=1}^{K}\bigcup_{\ell=1}^{N_k}\{w_{k,\ell}\}\right) = \mathrm{span}(W) \tag{68}$$

contains a DSS of $\mathbb{R}^n$ w.r.t. x. The set $\bigcup_{k=1}^{K}\{w_{k,\ell}\}_{\ell=1}^{N_k}$ has no dependence on x, thus it is true for all $x \in \mathbb{R}^n$. □

Lemma 5. *Given a convolution operator* $C \in \mathbb{R}^{N\times N}$*. Let* $0 \leq V$ *be such that* $V + O \leq N$*. For each* $x \in C$*,*

$$\mathrm{aug}_{V:V+O}(x) \in C, \quad \textit{where} \quad (\mathrm{aug}_{V:V+O}(x))_J = \begin{cases} (x)_{J-V} & \textit{if } 1+V \leq J \leq V+O \\ 0 & \textit{otherwise} \end{cases}. \tag{69}$$

Note that $(\mathrm{aug}_{V:V+O}(x))_J$ *restricted to the indices* $1 + V$ *through* $V + P$ *is exactly* x*.*

[3]This is a slight abuse of traditional spanning notation. Here we mean that there is a set of vectors of Ω_1, Ω_2, ... such that their union spans $\mathbb{R}^n$.

Definition 4 (Zero-Padded Kernel)**.** Let c be a convolution kernel of width O, and let P be another multi-index. We define the set of zero-padded kernels[4] of c as the set

$$\mathcal{Z}_P(c) = \left\{ \hat{c} \in \mathbb{R}^P \colon \exists Q, 0 \leq Q \leq P - O \quad \hat{c}_T = \begin{cases} c_{Q-T+O+1} & \text{if } 1 \leq T - Q - \leq O \\ 0 & \text{otherwise} \end{cases} \right\} \tag{70}$$

Note that the above notation implies that $\mathcal{Z}_P(c) = \{\}$ if $O \not\leq P$.

The above is always well defined, as $c_{Q-T+O+1}$ is well defined iff $1 \leq Q - T + O + 1 \leq O$ (recall that the kernel c is of width O), and so

$$1 \leq Q - T + O + 1 \leq O \iff -O \leq Q - T \leq -1 \iff 1 \leq T - Q \leq O \tag{71}$$

Now we proceed with the proof of Theorem 4.

Proof of Theorem 4. The strategy for this proof will be to use Lemma 4 by decomposing $\mathbb{R}^n$ into some number of different domains $\{\Omega_v\}_{v=1}^{n_v}$, each of which are a restriction of $\mathbb{R}^n$ to P non-zero components. For each Ω_v, the elements of W that lie in Ω_v can be identified as the elements in $W|_{\mathcal{Z}_P}$. If for one v the components of $W|_{\mathcal{Z}_P}$ form a DSS of Ω_v w.r.t. every $x \in \Omega_v$, then it has a DSS for every such Ω_k, so we can apply Lemma 4 and get that W has a DSS of $\mathbb{R}^n$ w.r.t. all $x \in \mathbb{R}^n$.

Given an offset $V' \geq 0$ such that $V' + P \leq N$, define $\Omega_{V'}$ as the subspace of all vectors $x \in \mathbb{R}^N$ such that

$$\Omega_{V'} = \left\{ x \in \mathbb{R}^N \colon x_J = 0 \text{ if } 1 + V' \not\leq J \text{ or } J \not\leq V' + P \right\}. \tag{72}$$

From Lemma 5, for any $k = 1, \ldots, n_v$, if $x^{k,P} \in \mathcal{Z}_P(c_k)$, C_k is a submatrix of W and

$$\text{aug}_{V':V'+P}(x^{k,P}) \in C_k. \tag{73}$$

Further, for any such $x^{k,P}$,

$$\text{aug}_{V':V'+P}(x^{k,P}) \in \Omega_{V'}. \tag{74}$$

If $W|_{\mathcal{Z}_P}$ contains a DSS for $\mathbb{R}^P$ w.r.t. all $x \in \mathbb{R}^P$, then

$$\text{aug}_{V':V'+P}(W|_{\mathcal{Z}_P}) \text{ contains a DSS for } \Omega_{V'} \text{ w.r.t. all } x \in \Omega_{V'}. \tag{75}$$

This follows from Lemma 4. From Lemma 5 for any V' such that

$$\text{aug}_{V':V'+P}(W|_{\mathcal{Z}_P}) \in W, \tag{76}$$

if $W|_{\mathcal{Z}_P}$ has a DSS of $\mathbb{R}^P$ w.r.t. all $x \in \mathbb{R}^P$, then W contains a DSS of $\Omega_{V'}$ for all $0 \leq V' \leq N - P$. Finally, note that $\text{span}(\{\Omega_{V'}\}_{V'=0}^{N-P})$, and so using (75), (76) we can apply Lemma 4 and find that W contains a DSS of $\mathbb{R}^N$ w.r.t. all $x \in \mathbb{R}^N$. □

For a multi-channel input (with n_c channels) $x \in \underbrace{\mathbb{R}^N \times \mathbb{R}^N \times \cdots \times \mathbb{R}^N}_{n_c \text{ times}}$, a multi-channel convolution C on x is given by $Cx = \sum_{q=1}^{n_c} C_q x_q$ where C_o is a convolution on $\mathbb{R}^N$ (defined by Definition 2) and $x_o \in \mathbb{R}^N$ is the restriction of x to the o'th channel. Because of the additive structure of multi-channel convolutions a n_c over $\mathbb{R}^N = \mathbb{R}^{N_1} \times \cdots \times \mathbb{R}^{N_p}$ dimensional domain of width $O = (O_1, \ldots, O_p)$ with kernels $c_1, \ldots, c_{n_c}$ is equivalent to a single convolution of width $(O, n_c) = (O_1, \ldots, O_p, n_c)$ over $\mathbb{R}^{(N,p)} = \mathbb{R}^{N_1} \times \cdots \times \mathbb{R}^{N_p} \times \mathbb{R}^{n_c}$. This follows from

$$(Cx)_J = \sum_{q=1}^{n_c} (C_q x_q)_J = \sum_{q=1}^{n_c} \sum_{I=1}^{O} (c_q)_{O-I-1} (x_q)_{J+I} = \sum_{(I,n_c)=1}^{(O,q)} c_{(O,n_c)-(I,q)-1} x_{(J,q)+(I,q)}.$$

[4]With many convolutional neural networks, padding refers to the act of padding the image (with e.g. zeroes), but the convolutional kernels are not padded. For our results, the variable P refers to the padding of the kernels, not the image.

APPENDIX B ROBUSTNESS TO BATCH, WEIGHT, AND SPECTRAL NORMALIZATION

Normalization strategies such as batch (Ioffe & Szegedy, 2015), layer (Ba et al., 2016), instance (Ulyanov et al., 2016), group (Wu & He, 2018), weight (Salimans & Kingma, 2016) and spectral (Miyato et al., 2018) normalization promote convergence during training and encourage low generalization error. Normalization is a many-to-one operation. In this section we show that batch, weight, and spectral normalization do not interfere with injectivity provided, of course, that the network is injective without normalization. We ask if injectivity is compatible with normalization in a *trained* network.

Let $\{x_i\}_{i=1,\dots,m}$ represent the inputs to a given layer over a mini-batch. The batch normalization adds two learnable parameters (γ, β) and transforms x_i to y_i as

$$\mu_{\mathcal{B}} = \frac{1}{m}\sum_{i=1}^{m} x_i, \qquad \sigma_{\mathcal{B}}^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2, \qquad \hat{x}_i = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}, \qquad y_i = \gamma \hat{x}_i + \beta. \tag{77}$$

Although for a given γ, β the relationship between x_i and y_i is not injective , batch normalization is usually present during training but not at test time. Provided that the learned weights satisfy Theorem 1 and Lemma 1, batch normalization does not spoil injectivity.

In batch renormalization (Ioffe, 2017) it is still desirable to whiten the input into each layer. During run time there may be no mini-batch, so running averages of the σ_i and μ_i (denoted $\hat{\sigma}, \hat{\mu}$) computed during training are used. Batch renormalization at test time is then $\hat{x} = \frac{x-\hat{\mu}}{\sqrt{\hat{\sigma}^2+\epsilon}}$, $y = \gamma\hat{x} + \beta$. Since, importantly, $\hat{\sigma}$ and $\hat{\mu}$ are not functions of x, the mapping between x and y is one-to-one (when $\gamma \neq 0$), thus such normalization in an injective network does not spoil the injectivity.

In weight normalization (Salimans & Kingma, 2016) the coefficients of the weight matrices are normalized to have a given magnitude. This normalization is not a function of the input or output signals, but rather of the weight matrices themselves. This plays no role in injectivity.

B.1 LAYER, INSTANCE AND GROUP NORMALIZATION

layer, instance and group normalization, unlike batch normalization, take place during both training and execution and, unlike weight and spectral normalization, the normalization is done on the input/outputs of layers instead of on the weight matrices. For these normalizations (2) is modified so that it becomes

$$N(z) = \phi_L(W_L M_L(\cdots \phi_2(W_2 M_2(\phi_1(W_1 z + b_1)) + b_2) \cdots + b_L)) \tag{78}$$

where $M_\ell \colon \mathbb{R}^{n_{i+1}} \to \mathbb{R}^{n_{i+1}}$ are normalization functions that are many-to-one. In general $\phi_\ell(W_\ell M_\ell(\cdot) + b_\ell)$ will not be injective for any $\phi_\ell, W_\ell, b_\ell$ on account of M_ℓ, but for all of the mentioned normalization techniques we can get near injectivity. Before we descend into the particular we make the following observation about normalization methods that obey a certain structure.

Definition 5 (Scalar-Augmented Injective Normalization)**.** Let $M_\ell(x) \colon \mathbb{R}^n \to \mathbb{R}^n$ be a normalization function that is understood to be many-to-one. We say that $M_\ell(x)$ is scalar-augmented injective if there exists a function $m_\ell(x) \colon \mathbb{R}^n \to \mathbb{R}^k$ where $k \ll n$ and $\tilde{M}_\ell \colon \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^n$ such that

$$M_\ell(x) \coloneqq \tilde{M}_\ell(x; m_\ell(x)) \tag{79}$$

and $\tilde{M}_\ell(x; m_\ell(x))$ is injective on x given $m_\ell(x)$.

An example of a normalization function that is scalar-augmented injective is

$$M_\ell(x) = \frac{x}{\|x\|_2}. \tag{80}$$

For this choice of M_ℓ, $k = 1$, and

$$\tilde{M}_\ell(x; c) = \frac{x}{c} \quad m_\ell(x) = \|x\|_2 . \tag{81}$$

With this definition, we can prove the following trivial but useful result

Lemma 6 (Restricted Injectivity of Scalar-Augmented Normalized Networks)**.** *Let N be a deep network of the form in (78) and let each $\phi_\ell(W_\ell \cdot)$ be layer-wise injective. Let the normalization functions $\{M_\ell\}_{\ell=1,\dots,L}$ each be scalar-augmented injective. Then given $\{m_\ell(x)\}_{\ell=1,\dots,L}$, the network*

$$\tilde{N}(z; m_1, \dots, m_\ell) = \phi_L(W_L \tilde{M}_L(\cdots \phi_2(W_2 \tilde{M}_2(\phi_1(W_1 z + b_1); m_2) + b_2) \cdots + b_L; m_L)) \quad (82)$$

is injective.

Proof. The proof of Lemma 6 follows from a straightforward application of induction, combined with Definition 5. □

Remark 5. Note that Lemma 6 implies that for a fixed $\{m_\ell(x)\}_{\ell=1,\dots,L}$, there is at most one value of z such that

$$\tilde{N}(z; m_1, \dots, m_\ell) = N(z), \quad (83)$$

where $N(z)$ is given by (78), and that the z's on both sides of (83) are the same. It is still entirely possible that there are is another choice of z', $\{m_\ell(x)\}_{\ell=1,\dots,L}$ such that

$$\tilde{N}(z; m_1, \dots, m_\ell) = \tilde{N}(z'; m'_1, \dots, m'_\ell). \quad (84)$$

An example of this would be if M_ℓ is of the form in (80), then

$$\tilde{M}_\ell(x; m_\ell(x)) = \tilde{M}_\ell(2x; 2m_\ell(x)). \quad (85)$$

In other words, Lemma 6 implies that the deep network is injective (in z) for a fixed $\{m_\ell(x)\}_{\ell=1,\dots,L}$, but it may still not be injective for all z and $\{m_\ell(x)\}_{\ell=1,\dots,L}$.

With Lemma 6 in tow, we can show that layer, instance, and group normalization are all scalar-augmented injective normalizations, so Lemma 6 applies and yields a kind of injectivity. Layer, instance and group normalization are all related insofar as they can all be expressed in the same abstract form. For a given input x, all three break x up into K parts denoted $\{x|_{S_k}\}_{k=1,\dots,K}$ such that for each $k = 1, \dots, K$

$$\mu_k = \frac{1}{m} \sum_{i=1}^{m} (x|_{S_k})_i \quad \sigma_k^2 = \frac{1}{m} \sum_{i=1}^{m} ((x|_{S_k})_i - \mu_k)^2 \quad (86)$$

$$(\hat{x}|_{S_k})_i = \frac{(x|_{S_k})_i - \mu_k}{\sqrt{\sigma_k^2 + \epsilon}} \quad M(x)|_{S_i} = \gamma_k (\hat{x}|_{S_k})_i + \beta_k. \quad (87)$$

The differences between the three normalization are how $\{S_k\}_{k=1,\dots,K}$ is chosen. For layer normalization $K = 1$ and the normalization is applied to the entire input signal. For instance normalization, there is one S_k for each channel, and the $x|_{S_k}$ restricts x to just one channel of inputs, that is the normalization is done channel-wise. Group normalization is part way between these two, where k is less than the number of channels, and channels are batched together.

In any case, for any of these normalization methods, they are all scalar-augmented injective normalization where

$$M_\ell(x) = \tilde{M}_\ell(x; \{\sigma_{k,\ell}, \mu_{k,\ell}\}_{k=1,\dots,K}). \quad (88)$$

Thus, by Lemma 6 their corresponding deep networks are all injective, provided that for each ℓ, $\{\sigma_{k,\ell}, \mu_{k,\ell}\}_{k=1,\dots,K}$ is saved.

B.2 Pooling Operations

Although pooling may have a different aim than typical normalization, we consider it in this section, as it is mathematically similar to (78). Pooling is similar to layer, instance and group normalization in the sense that they partition the input space into K disjoint pieces, and then output a weighted average upon each piece. Specifically, if $M_p(x) \colon \mathbb{R}^n \to \mathbb{R}^K$ where for $k = 1, \dots, K$,

$$M_p(x)|_{S_k} = \|x|_{S_k}\|_p \quad (89)$$

where $\|\cdot\|_p$ is the discrete p norm of x restricted to the set S_k. For $p = 1$ this is the mean of the absolute value, for $p = 2$ this is the Euclidean mean and for $p = \infty$ it is the maximum of the absolute value. The injectivity of this operation in the cases where $p = 1, 2, \infty$ is considered in the work Bruna et al. (2013).

APPENDIX C PROOFS FROM SECTION 3

C.1 PROOF OF THEOREM 5

To prove Theorem 5 we combine the approximation results for neural networks and the low regularity version of the generic orthogonal projector technique used to prove the easy version of the Whitney's embedding theorem (Hirsch, 2012, Chapter 2, Theorem 3.5) that shows that a C^2-smooth manifold of dimension n can be embedded in $\mathbb{R}^{2n+1}$ with an injective, C^2-smooth map. To prove the result, we first approximate $f(x)$ by a ReLU-type neural network that is only Lipschitz-smooth, so the graph of the map F_θ is only a Lipschitz-manifold. We note that limited regularity often causes significant difficulties for embedding results, as for example for the Lipschitz-smooth manifolds it is presently known only that a n-dimensional manifold can be embedded (without preserving distances) in the Euclidean space $\mathbb{R}^N$ of dimension $N = (n+1)^2$, and the classical Whitney problem, whether a n-dimensional manifold can be embedded in $\mathbb{R}^{2n+1}$, is still open Luukkainen & Väisälä (1977); Cobzaş et al. (2019). Due to this lack of smoothness, we recall the details how this generic projector technique works.

Proof of Theorem 5. Let $\varepsilon > 0$ and $\mathcal{Z} \subset \mathbb{R}^n$ be a compact set. As a continuous function $f : \mathbb{R}^n \to \mathbb{R}^n$ can be uniformly approximated in a compact set by a C^∞-smooth function (see e.g. (Adams, 1975, Thm. 2.29)), we can without loss of generality assume that f smooth and therefore a locally Lipschitz function.

By classical results of approximation theory for shallow neural networks, see Hornik (1991); Leshno et al. (1993); Pinkus (1999), for any $L \geq 1$ there are $\boldsymbol{m}$ and a neural network $F_\theta \in \mathcal{NN}(n, m, L, \boldsymbol{m})$ such that

$$|f(x) - F_\theta(x)| \leq \frac{1}{2}\varepsilon, \quad \text{for all } x \in \mathcal{Z}. \tag{90}$$

We note that by using recent results for deep neural networks, e.g. by Yarotsky (2017), one can obtain efficient estimates on how a given accuracy ε can be obtained using sufficiently large L and $\boldsymbol{m}$. Our aim is the perturb $F_\theta : \mathbb{R}^n \to \mathbb{R}^m$ so that it becomes injective.

We assume that $\mathcal{Z} \subset B^n(0, r_1)$, where $B^n(0, r_1) \subset \mathbb{R}^n$ is an open ball having centre 0 and radius $r_1 > 0$. We denote the closure of this ball by $\overline{B}^n(0, r_1)$.

Let $D = m + n$, $\alpha > 0$, and define a map $H_\theta : \mathbb{R}^n \to \mathbb{R}^{m+n}$,

$$H_\theta(x) = (\alpha x, F_\theta(x)) \in \mathbb{R}^n \times \mathbb{R}^m = \mathbb{R}^D. \tag{91}$$

Observe that the map $H_\theta : \mathbb{R}^n \to \mathbb{R}^D$ is injective.

Let $\mathcal{V}(k, D)$ denote the set of k-tuples $(v_1, v_2, \ldots, v_k)$ where v_j are orthonormal vectors in $\mathbb{R}^D$. Such vectors span a k-dimensional linear space. Furthermore, let $G(k, D)$ denote the set of k-dimensional linear subspaces of $\mathbb{R}^D$, and for $V \in G(k, D)$, $V = \text{span}(v_1, v_2, \ldots, v_k)$, let $P_V = P_{(v_1, v_2, \ldots, v_k)} : \mathbb{R}^D \to \mathbb{R}^D$ be an orthogonal projection which image is the space V. As the dimension of the orthogonal group $O(k)$ is $k(k-1)/2$ and by Milnor & Stasheff (1974), the set $G(k, D)$, called the Grassmannian, is a smooth algebraic variety, of dimension $k(D-k)$ and the dimension of $\mathcal{V}(k, D)$ is $k(D-k) + k(k-1)/2 = k(2D-k-1)/2$.

To prove Theorem 5, we need the following lemma.

Lemma 7. *Let $H_\theta \in \mathcal{NN}(n, D)$, $D > d \geq 2n+1$ be a neural network such that $H_\theta : \mathbb{R}^n \to \mathbb{R}^D$ is injective. Let $X_\theta = \{V \in G(d, D) : \ P_V \circ H_\theta : \mathbb{R}^n \to \mathbb{R}^D$ is injective$\}$. Then the set X_θ is an intersection of countably many open and dense subsets of $G(d, D)$, that is, elements of X_θ are generic. Moreover, the $d(D-d)$ dimensional Hausdorff measure of the complement of X_θ in $G(d, D)$ is zero.*

Note that for $V \in X_\theta$, we have $P_V \circ H_\theta \in \mathcal{NN}(n, D)$.

Proof. We use that fact that $H_\theta : \mathbb{R}^n \to \mathbb{R}^D$ is injective and locally Lipschitz-smooth. Recall that $f : \mathbb{R}^n \to \mathbb{R}^m$ is locally Lipschitz if for any compact set $\mathcal{Z} \subset \mathbb{R}^n$ there is $L_\mathcal{Z} > 0$ such that $|f(x) - f(y)| \leq L_\mathcal{Z}|x - y|$ for all $x, y \in \mathcal{Z}$.

Let $v \in \mathbb{R}^m$ be a unit vector and let $Q_v : \mathbb{R}^D \to \mathbb{R}^D$ be the projection $Q_v(z) = z - (z \cdot v)v$. Let

$$A = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n : \ x \neq y\}.$$

As $H_\theta : \mathbb{R}^n \to \mathbb{R}^D$ is an injection, we can define the map

$$s_\theta : A \to \mathbb{S}^{D-1} = \{w \in \mathbb{R}^D : \ |v| = 1\}, \quad s_\theta(x, y) = \frac{H_\theta(x) - H_\theta(y)}{|H_\theta(x) - H_\theta(y)|}. \tag{92}$$

As observed in the proof of the Whitney's embedding theorem (Hirsch, 2012, Chapter 2, Theorem 3.5), the map $Q_w \circ H_\theta : \mathbb{R}^n \to \mathbb{R}^D$ is an injection when $w \in \mathbb{S}^{D-1}$. To see this, assume that $w \in \mathbb{S}^{D-1}$ satisfies $w \notin s_\theta(A)$, that is, w is not in the image of s_θ. Then, if there are $x, y \in \mathbb{R}^n$, $x \neq y$ such that $Q_w H_\theta(x) = Q_w H_\theta(y)$, we see that there is $t \in \mathbb{R}$ such that$H_\theta(x) - H_\theta(y) = tw$. As w is a unit vector, this yields that $t = |H_\theta(x) - H_\theta(y)| \neq 0$ and

$$w = (H_\theta(x) - H_\theta(y))/t = s_\theta(x, y),$$

which is in contradiction with the assumption that $w \notin s_\theta(A)$. Hence, $w \notin s_\theta(A)$ yields that $Q_w \circ H_\theta : \mathbb{R}^n \to \mathbb{R}^D$ is an injection.

We consider the image $s_\theta(A) \subset \mathbb{S}^{D-1}$. For $h > 0$, let

$$A_h = \{(x, y) \in \overline{B}^n(0, h^{-1}) \times \overline{B}^n(0, h^{-1}) : \ |H_\theta(x) - H_\theta(y)| \geq h\}.$$

As $H_\theta : \overline{B}^n(0, h^{-1}) \to \mathbb{R}^m$ is Lipschitz-smooth with some Lipschitz constant L_h, we see that the map $s_\theta : A_h \to \mathbb{S}^{D-1}$ is Lipschitz-smooth. Since the set A has the Hausdorff dimension $2n$ and the map $s_\theta : A_h \to \mathbb{S}^{D-1}$ is Lipschitz-smooth, the Hausdorff dimension of the set $s_\theta(A_h)$ is at most $2n$, see e.g. Morgan (2016, p. 26). The set A is the union of all sets A_{h_j}, where $h_j = 1/j$ and $j \in \mathbb{Z}$. By Mattila (1995, p. 59) the Hausdorff dimension of a countable union of sets S_j is the supremum of the Hausdorff dimension of the sets S_j. Hence $s_\theta(A) = \bigcup_{j=1}^\infty s_\theta(A_{h_j})$ has the Hausdorff dimension less or equal $2n$.

Since the dimension $D - 1$ of $\mathbb{S}^{D-1}$ is strictly larger than $2n$, we see that the set $s_\theta(A_{h_j})$ is closed, its complement is an open and dense set, and thus the set $Y_1(\theta) := \mathbb{S}^{D-1} \setminus s_\theta(A)$ is an intersection of countably many open and dense sets.

Observe that as Q_{w_1} is a linear map, the map $Q_{w_1} \circ H_\theta$ is also a neural network that belongs in $\mathcal{NN}(n, D)$, and we can denote $Q_{w_1} \circ H_\theta = H_{\theta_1}$ with some parameters θ_1. Thus we can repeat the above arguments using the map $Q_{w_1} \circ H_\theta : \mathbb{R}^n \to \text{span}(w_1)^\perp \equiv \mathbb{R}^{D-1}$ instead of $H_\theta : \mathbb{R}^n \to \mathbb{R}^D$. Repeating the above arguments $D - d$ times, can choose orthonormal vectors $w_j \in \mathbb{S}^{D-1}$, $j = 1, 2, \ldots, D - d$, and sets $Y_j(\theta, w_1, \ldots, w_{j-1}) \subset \mathbb{S}^{D-1} \cap (w_1, \ldots, w_{j-1})^\perp$, which $D - j$ dimensional Hausdorff measures vanish and which complements are intersections of countably many open and dense sets. Let B_θ to be the set of all n-tuples $(w_1, \ldots, w_{D-d})$ where $w_1 \in Y_1(\theta)$ and $w_j \in Y_2(\theta, w_1, \ldots, w_{j-1})$ for all $j = 2, \ldots, D - d$. Note that all such vectors w_j, $j = 1, 2, \ldots, D - d$ are orthogonal vectors spanning a $D - d$ dimensional vector space V, and the map

$$P_V \circ H_\theta : \mathbb{R}^n \to \mathbb{R}^D, \quad \text{where } P_V = Q_{w_{D-d}} \circ \cdots \circ Q_{w_2} \circ Q_{w_1} \tag{93}$$

is injective. By the above, construction the $(D - (d + 1)/2)d$ dimensional Hausdorff measure of the complement of B_θ in $\mathcal{V}(D, d)$ is zero and B_θ is generic set. As the dimension of the set of the orthogonal basis in a d-dimensional vector space is $(d - 1)d/2$, we obtain the claim. □

Now we continue with the proof of Theorem 5. Let $V_0 = \{(0, 0, \ldots, 0)\} \times \mathbb{R}^m \subset \mathbb{R}^D$. By applying Lemma 7 with $d = m$ we see that any neighborhood of V_0 in $G(d, D)$ contain a m-dimensional vector space $V \in X_\theta$. Then $P_V \circ H_\theta$ is injective. We can choose $V \in X_\theta$ to be so close to V_0 that there is a rotation $R_V \in O(D)$ of the space $\mathbb{R}^D$, that maps the subspace V to the subspace V_0, such that

$$\|R_V - \text{Id}\|_{\mathbb{R}^d \to \mathbb{R}^d} < \frac{1}{2(1 + \alpha + \|F_\theta\|_{C(\mathcal{Z})})} \varepsilon. \tag{94}$$

where Id is the identity function. Let $\pi_0 : \mathbb{R}^D \to \mathbb{R}^m$ be the map $\pi_0(y', y'') = y''$ be the projection to the last m coordinates. Then

$$\|R_V \circ QP_V \circ H_\theta - H_\theta\|_{C(\mathcal{Z})} < \frac{1}{2}\varepsilon, \qquad \pi_0 \circ H_\theta = F_\theta \tag{95}$$

This and (90) imply that the claim of Theorem 5 holds for the injective neural network $N_\theta = \pi_0 \circ R_V \circ P_V \circ H_\theta : \mathbb{R}^n \to \mathbb{R}^m$. □

Lemma 7 used in the above proof yields also Corollary 3.

Proof of Corollary 3. Observe that a measure μ that is absolutely continuous with respect to the Lebesgue $\prod_{j=1}^{k} \mathbb{R}^{d_{2j} \times d_{2j-1}}$ is absolutely continuous also with respect to the normalized Gaussian distribution, and that if the set $S = \{F_k \text{ is not injective}\}$ has measure zero with respect to the normalized Gaussian distribution, then its μ-measure is also zero. Thus we can assume without loss of generality that the elements of matrices B_j are independent and have normalized Gaussian distributions.

Assume next that we have shown that $F_{j-1} : \mathbb{R}^n \to \mathbb{R}^{d_{2j-2}}$ is injective almost surely. Then, $f_\theta^{(j)} \circ F_{j-1} : \mathbb{R}^n \to \mathbb{R}^{d_{2j-1}}$ is injective almost surely. If $d_{2j} > d_{2j-1}$, the matrix B_j is almost surely injective and so is $F_j = B_j \circ f_\theta^{(j)} \circ F_{j-1} : \mathbb{R}^n \to \mathbb{R}^{d_{2j}}$. Thus, it is enough to consider the case when $d_{2j-1} \geq d_{2j} \geq 2n+1$. Then the matrix $B_j : \mathbb{R}^{d_{2j-1}} \to \mathbb{R}^{d_{2j}}$ has almost surely rank d_{2j}. By using the singular value decomposition, we can write $B_j = R_j^1 D_j R_j^2$ where $R_j^1 \in O(d_{2j})$, $R_j^2 \in O(d_{2j-1})$, $D_j \in \mathbb{R}^{d_{2j} \times d_{2j-1}}$ is matrix which principal diagonal elements are almost surely strictly positive and the other elements are zeros. Let $V_j = (R_j^2)^{-1}(\mathbb{R}^{d_{2j}} \times \{(0,0,\ldots,0)\}) \subset \mathbb{R}^{d_{2j-1}}$. Let P_{V_j} be an orthogonal projector in $\mathbb{R}^{d_{2j-1}}$ onto the space V_j of dimension d_{2j}. As the distribution of the matrix B_j is invariant in rotations of the space, so is the distribution of linear space V_j in $G(d_{2j-1}, d_{2j})$. By Lemma 7, we see that the map $P_{V_j} \circ f_\theta^{(j)} \circ F_{j-1} : \mathbb{R}^n \to \mathbb{R}^{d_{2j-1}}$ is injective almost surely. This implies that the map $B_j \circ f_\theta^{(j)} \circ F_{j-1} : \mathbb{R}^n \to \mathbb{R}^{d_{2j}}$ is injective almost surely, that is, the map $F_j : \mathbb{R}^n \to \mathbb{R}^{d_{2j}}$ is injective almost surely. The claim follows by induction. $\square$

In Baraniuk & Wakin (2009), see also Hegde et al. (2008); Iwen & Maggioni (2013), Broomhead & Kirby (2001; 2000), the authors study manifold learning using random projectors. These results are related to the proof of Theorem 5 above. Let H_θ be given by (91), a ReLU-based neural network whose graph $M \subset \mathbb{R}^d$, $d = 2n+m$. When P_V is a random projector in $\mathbb{R}^d$ onto a m-dimensional linear subspace V, the injectivity of the neural network $P_V \circ H_\theta$ is closely related to the property that $P_V(M)$ is an n-dimensional submanifold with a large probability. In Broomhead & Kirby (2001; 2000) the authors use Whitney embedding results for C^2-smooth manifold for dimension reduction of data. Our proof applies similar techniques for Lipschitz-smooth maps. In Baraniuk & Wakin (2009); Hegde et al. (2008); Iwen & Maggioni (2013), the authors apply the result that when $M \subset \mathbb{R}^D$ is a submanifold and D is large enough, a random m-dimensional projector P_V satisfies on M the restricted isometry property with a large probability. In this case, $P_V \circ H_\theta$ is not only an injection but its inverse map is also a local Lipschitz map. In this sense, the techniques in Baraniuk & Wakin (2009); Hegde et al. (2008); Iwen & Maggioni (2013) would give improved results to the generic projection technique used in this paper. The results in Baraniuk & Wakin (2009); Hegde et al. (2008); Iwen & Maggioni (2013), however, require that the dimension m of image space of the map $f : \mathbb{R}^n \to \mathbb{R}^m$ satisfies $m \geq C \log(\varepsilon^{-1})$, where ε is the precision parameter in the inequality (10). Our result, Theorem 5 requires only that $m \geq 2n+1$.

APPENDIX D MISCELLANEOUS LEMMAS

Lemma 8. *Let $x, y, z \in \mathbb{R}$ and $z \geq 0$. Then*

$$\mathrm{ReLU}(x+z) = \mathrm{ReLU}(y+z) \implies \mathrm{ReLU}(x) = \mathrm{ReLU}(y). \tag{96}$$

Lemma 9 (Useful Inequalities)**.** *The following geometric inequalities are useful and have straight forward proofs. Suppose that $a, b, c \in \mathbb{R}^n$, then*

1. *For $i = 1, \ldots, k$, let $a_i \in \mathbb{R}^m$. If*

$$\sum_{i=1}^{k} \|a_i\|_2^2 \leq \left\| \sum_{i=1}^{k} a_i \right\|_2^2, \tag{97}$$

then if for each $j = 1, \ldots, m$, $a_i|_j \cdot a_{i'}|_j \geq 0$ for each pair $i, i' \in [[m]]$, then

$$\sum_{i=1}^{k} \|a_i\|_2 \leq \sqrt{k} \left\| \sum_{i=1}^{k} a_i \right\|_2. \tag{98}$$

2. *If* $a, b \in \mathbb{R}^n$ *and* $\langle a, b\rangle \geq 0$*, then*

$$\|a\|_2^2 + \|b\|_2^2 \leq \|a+b\|_2^2 . \tag{99}$$

Lemma 10 (Co-linear Additivity of $\mathrm{ReLU}(W\cdot)$)**.** *Let* $W \in \mathbb{R}^{m\times n}$*,* $x_1, x_2 \in \mathbb{R}^n$*,* $\ell^{x_1,x_2}(t) = (1-t)x_1 + tx_2$*. Let there be*

$$0 = t_1 \leq t_2 \leq \cdots \leq t_{n_t-1} \leq t_{n_t} = 1 \tag{100}$$

then

$$\sum_{k=1}^{n_t-1} \left\|\mathrm{ReLU}(Wx_{t_k}) - \mathrm{ReLU}(Wx_{t_{k+1}})\right\|_2^2 \leq \|\mathrm{ReLU}(Wx_1) - \mathrm{ReLU}(Wx_2)\|_2^2 . \tag{101}$$

and

$$\sum_{k=1}^{n_t-1} \left\|\mathrm{ReLU}(Wx_{t_k}) - \mathrm{ReLU}(Wx_{t_{k+1}})\right\|_2 \leq \sqrt{n_t}\, \|\mathrm{ReLU}(Wx_1) - \mathrm{ReLU}(Wx_2)\|_2 . \tag{102}$$

Proof. Let $x_t = \ell^{x_1,x_2}(t)$, then as a function of t, the j'th component of $\mathrm{ReLU}(Wx_t)$ is either increasing (if $\langle w_j, x_2 - x_1\rangle \geq 0$) or decreasing (if $\langle w_j, x_2 - x_1\rangle \leq 0$). In either case, it is clear that for each $j = 1, \ldots, m$

$$(\mathrm{ReLU}(Wx_1) - \mathrm{ReLU}(Wx_2))|_j \cdot (\mathrm{ReLU}(Wx_2) - \mathrm{ReLU}(Wx_3))|_j \geq 0 \tag{103}$$

hence clearly

$$\langle (\mathrm{ReLU}(Wx_1) - \mathrm{ReLU}(Wx_2)), (\mathrm{ReLU}(Wx_2) - \mathrm{ReLU}(Wx_3))\rangle \geq 0 \tag{104}$$

thus we can apply (99) and we obtain (101). Applying (98) then yields (102). □

Appendix E Detailed Comparison to Prior Work

E.1 Comparison to Bruna *et al.*

In Bruna et al. (2013, Proposition 2.2.) the authors give a result invoking a condition similar to our DSS condition (Definition 1). It also concerns injectivity of a ReLU layer in terms of the injectivity of the weight matrix restricted to certain rows. The authors also compute a bi-Lipschitz bound for a layer (similar to our Theorem 3), though as we show in the following examples their analysis is in some cases not precisely aligned with injectivity.

Their criterion is given in two parts. For a weight matrix, they first define a notion of admissible set which indicates the points where the weight matrix's injectivity must be tested. Injectivity follows provided that the weight matrix is non-singular when restricted to each admissible set. Given a weight matrix $W \in \mathbb{R}^{M\times N}$ and bias b, the authors say that $\Omega \subset \{1, \ldots, M\}$ is admissible if

$$\bigcap_{i\in\Omega} \{x\colon \langle x, w_i\rangle > b_i\} \cap \bigcap_{i\notin\Omega} \{x\colon \langle x, w_i\rangle < b_i\} \tag{105}$$

is not empty. For our analysis we focus on the case when $b \equiv 0$. In this case Ω is admissible if and only if

$$\exists x \in \mathbb{R}^n \quad \text{such that} \quad \langle x, w_i\rangle \begin{cases} > 0 \text{ if } i \in \Omega \\ < 0 \text{ if } i \notin \Omega \end{cases} . \tag{106}$$

Note that the inequality in (106) is strict, unlike (2). If, for example, W has a column that is the zero vector, then there are no admissible Ω. The authors use the notation $\overline{\Omega}$ to denote all admissible sets for a given weight matrix. In their notation F is the transpose of our weight matrix W, F_Ω are the Ω rows of the weight matrix, $F_\Omega|_{V_\Omega}$ is the subspace generated by the Ω rows of W. The authors also call the ReLU function the half-rectification function. $\lambda_-(F)$ and $\lambda_+(F)$ denote the lower and upper frame bounds of F respectively. The injectivity criterion from Bruna et al. (2013) is

Proposition 1. *Let* $A_0 = \min_{\Omega \in \overline{\Omega}} \lambda_-(F_\Omega|_{V_\Omega})$. *Then the half-rectification operator* $M_b(x) = \mathrm{ReLU}(F^T x + b)$ *is injective if and only if* $A_0 > 0$. *Moreover, it satisfies*

$$\forall x, x', A_0 \|x - x'\| \leq \|M_b(x) - M_b(x')\| \leq B_0 \|x - x'\| \tag{107}$$

with $B_0 = \max_{\Omega \in \overline{\Omega}} \lambda_+(F_\Omega) \leq \lambda_+(F)$.

We now show that Proposition 1 does not precisely align with injectivity of $\mathrm{ReLU}(W(\cdot))$. We construct a weight matrix for which $A_0 > 0$, but does not yield an injective $\mathrm{ReLU}(W(\cdot))$. If

$$W = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{108}$$

then clearly $\mathrm{ReLU}(Wx)$ is not injective (for all $\alpha < 0, \mathrm{ReLU}(W \begin{bmatrix} 0 \\ \alpha \end{bmatrix}) = \overline{0}$). The only admissible sets are $\bar{\Omega} = \{\{1\}, \{3\}, \{1, 2\}, \{2, 3\}\}$ (notably $\{2\}$ is not admissible). W is full rank on all $\Omega \in \bar{\Omega}$, so $A_0 > 0$ so Proposition 1 implies that $\mathrm{ReLU}(W(\cdot))$ is injective. Now consider the case when

$$W = \begin{bmatrix} B \\ -DB \\ 0 \end{bmatrix} \tag{109}$$

where B is a basis of $\mathbb{R}^n$, D is a strictly positive diagonal matrix, and 0 is the zero row vector. From Corollary 2, W satisfies Theorem 1, and so $\mathrm{ReLU}(W(\cdot))$ is injective. On account of the zero row vector in (109), $\forall x \in \mathbb{R}^n$, $\langle x, 0 \rangle = 0$ so there are no Ω that are admissible Ω according to (105). Thus A_0 is undefined.

Now we construct an example of a W and $x, x' \in \mathbb{R}^n$ for which $A_0 \|x - x'\| > \|\mathrm{ReLU}(Wx) - \mathrm{ReLU}(Wx')\|$. Let

$$W = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad x = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad x' = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix}. \tag{110}$$

Clearly on every admissible set $\lambda_-(W_\Omega|_{V_\Omega}) = 1$, so

$$\begin{aligned} \mathrm{ReLU}(Wx) &= \frac{1}{\sqrt{2}} \mathrm{ReLU}\left(\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \right) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \\ \mathrm{ReLU}(Wx') &= \frac{1}{\sqrt{2}} \mathrm{ReLU}\left(\begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \right) = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \end{aligned} \tag{111}$$

hence,

$$\|\mathrm{ReLU}(Wx) - \mathrm{ReLU}(Wx')\| = \frac{1}{\sqrt{2}} \left\| \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} \right\| = 1 \tag{112}$$

and

$$\|x - x'\| = \frac{1}{\sqrt{2}} \left\| \begin{bmatrix} 2 \\ 0 \end{bmatrix} \right\| = \sqrt{2}. \tag{113}$$

From this we have

$$\sqrt{2} = A_0 \|x - x'\| > \|\mathrm{ReLU}(Wx) - \mathrm{ReLU}(Wx')\| = 1. \tag{114}$$

On the other hand, substituting this into (6) yields

$$\frac{C}{\sqrt{m}} \|x - x'\| = \frac{1}{\sqrt{8}} \sqrt{2} = \frac{1}{2} \leq 1, \tag{115}$$

which does hold, suggesting that the lower bound in Proposition 1 is not pessimistic enough.

E.2 RELATIONSHIP TO MALLAT ET AL. (2018)

In Mallat et al. (2018) the authors consider a construction analogous to our convolutional construction (in Definition 3) defined on a continuum (i.e. infinite-dimensional function defined on an interval) rather than on a vector (i.e. discrete finite dimensional function) defined on a subset of $\mathbb{R}^n$. The authors posit that CNNs first learn a layer of filters localized in frequency varied in phase. The authors also show that a a ReLU activation function acts as a filter on the phase of the convolution of the filters against the input signal and that, provided that the filters are sufficiently different in phase and satisfy a frame condition then the layer is bi-Lipschitz, and hence is injective. Their analysis a particularized version of ours, and can be straight forwardly subsumed by our work.

The frame condition is given by Proposition 2.6 in Mallat et al. (2018) that the weight matrix must satisfy in order ensure that W is invertible and stable. In the notation of Mallat et al. (2018) the filters $\hat{\psi}_\lambda$ are analogous to the Fourier transform of the kernels in Definition 2, and the condition in (2.25) in Mallat et al. (2018) is one natural way to generalize the notion of a basis to a continuous signal. Hence, Proposition 2.6 in Mallat et al. (2018) can be loosely interpreted as a statement that the kernels in a given layer of width P form a basis of $\mathbb{R}^P$.

The second condition is that kernels are given in terms of belonging to a family, and members of this family are related to each other in the sense that members of the same family are centered in the same Fourier domain, and act as phase offsets of differing phase. Equation 2.14 of Mallat et al. (2018) describes that a phase filter $H\colon \mathbb{C} \times [0, 2, \pi] \to \mathbb{C}$ is defined by

$$\forall z \in \mathbb{C}, \alpha \in [0, 2\pi], \quad Hz(\alpha) = |z| h(\alpha - \varphi(z)) \tag{116}$$

where $|z|$ standard modulus of a complex number, $\varphi(z)$ is the complex phase, and $h(\alpha) = \mathrm{ReLU}(\cos(\alpha))$. If we consider just the value of $Hz(0)$ and $Hz(\pi)$, then we find

$$Hz(0) = |z| \,\mathrm{ReLU}(\cos(\varphi(z))) = \mathrm{ReLU}(|z| \cos(\varphi(z))) = \mathrm{ReLU}(\mathcal{R}z), \tag{117}$$
$$Hz(\pi) = |z| \,\mathrm{ReLU}(-\cos(\varphi(z))) = \mathrm{ReLU}(-|z| \cos(\varphi(z))) = \mathrm{ReLU}(-\mathcal{R}z), \tag{118}$$

where $\mathcal{R}z$ is the real part of z. If (as in Mallat et al. (2018)) z is given by $z = x \star c_k$, where x is a real signal and $\star c_k$ denotes the convolution against a kernel c_k, then (117) and (118) imply that

$$Hz(0) = \mathrm{ReLU}(x \star \mathcal{R}c_k), \tag{119}$$
$$Hz(\pi) = \mathrm{ReLU}(x \star (-\mathcal{R}c_k)), \tag{120}$$

that is, that for every kernel c_k in a layer, the kernel $-c_k$ is also in that layer.-

Combining the two logical conditions above implies that the kernels of width c_k form a basis of $\mathbb{R}_P$ and that for every kernel c_k there is also a kernel $-c_k$. Together these two (by Corollary 2) that the c_k form a DSS of $\mathbb{R}^P$, and thus by Theorem 4 the entire layer is injective.

APPENDIX F ARCHITECTURE DETAILS FOR EXPERIMENTS

Generator network: We train a generator with 5 convolutional layers. The input latent code is 256-dimensional which is treated by the network as a $1 \times 1 \times 256$ size tensor. The first layer is a transposed convolution with a kernel size of 4×4 with stride 1 and 1024 output channels. This is followed by a leaky ReLU. We follow this up by 3 conv layers each of which halve the number of channels and double the image size (i.e. we go from $N/2 \times N/2 \times C$ to $N \times N \times C/2$ tensor) giving an expansivity of two, the minimum required for injectivity of ReLU networks. Each of these 3 convolution layers has kernel size 3, stride 2 and is followed by the ReLU activation. These layers are made injective by having half the filters as w and the other half as $-s^2 w$. Here, w and s are trainable parameters. The biases in these layers are kept at zero. We do not employ any normalization schemes. Lastly, we have a convolution layer at the end to get to 3 channels and required image size. This layer is followed by the sigmoidal activation. We compare this to a regular GAN which has all the same architectural components including nonlinearities except the filters are not chosen as w and $-s^2 w$ and we also allow biases (see Figure 9 for a qualitative comparison).

Figure 9: Samples generated with a DCGAN on FFHQ dataset: injective layers (left) vs generic layers (right)

Critic network: The discriminator has 5 convolution layers with $128, 256, 512, 1024$ and 1 channels per layer. Each convolution layer has 4×4 kernels with stride 2. Each layer is followed by the leaky-ReLU activation function. The last layer of the network is followed by identity.

Inference network: The inference network has the same architecture as the first 4 convolution layers of the discriminator. This is followed by 3 fully-connected layers of size $512, 256$ and 256. The first 2 fully-connected layers have a Leaky ReLU activation while the last layer has identity activation function. The inference net is trained in tandem with the GAN.

We use the Wasserstein loss with gradient penalty (Gulrajani et al., 2017) to train our networks. We train for 40 epochs on a data set of size 80000 samples. We use a batch size of 64 and Adam optimizer for training with learning rate of 10^{-4}.

We report FID (Heusel et al., 2017) and Inception score (Salimans et al., 2016) using 10000 generated samples. The standard deviation was calculated using 5 sets of 10000 generated samples. In order to calculate the mean and covariance of generated distributions, we sample 50000 codes.