Representational aspects of depth and conditioning in normalizing flows

Frederic Koehler [1] **Viraj Mehta** [2] **Andrej Risteski** [3]

Abstract

Normalizing flows are among the most popular paradigms in generative modeling, especially for images, primarily because we can efficiently evaluate the likelihood of a data point. This is desirable both for evaluating the fit of a model, and for ease of training, as maximizing the likelihood can be done by gradient descent. However, training normalizing flows comes with difficulties as well: models which produce good samples typically need to be extremely deep – which comes with accompanying vanishing/exploding gradient problems. A very related problem is that they are often poorly *conditioned*: since they are parametrized as invertible maps from $\mathbb{R}^d \to \mathbb{R}^d$, and typical training data like images intuitively is lower-dimensional, the learned maps often have Jacobians that are close to being singular.

In our paper, we tackle representational aspects around depth and conditioning of normalizing flows: both for general invertible architectures, and for a particular common architecture, affine couplings. We prove that $\Theta(1)$ affine coupling layers suffice to exactly represent a permutation or 1×1 convolution, as used in GLOW, showing that representationally the choice of partition is not a bottleneck for depth. We also show that shallow affine coupling networks are universal approximators in Wasserstein distance if ill-conditioning is allowed, and experimentally investigate related phenomena involving padding. Finally, we show a depth lower bound for general flow architectures with few neurons per layer and bounded Lipschitz constant.

[1]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA [2]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA [3]Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA, USA. Correspondence to: Viraj Mehta <virajm@cs.cmu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

1. Introduction

Deep generative models are one of the lynchpins of unsupervised learning, underlying tasks spanning distribution learning, feature extraction and transfer learning. Parametric families of neural-network based models have been improved to the point of being able to model complex distributions like images of human faces. One paradigm that has received a lot attention is normalizing flows, which model distributions as pushforwards of a standard Gaussian (or other simple distribution) through an *invertible* neural network G. Thus, the likelihood has an explicit form via the change of variables formula using the Jacobian of G. Training normalizing flows is challenging due to a couple of main issues. Empirically, these models seem to require a much larger size than other generative models (e.g. GANs) and most notably, a much larger depth. This makes training challenging due to vanishing/exploding gradients. A very related problem is *conditioning*, more precisely the smallest singular value of the forward map G. It's intuitively clear that natural images will have a low-dimensional structure, thus a close-to-singular G might be needed. On the other hand, the change-of-variables formula involves the determinant of the Jacobian of G^{-1}, which grows larger the more singular G is.

While recently, the universal approximation power of various types of invertible architectures has been studied if the input is padded with a sufficiently large number of all-0 coordinates (Dupont et al., 2019; Huang et al., 2020) or arbitrary partitions and permutations are allowed (Teshima et al., 2020), precise quantification of the cost of invertibility in terms of the depth required and the conditioning of the model has not been fleshed out.

In this paper, we study both mathematically and empirically representational aspects of depth and conditioning in normalizing flows and answer several fundamental questions.

2. Related Work

On the empirical side, flow models were first popularized by (Dinh et al., 2014), who introduce the NICE model and the idea of parametrizing a distribution as a sequence of transformations with triangular Jacobians, so that maximum likelihood training is tractable. Quickly thereafter, (Dinh et al., 2016) improved the affine coupling block architecture they

introduced to allow non-volume-preserving (NVP) transformations, (Papamakarios et al., 2017) introduced an autoregressive version, and finally (Kingma & Dhariwal, 2018) introduced 1x1 convolutions in the architecture, which they view as relaxations of permutation matrices—intuitively, allowing learned partitions for the affine blocks. Subsequently, there have been variants on these ideas: (Grathwohl et al., 2018; Dupont et al., 2019; Behrmann et al., 2018) viewed these models as discretizations of ODEs and introduced ways to approximate determinants of non-triangular Jacobians, though these models still don't scale beyond datasets the size of CIFAR10. The conditioning/invertibility of trained models was experimentally studied in (Behrmann et al., 2019), along with some "adversarial vulnerabilities" of the conditioning. Mathematically understanding the relative representational power and statistical/algorithmic implications thereof for different types of generative models is still however a very poorly understood and nascent area of study.

Most closely related to our results are the recent works of (Huang et al., 2020), (Zhang et al.) and (Teshima et al., 2020). The first two prove universal approximation results for invertible architectures (the former affine couplings, the latter neural ODEs) if the input is allowed to be padded with zeroes. The latter proves universal approximation when GLOW-style permutation layers are allowed through a construction that operates on one dimension at a time. This is very different than how flows are trained in practice, which is typically with a partition which splits the data roughly in half. It also requires the architectural modification of GLOW to work. As we'll discuss in the following section, our results prove universal approximation even without padding and permutations, but we focus on more fine-grained implications to depth and conditioning of the learned model and prove universal approximation in a setting that is used in practice. Another work (Kong & Chaudhuri, 2020) studies the representational power of Sylvester and Householder flows, normalizing flow architectures which are quite different from affine coupling networks. In particular, they prove a depth lower bound for local planar flows with bounded weights; for planar flows, our general Theorem 5 can also be applied, but the resulting lower bound instances are very different (ours targets multimodality, theirs targets tail behavior).

More generally, there are various classical results that show a particular family of generative models can closely approximate most sufficiently regular distributions over some domain. Some examples are standard results for mixture models with very mild conditions on the component distribution (e.g. Gaussians, see (Everitt, 2014)); Restricted Boltzmann Machines and Deep Belief Networks (Montúfar et al., 2011; Montufar & Ay, 2011); GANs (Bailey & Telgarsky, 2018).

3. Overview of Results

3.1. Results About Affine Coupling Architectures

We begin by proving several results for a particularly common normalizing flow architectures: those based on affine coupling layers (Dinh et al., 2014; 2016; Kingma & Dhariwal, 2018). The appeal of these architecture comes from training efficiency. Although layerwise invertible neural networks (i.e. networks for which each layer consists of an invertible matrix and invertible pointwise nonlinearity) seem like a natural choice, in practice these models have several disadvantages: for example, computing the determinant of the Jacobian is expensive unless the weight matrices are restricted.

Consequently, it's typical for the transformations in a flow network to be constrained in a manner that allows for efficient computation of the Jacobian determinant. The most common building block is an *affine coupling* block, originally proposed by (Dinh et al., 2014; 2016). A coupling block partitions the coordinates $[d]$ into two parts: S and $[d] \setminus S$, for a subset S with $|S|$ containing around half the coordinates of d. The transformation then has the form:

Definition 1. An *affine coupling block* is a map $f : \mathbb{R}^d \to \mathbb{R}^d$, s.t. . $f(x_S, x_{[d]\setminus S}) = (x_S, x_{[d]\setminus S} \odot s(x_S) + t(x_S))$, $s(x) > 0, \forall x \in \mathbb{R}^d$.

Of course, the modeling power will be severely constrained if the coordinates in S never change: so typically, flow models either change the set S in a fixed or learned way (e.g. alternating between different partitions of the channel in (Dinh et al., 2016) or applying a learned permutation in (Kingma & Dhariwal, 2018)). As a permutation is a discrete object, it is difficult to learn in a differentiable manner – so (Kingma & Dhariwal, 2018) simply learns an invertible linear function (i.e. a 1x1 convolution) as a differentiation-friendly relaxation thereof. In order to preserve the invertibility of an affine coupling, s is typically restricted to be strictly positive.

3.1.1. Universal Approximation with Ill-Conditioned Affine Coupling Networks

First, we address universal approximation of normalizing flows and its close ties to conditioning. Namely, a recent work (Theorem 1 of (Huang et al., 2020)) showed that deep affine coupling networks are universal approximators if we allow the training data to be padded with sufficiently many zeros. While zero padding is convenient for their analysis (in fact, similar proofs have appeared for other invertible architectures like Augmented Neural ODEs (Zhang et al.)), in practice models trained on zero-padded data often perform poorly (see Appendix D). Another work (Teshima et al., 2020) proves universal approximation with the optional per-

mutations and $|S| = d - 1$ needed for the nonconstructive proof. We remove that requirement in two ways, first by giving a construction that gives universal approximation without permutations in 3 composed couplings and second by showing that the permutations can be simulated by a constant number of alternating but fixed coupling layers.

First we show that neither padding nor permutations nor depth is necessary representationally: shallow models without zero padding are already universal approximators in Wasserstein.

Theorem 1 (Universal approximation without padding)**.** *Suppose that P is the standard Gaussian measure in $\mathbb{R}^n$ with n even and Q is a distribution on $\mathbb{R}^n$ with bounded support and absolutely continuous with respect to the Lebesgue measure. Then for any $\epsilon > 0$, there exists a network g consisting of 3 alternating affine couplings, with maps s, t represented by feedforward ReLU networks such that $W_2(g_{\#}P, Q) \leq \epsilon$.*

Remark 1. A shared caveat of the universality construction in Theorem 1 with the construction in (Huang et al., 2020) is that the resulting network is poorly conditioned. In the case of the construction in (Huang et al., 2020), this is obvious because they pad the d-dimensional training data with d additional zeros, and a network that takes as input a Gaussian distribution in $\mathbb{R}^{2d}$ (i.e. has full support) and outputs data on d-dimensional manifold (the space of zero padded data) must have a singular Jacobian almost everywhere.[1] In the case of Theorem 1, the condition number of the network blows up at least as quickly as $1/\epsilon$ as we take the approximation error $\epsilon \to 0$, so this network is also ill-conditioned if we are aiming for a very accurate approximation.

Remark 2. Based on Theorem 3, the condition number blowup of either the Jacobian or the Hessian is necessary for a shallow model to be universal, even when approximating well-conditioned linear maps (see Remark 6). The network constructed in Theorem 1 is also consistent with the lower bound from Theorem 5, because the network we construct in Theorem 1 has a large Lipschitz constant and uses many parameters per layer.

3.1.2. The effect of choice of partition on depth

Next, we ask how much of a saving in terms of the depth of the network can one hope to gain from using learned partitions (ala GLOW) as compared to a fixed partition. More precisely:

Question 1: Can models like Glow (Kingma & Dhariwal, 2018) be simulated by a sequence of affine blocks with a fixed partition without increasing the depth by much?

[1] Alternatively, we could feed a degenerate Gaussian supported on a d-dimensional subspace into the network as input, but there is no way to train such a model using maximum-likelihood training, since the prior is degenerate.

We answer this question in the affirmative at least for equally sized partitions (which is what is typically used in practice). We show the following surprising fact: consider an arbitrary partition $(S, [2d] \setminus S)$ of $[2d]$, such that S satisfies $|S| = d$, for $d \in \mathbb{N}$. Then for any invertible matrix $T \in \mathbb{R}^{2d \times 2d}$, the linear map $T : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ can be exactly represented by a composition of $O(1)$ affine coupling layers that are *linear*, namely have the form $L_i(x_S, x_{[2d]\setminus S}) = (x_S, B_i x_{[2d]\setminus S} + A_i x_S)$ or $L_i(x_S, x_{[2d]\setminus S}) = (C_i x_S + D_i x_{[2d]\setminus S}, x_{[2d]\setminus S})$ for matrices $A_i, B_i, C_i, D_i \in \mathbb{R}^{d \times d}$, s.t. each B_i, C_i is diagonal. For convenience of notation, without loss of generality let $S = [d]$. Then, each of the layers L_i is a matrix of the form $\begin{bmatrix} I & 0 \\ A_i & B_i \end{bmatrix}$ or $\begin{bmatrix} C_i & D_i \\ 0 & I \end{bmatrix}$, where the rows and columns are partitioned into blocks of size d.

With this notation in place, we show the following theorem:

Theorem 2. *For all $d \geq 4$, there exists a $k \leq 24$ such that for any invertible $T \in \mathbb{R}^{2d \times 2d}$ with $det(T) > 0$, there exist matrices $A_i, D_i \in \mathbb{R}^{d \times d}$ and diagonal matrices $B_i, C_i \in \mathbb{R}^{d \times d}_{\geq 0}$ for all $i \in [k]$ such that*

$$T = \prod_{i=1}^{k} \begin{bmatrix} I & 0 \\ A_i & B_i \end{bmatrix} \begin{bmatrix} C_i & D_i \\ 0 & I \end{bmatrix}$$

Note that the condition $\det(T) > 0$ is required, since affine coupling networks are always orientation-preserving. Adding one diagonal layer with negative signs suffices to model general matrices. In particular, since permutation matrices are invertible, this means that any applications of permutations to achieve a different partition of the inputs (e.g. like in Glow (Kingma & Dhariwal, 2018)) can in principle be represented as a composition of not-too-many affine coupling layers.

It's a reasonable to ask how optimal the $k \leq 24$ bound is – we supplement our upper bound with a lower bound, namely that $k \geq 3$. This is surprising, as naive parameter counting would suggest $k = 2$ might work. Namely, we show:

Theorem 3. *For all $d \geq 4$ and $k \leq 2$, there exists an invertible $T \in \mathbb{R}^{2d \times 2d}$ with $det(T) > 0$, s.t. for all $A_i, D_i \in \mathbb{R}^{d \times d}$ and for all diagonal matrices $B_i, C_i \in \mathbb{R}^{d \times d}_{\geq 0}, i \in [k]$ it holds that*

$$T \neq \prod_{i=1}^{k} \begin{bmatrix} I & 0 \\ A_i & B_i \end{bmatrix} \begin{bmatrix} C_i & D_i \\ 0 & I \end{bmatrix}$$

Beyond the relevance of this result in the context of how important the choice of partitions is, it also shows a lower bound on the depth for an equal number of *nonlinear* affine coupling layers (even with quite complex functions s and t in each layer) – since a nonlinear network can always be linearized about a (smooth) point to give a linear network with the same number of layers.

Corollary 4. *There exists a continuous function f which cannot be exactly represented by a depth-4 affine coupling network with arbitrary continuously differentiable functions as the s and t functions in each block.*

The proof is in Appendix B.3. In other words, studying linear affine coupling networks lets us prove a *depth lower bound/depth separation* for nonlinear networks for free.

Finally, in Section 5.3, we include an empirical investigation of our theoretical results on synthetic data, by fitting random linear functions of varying dimensionality with linear affine networks of varying depths in order to see the required number of layers. The results there suggest that the constant in the upper bound is quite loose – and the correct value for k is likely closer to the lower bound – at least for random matrices.

Remark 3 (Significance of Theorem 2 for Approximation in Likelihood/KL)**.** All of the universality results in the literature for normalizing flows, including Theorem 1, prove universality in the Wasserstein distance or in the related sense of convergence of distributions. A stronger and probably much more difficult problem is to prove universality under the KL divergence instead: i.e. to show for a well-behaved distribution P, there exists a sequence Q_n of distributions generated by normalizing flow models such that

$$\mathrm{KL}(P, Q_n) \to 0. \tag{1}$$

This is important because Maximum-Likelihood training attempts to pick the model with the smallest KL divergence to the empirical distribution, not the smallest Wasserstein distance, and the minimizers of these two objectives can be extremely different. For $P = N(0, \Sigma)$, Theorem 2 certainly implies (1) for bounded depth linear affine couplings, and thus gives the first proof that global optimization of the max-likelihood objective with unlimited data of a normalizing flow model would successfully learn a Gaussian with arbitrary nondegenerate Σ; see Appendix B.4.

3.2. Results about General Architectures

In order to guarantee that the network is invertible, normalizing flow models place significant restrictions on the architecture of the model. The most basic and general question we can ask is how this restriction affects the expressive power of the model — in particular, how much the depth must increase to compensate.

More precisely, we ask:

Question 2: is there a distribution over $\mathbb{R}^d$ which can be written as the pushforward of a Gaussian through a small, shallow generator, which cannot be approximated by the pushforward of a Gaussian through a small, shallow *layerwise invertible* neural network?

Given that there is great latitude in terms of the choice of layer architecture, while keeping the network invertible, the most general way to pose this question is to require each layer to be a function of p parameters – i.e. $f = f_1 \circ f_2 \circ \cdots \circ f_\ell$ where $\circ$ denotes function composition and each $f_i : \mathbb{R}^d \to \mathbb{R}^d$ is an invertible function specified by a vector $\theta_i \in \mathbb{R}^p$ of parameters. This framing is extremely general: for instance it includes *layerwise invertible feedforward networks* in which $f_i(x) = \sigma^{\otimes d}(A_i x + b_i)$, σ is invertible, $A_i \in \mathbb{R}^{d \times d}$ is invertible, $\theta_i = (A_i, b_i)$ and $p = d(d+1)$. It also includes popular architectures based on affine coupling blocks which we discussed in more detail in the previous subsection.

We answer this question in the affirmative: namely, we show for any k that there is a distribution over $\mathbb{R}^d$ which can be expressed as the pushforward of a network with depth $O(1)$ and size $O(k)$ that cannot be (even very approximately) expressed as the pushforward of a Gaussian through a Lipschitz layerwise invertible network of depth smaller than k/p.

Towards formally stating the result, let $\theta = (\theta_1, \ldots, \theta_\ell) \in \Theta \subset \mathbb{R}^{d'}$ be the vector of all parameters (e.g. weights, biases) in the network, where $\theta_i \in \mathbb{R}^p$ are the parameters that correspond to layer i, and let $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ denote the resulting function. Define R so that Θ is contained in the Euclidean ball of radius R.

We say the family f_θ is *L-Lipschitz with respect to its parameters and inputs*, if

$$\forall \theta, \theta' \in \Theta : \mathrm{E}_{x \sim \mathcal{N}(0, I_{d \times d})} \|f_\theta(x) - f_{\theta'}(x)\| \le L \|\theta - \theta'\|$$

and $\forall x, y \in \mathbb{R}^d, \|f_\theta(x) - f_\theta(y)\| \le L\|x - y\|$. [2] We will discuss the reasonable range for L in terms of the weights after the Theorem statement. We show[3]:

Theorem 5. *For any $k = \exp(o(d)), L = \exp(o(d)), R = \exp(o(d))$, we have that for d sufficiently large and any $\gamma > 0$ there exists a neural network $g : \mathbb{R}^{d+1} \to \mathbb{R}^d$ with $O(k)$ parameters and depth $O(1)$, s.t. for any family $\{f_\theta, \theta \in \Theta\}$ of layerwise invertible networks that are L-Lipschitz with respect to its parameters and inputs, have p parameters per layer and depth at most k/p we have*

$$\forall \theta \in \Theta, W_1((f_\theta)_{\#\mathcal{N}}, g_{\#\mathcal{N}}) \ge 10\gamma^2 d$$

Furthermore, for all $\theta \in \Theta$, KL$((f_\theta)_{\#\mathcal{N}}, g_{\#\mathcal{N}}) \ge 1/10$ and KL$(g_{\#\mathcal{N}}, (f_\theta)_{\#\mathcal{N}}) \ge \frac{10\gamma^2 d}{L^2}$.

[2] Note for architectures having trainable biases in the input layer, these two notions of Lipschitzness should be expected to behave similarly.

[3] In this Theorem and throughout, we use the standard asymptotic notation $f(d) = o(g(d))$ to indicate that $\limsup_{d \to \infty} \frac{f(d)}{g(d)} = 0$. For example, the assumption $k, L, R = \exp(o(d))$ means that for any sequence $(k_d, L_d, R_d)_{d=1}^\infty$ such that $\limsup_{d \to \infty} \frac{\max(\log k_d, \log L_d, \log R_d)}{d} = 0$ the result holds true.

Remark 4. First, note that while the number of parameters in both networks is comparable (i.e. it's $O(k)$), the invertible network is deeper, which usually is accompanied with algorithmic difficulties for training, due to vanishing and exploding gradients. For layerwise invertible generators, if we assume that the nonlinearity σ is 1-Lipschitz and each matrix in the network has operator norm at most M, then a depth ℓ network will have $L = O(M^\ell)$[4] and $p = O(d^2)$. For an affine coupling network with g, h parameterized by H-layer networks with $p/2$ parameters each, 1-Lipschitz activations and weights bounded by M as above, we would similarly have $L = O(M^{\ell H})$.

Remark 5. We make a couple of comments on the "hard" distribution g we construct, as well as the meaning of the parameter γ and how to interpret the various lower bounds in the different metrics. The distribution g for a given γ will in fact be close to a mixture of k Gaussians, each with mean on the sphere of radius $10\gamma^2 d$ and covariance matrix $\gamma^2 I_d$. Thus this distribution has most of it's mass in a sphere of radius $O(\gamma^2 d)$ — so the Wasserstein guarantee gives close to a trivial approximation for g. The KL divergence bounds are derived by so-called transport inequalities between KL and Wasserstein for subgaussian distributions (Bobkov & Götze, 1999). The discrepancy between the two KL divergences comes from the fact that the functions g, f_θ may have different Lipschitz constants, hence the tails of $g_{\#\mathcal{N}}$ and $f_{\#\mathcal{N}}$ behave differently. In fact, if the function f_θ had the same Lipschitz constant as g, both KL lower bounds would be on the order of a constant.

Practical takeaways from our results. Theorem 2 suggests the (representational) value of 1x1 convolutions as in (Kingma & Dhariwal, 2018) is limited, as we can simulate them with a (small) constant number of affine couplings. Theorem 1 shows that though affine couplings are universal approximators (even without padding), such constructions may result in poorly conditioned networks, even if the target distributions they are approximating are nondegenerate. Finally, Theorem 5 makes quantitative the intuition that normalizing flow models with small layers may need to be deep to model complex distributions.

4. Proof Sketch of Theorem 1: Universal Approximation with Ill-Conditioned Affine Coupling Networks

In this section, we sketch the proof of Theorem 1 to show how to approximate a distribution in $\mathbb{R}^n$ using three layers of affine coupling networks, where the dimension $n = 2d$ is even. The partition in the affine coupling network is between the first d coordinates and second d coordinates in $\mathbb{R}^{2d}$.

The first element in the proof is a well-known theorem from optimal transport called Brenier's theorem, which states that for Q a probability measure over $\mathbb{R}^n$ satisfying weak regularity conditions (see Theorem 8 in Section A), there exists a map $\varphi : \mathbb{R}^n \to \mathbb{R}^n$ such that if $X \sim N(0, I_{n\times n})$, then the pushforward $\varphi_\#(X)$ is distributed according to Q.

The proof then proceeds by using a lattice-based encoding and decoding scheme. Concretely, let $\epsilon > 0$ be a small constant, to be taking sufficiently small. Let $\epsilon' \in (0, \epsilon)$ be a further constant, taken sufficiently small with respect to ϵ and similar for ϵ'' wrt ϵ'. Let the input to the affine coupling network be $X = (X_1, X_2)$ such that $X_1 \sim N(0, I_{d\times d})$ and $X_2 \sim N(0, I_{d\times d})$. Let $f(x)$ be the map which rounds $x \in \mathbb{R}^d$ to the closest grid point in the lattice $\epsilon\mathbb{Z}^d$ and define $g(x) = x - f(x)$. Note that for a point of the form $z = f(x) + \epsilon' y$ for y which is not too large, we have that $f(z) = f(x)$ and $g(z) = y$. Suppose the optimal transportation map from Brenier's Theorem is $\varphi(x) = (\varphi_1(x), \varphi_2(x))$ where $\varphi_1, \varphi_2 : \mathbb{R}^d \to \mathbb{R}^n$ correspond to the two halves of the output. Now we consider the following sequence of maps, all which form an affine coupling layer:

$$
\begin{aligned}
&(X_1, X_2)\\
&\mapsto (X_1, \epsilon' X_2 + f(X_1))\\
&\mapsto (f(\varphi_1(f(X_1), X_2)) + \epsilon'\varphi_2(f(X_1), X_2) + O(\epsilon''),\\
&\qquad \epsilon' X_2 + f(X_1))\\
&\mapsto (f(\varphi_1(f(X_1), X_2)) + \epsilon'\varphi_2(f(X_1), X_2) + O(\epsilon''),\\
&\qquad \varphi_2(f(X_1), X_2) + O(\epsilon''/\epsilon')).
\end{aligned}
$$

To explicitly see why the above are affine coupling layers, in the first step we take $s_1(x) = \log(\epsilon')\vec{1}$ and $t_1(x) = f(x)$. In the second step, we take $s_2(x) = \log(\epsilon'')\vec{1}$ and t_2 is defined by $t_2(x) = f(\varphi_1(f(x), g(x))) + \epsilon'\varphi_2(f(x), g(x))$. In the third step, we take $s_3(x) = \log(\epsilon'')\vec{1}$ and define $t_3(x) = \frac{g(x)}{\epsilon'}$. Taking sufficiently good approximations to all of the maps allows to approximate this map with neural networks, which we formalize in Appendix A.

4.1. Experimental Results

On the empirical side, we explore the effect that different types of padding has on the training on various synthetic datasets. For Gaussian padding, this means we add to the d-dimensional training data point an additional d dimensions sampled from $N(0, I_d)$. We consistently observe that zero padding has the worst performance and Gaussian padding has the best performance. For zero padding, we concatenate an additional d dimensions with value zero to the training data, as in the universality construction in (Huang et al., 2020). In both cases we then train the models using maximum likelihood, treating the augmented part of the samples as if it was generated as part of the training samples. In

[4]Note, our theorem applies to exponentially large Lipschitz constants.

Figure 1 we show the performance of a simple RealNVP architecture trained via max-likelihood on a mixture of 4 Gaussians, as well as plot the condition number of the Jacobian during training for each padding method. The latter gives support to the fact that conditioning is a major culprit for why zero padding performs so badly. In Appendix D.2 we provide figures from more synthetic datasets.

5. Proof Sketch of Theorems 2 and 3: Simulating Linear Functions with Affine Couplings

In this section, we will prove Theorems 3 and 2. Before proceeding to the proofs, we will introduce a bit of helpful notation. We let $GL^+(2d, \mathbb{R})$ denote the group of $2d \times 2d$ matrices with positive determinant (see (Artin, 2011) for a reference on group theory). The lower triangular linear affine coupling layers are the subgroup $\mathcal{A}_{\mathcal{L}} \subset GL^+(2d, \mathbb{R})$ of the form

$$\mathcal{A}_{\mathcal{L}} = \left\{ \begin{bmatrix} I & 0 \\ A & B \end{bmatrix} : A, B \in \mathbb{R}^{d \times d} \right\},$$

with B diagonal with positive entries and likewise the upper triangular linear affine coupling layers are the subgroup $\mathcal{A}_{\mathcal{U}} \subset GL^+(2d, \mathbb{R})$ of the form

$$\mathcal{A}_{\mathcal{U}} = \left\{ \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} : C, D \in \mathbb{R}^{d \times d} \right\},$$

with C diagonal with positive entries.

Finally, define $\mathcal{A} = \mathcal{A}_{\mathcal{L}} \cup \mathcal{A}_{\mathcal{U}} \subset GL^+(2d, \mathbb{R})$. This set is not a subgroup because it is not closed under multiplication. Let $\mathcal{A}^k$ denote the kth power of $\mathcal{A}$, i.e. all elements of the form $a_1 \cdots a_k$ for $a_i \in \mathcal{A}$.

5.1. Upper Bound

The main result of this section is the following:

Theorem 6 (Restatement of Theorem 2)**.** *There exists an absolute constant $1 < K \leq 47$ such that for any $d \geq 1$, $GL^+(2d, \mathbb{R}) = \mathcal{A}^K$.*

In other words, any linear map with positive determinant ("orientation-preserving") can be implemented using a bounded number of linear affine coupling layers. Note that there is a difference in a factor of two between the counting of layers in the statement of Theorem 2 and the counting of matrices in Theorem 6, because each layer is composed of two matrices.

In group-theoretic language, this says that $\mathcal{A}$ generates $GL^+(2d, \mathbb{R})$ and furthermore the diameter of the corresponding (uncountably infinite) Cayley graph is upper bounded by a constant independent of d. The proof relies on the following two structural results. The first one is

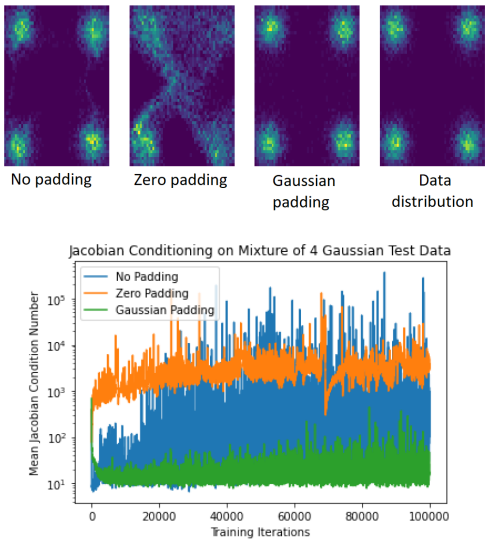

Figure 1. Fitting a 4-component mixture of Gaussians using a RealNVP model with no padding, zero padding and Gaussian padding.

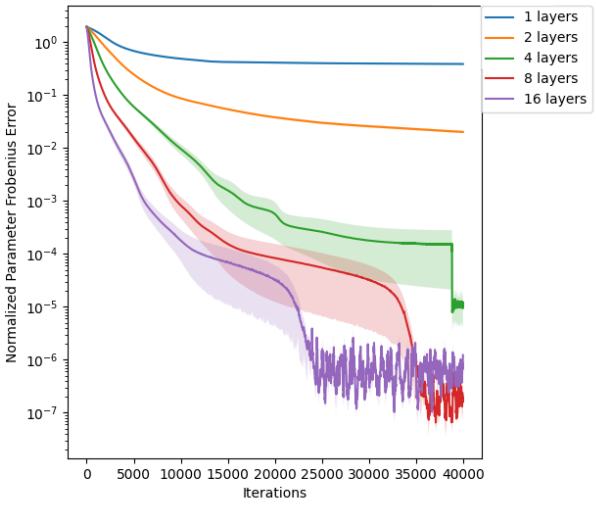

Figure 2. Fitting 32-dimensional linear maps on a using n-layer linear affine coupling networks. The squared Frobenius error is normalized by $1/d^2$ so it is independent of dimensionality. We shade the standard error regions of these losses across the seeds tried.

about representing permutation matrices, up to sign, using a constant number of linear affine coupling layers:

Lemma 1. *For any permutation matrix $P \in \mathbb{R}^{2d\times 2d}$, there exists $\tilde{P} \in \mathcal{A}^{21}$ with $|\tilde{P}_{ij}| = |P_{ij}|$ for all i, j.*

The second one proves how to represent using a constant number of linear affine couplings matrices with special eigenvalue structure:

Lemma 2. *Let M be an arbitrary invertible $d \times d$ matrix with distinct real eigenvalues and S be a $d \times d$ lower triangular matrix with the same eigenvalues as M^{-1}. Then $\begin{bmatrix} M & 0 \\ 0 & S \end{bmatrix} \in \mathcal{A}^4$.*

Given these Lemmas, we briefly describe the strategy to prove Theorem 6. Every matrix has a an LUP factorization (Horn & Johnson, 2012) into a lower-triangular, upper-triangular, and permutation matrix. Lemma 1 takes care of the permutation part, so what remains is building an arbitrary lower/upper triangular matrix; because the eigenvalues of lower-triangular matrices are explicit, a careful argument allows us to reduce this to Lemma 2. All the proofs are in Section B.

5.2. Lower Bound

We proceed to the lower bound. Note, a simple parameter counting argument shows that for sufficiently large d, at least four affine coupling layers are needed to implement an arbitrary linear map (each affine coupling layer has only $d^2 + d$ parameters whereas $GL_+(2d, \mathbb{R})$ is a Lie group of dimension $4d^2$). Perhaps surprisingly, it turns out that four affine coupling layers *do not* suffice to construct an arbitrary linear map. We prove this in the following Theorem.

Theorem 7 (Restatement of Theorem 3)**.** *For $d \geq 4$, $\mathcal{A}^4$ is a* proper subset *of $GL_+(2d, \mathbb{R})$. In other words, there exists a matrix $T \in GL_+(2d, \mathbb{R})$ which is not in $\mathcal{A}^4$.*

Again, this translates to the result in Theorem 3 because each layer corresponds to two matrices — so this shows two layers are not enough to get arbitrary matrices. The key observation is that matrices in $\mathcal{A}_\mathcal{L}\mathcal{A}_\mathcal{U}\mathcal{A}_\mathcal{L}\mathcal{A}_\mathcal{U}$ satisfy a strong algebraic invariant which is not true of arbitrary matrices. This invariant can be expressed in terms of the *Schur complement* (Zhang, 2006):

Lemma 3. *Suppose that $T = \begin{bmatrix} X & Y \\ Z & W \end{bmatrix}$ is an invertible $2d \times 2d$ matrix and suppose there exist matrices $A, E \in \mathbb{R}^{d\times d}$, $D, H \in \mathbb{R}^{d\times d}$ and diagonal matrices $B, F \in \mathbb{R}^{d\times d}$, $C, G \in \mathbb{R}^{d\times d}$ such that*

$$T = \begin{bmatrix} I & 0 \\ A & B \end{bmatrix} \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ E & F \end{bmatrix} \begin{bmatrix} G & H \\ 0 & I \end{bmatrix}.$$

Then the Schur complement $T/X := W - ZX^{-1}Y$ is similar to $X^{-1}C$: more precisely, if $U = Z - AX$ then $T/X = UX^{-1}CU^{-1}$.

The proof of this Lemma is presented in Appendix B, as well as the resulting proof of Theorem 7. We remark that the obstruction is reasonably general; it can be shown, for example, that for a random choice of X and W from the Ginibre ensemble, that T cannot typically be expressed in $\mathcal{A}_4$. So there are significant restrictions on what matrices can be expressed with even four affine coupling layers.

Remark 6 (Connection to Universal Approximation)**.** As mentioned earlier, this lower bound shows that the map computed by general 4-layer affine coupling networks is quite restricted in its local behavior (it's Jacobian cannot be arbitrary). This implies that smooth 4-layer affine coupling networks, where smooth means the Hessian (of each coordinate of the output) is bounded in spectral norm by a fixed constant, cannot be universal function approximators as they cannot even approximate some linear maps. In contrast, if we allow the computed function to be very jagged then three layers are universal (see Theorem 1).

5.3. Experimental results

We also verify the bounds from this section. At least on randomly chosen matrices, the number of layers required is closer to the lower bound. Precisely, we generate (synthetic) training data of the form Az, where $z \sim \mathcal{N}(0, I)$ for a fixed $d \times d$ square matrix A with random standard Gaussian entries and train a linear affine coupling network with $n = 1, 2, 4, 8, 16$ layers by minimizing the loss $\mathbb{E}_{z\sim\mathcal{N}(0,I)}\left[(f_n(z) - Az)^2\right]$. We are training this "supervised" regression loss instead of the standard unsupervised likelihood loss to minimize algorithmic (training) effects as the theorems are focusing on the representational aspects. The results for $d = 16$ are shown in Figure 2, and more details are in Section D. To test a different distribution other than the Gaussian ensemble, we also generated random Toeplitz matrices with constant diagonals by sampling the value for each diagonal from a standard Gaussian and performed the same regression experiments. We found the same dependence on number of layers but an overall higher error, suggesting that that this distribution is slightly 'harder'. We provide results in Section D. We also regress a nonlinear RealNVP architecture on the same problems and see a similar increase in representational power though the nonlinear models seem to require more training to reach good performance.

Additional Remarks Finally, we also note that there are some surprisingly simple functions that cannot be *exactly* implemented by a finite affine coupling network. For instance, an entrywise $\tanh$ function (i.e. an entrywise nonlinearity) cannot be exactly represented by any finite affine coupling network, regardless of the nonlinearity used. De-

tails of this are in Appendix E.

6. Proof Sketch of Theorem 5: Depth Lower Bounds on Invertible Models

In this section we sketch the proof of Theorem 5, leaving full details to Appendix C. The intuition behind the k/p bound on the depth relies on parameter counting: a depth k/p invertible network will have k parameters in total (p per layer)—which is the size of the network we are trying to represent. Of course, the difficulty is that we need more than f_θ, g simply not being identical: we need a quantitative bound in various probability metrics.

The proof will proceed as follows. First, we will exhibit a large family of distributions (of size $\exp(kd)$), s.t. each pair of these distributions has a large pairwise Wasserstein distance between them. Moreover, each distribution in this family will be approximately expressible as the pushforward of the Gaussian through a small neural network. Since the family of distributions will have a large pairwise Wasserstein distance, by the triangle inequality, no other distribution can be close to two distinct members of the family.

Second, we can count the number of "approximately distinct" invertible networks of depth l: each layer is described by p weights, hence there are lp parameters in total. The Lipschitzness of the neural network in terms of its parameters then allows to argue about discretizations of the weights.

Formally, we show the following lemma:

Lemma 4 (Large family of well-separated distributions)**.** *For every $k = o(exp(d))$, for d sufficiently large and $\gamma > 0$ there exists a family $\mathcal{D}$ of distributions, s.t. $|\mathcal{D}| \geq \exp(kd/20)$ and:*

1. *Each distribution $p \in \mathcal{D}$ is a mixture of k Gaussians with means $\{\mu_i\}_{i=1}^k, \|\mu_i\|^2 = 20\gamma^2 d$ and covariance $\gamma^2 I_d$.*
2. *$\forall p \in \mathcal{D}$ and $\forall \epsilon > 0$, we have $W_1(p, g_{\#\mathcal{N}}) \leq \epsilon$ for a neural network g with at most $O(k)$ parameters.*[5]
3. *For any $p, p' \in \mathcal{D}$, $W_1(p, p') \geq 20\gamma^2 d$.*

The proof of this Lemma relies on two steps: first, construction of a large family of statistically different mixtures of Gaussians, and second a generator for such mixtures given by pushing forward a Gaussian through a neural network.

For the first part, concentration of measure implies that there exists an exponentially large family of well-separated points on a high dimensional sphere. Given this, to design a family of mixtures of Gaussians with large pairwise Wasserstein distance, it suffices to construct a large family of size k subsets for the means such that no pair of sets overlap too much. This can be done leveraging tools from coding theory (essentially the Gilbert-Varshamov bound (Gilbert, 1952; Varshamov, 1957)).

To handle part 2 of Lemma 7, we also show that a mixture of k Gaussians can be approximated as the pushforward of a Gaussian through a network of size $O(k)$. As input, the network will use a sample from a standard Gaussian in $\mathbb{R}^{d+1}$. We will subsequently use the first coordinate to implement a "mask" that most of the time masks all but one randomly chosen coordinate in $[k]$. The remaining coordinates are used to produce a sample from each of the components in the Gaussian, and the mask is used to select only one of them.

With this lemma in hand, we finish the Wasserstein lower bound with a standard epsilon-net argument, using the parameter Lipschitzness of the invertible networks by showing the number of "different" invertible neural networks is on the order of $O\left((LR)^{d'}\right)$. This is Lemma 11 in Appendix C. Finally, KL divergence bounds can be derived from the Bobkov-Götze inequality (Bobkov & Götze, 1999), which lower bounds KL divergence by the squared Wasserstein distance. The details are in Appendix C.

7. Conclusion

Normalizing flows are one of the most heavily used generative models across various domains, though we still have a relatively narrow understanding of their relative pros and cons compared to other models. In this paper, we tackled representational aspects of two issues that are frequent sources of training difficulties, depth and conditioning. We hope this work will inspire more theoretical study of fine-grained properties of different generative models.

Acknowledgements. Frederic Koehler was supported in part by NSF CAREER Award CCF-1453261, NSF Large CCF-1565235, Ankur Moitra's ONR Young Investigator Award, and E. Mossel's Vannevar Bush Fellowship ONR-N00014-20-1-2826.

References

Arora, S., Ge, R., Liang, Y., Ma, T., and Zhang, Y. Generalization and Equilibrium in Generative Adversarial Nets (GANs). *arXiv:1703.00573 [cs, stat]*, August 2017. URL http://arxiv.org/abs/1703.00573. arXiv: 1703.00573.

Artin, M. *Algebra*. Pearson, 2011.

Bailey, B. and Telgarsky, M. J. Size-noise tradeoffs in generative networks. In *Advances in Neural Information Processing Systems*, pp. 6489–6499, 2018.

[5] The size of g doesn't indeed depend on ϵ. The weights in the networks will simply grow as ϵ becomes small.

Behrmann, J., Grathwohl, W., Chen, R. T., Duvenaud, D., and Jacobsen, J.-H. Invertible residual networks. *arXiv preprint arXiv:1811.00995*, 2018.

Behrmann, J., Vicol, P., Wang, K.-C., Grosse, R. B., and Jacobsen, J.-H. On the invertibility of invertible neural networks. 2019.

Bobkov, S. G. and Götze, F. Exponential integrability and transportation cost related to logarithmic sobolev inequalities. *Journal of Functional Analysis*, 163(1):1–28, 1999.

Caffarelli, L. A. The regularity of mappings with a convex potential. *Journal of the American Mathematical Society*, 5(1):99–104, 1992.

Devroye, L., Mehrabian, A., and Reddad, T. The total variation distance between high-dimensional gaussians. *arXiv preprint arXiv:1810.08693*, 2018.

Dinh, L., Krueger, D., and Bengio, Y. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Dupont, E., Doucet, A., and Teh, Y. W. Augmented neural odes. In *Advances in Neural Information Processing Systems*, pp. 3134–3144, 2019.

Everitt, B. S. Finite mixture distributions. *Wiley StatsRef: Statistics Reference Online*, 2014.

Gilbert, E. N. A comparison of signalling alphabets. *The Bell system technical journal*, 31(3):504–522, 1952.

Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., and Duvenaud, D. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Grauert, H. and Fritzsche, K. *Several complex variables*, volume 38. Springer Science & Business Media, 2012.

Horn, R. A. and Johnson, C. R. *Matrix analysis*. Cambridge university press, 2012.

Huang, C.-W., Dinh, L., and Courville, A. Augmented normalizing flows: Bridging the gap between generative flows and latent variable models. *arXiv preprint arXiv:2002.07101*, 2020.

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

Kong, Z. and Chaudhuri, K. The expressive power of a class of normalizing flow models. *arXiv preprint arXiv:2006.00392*, 2020.

Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

Montufar, G. and Ay, N. Refinements of universal approximation results for deep belief networks and restricted boltzmann machines. *Neural computation*, 23(5):1306–1319, 2011.

Montúfar, G. F., Rauh, J., and Ay, N. Expressive power and approximation errors of restricted boltzmann machines. In *Advances in neural information processing systems*, pp. 415–423, 2011.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pp. 2338–2347, 2017.

Rao, C. R., Rao, C. R., Statistiker, M., Rao, C. R., and Rao, C. R. *Linear statistical inference and its applications*, volume 2. Wiley New York, 1973.

Rao, S. Lecture notes for cs270: Algorithms. https://people.eecs.berkeley.edu/~satishr/cs270/sp11/rough-notes/measure-concentration.pdf, 2011.

Rödl, V. and Thoma, L. Asymptotic packing and the random greedy algorithm. *Random Structures & Algorithms*, 8 (3):161–177, 1996.

Teshima, T., Ishikawa, I., Tojo, K., Oono, K., Ikeda, M., and Sugiyama, M. Coupling-based invertible neural networks are universal diffeomorphism approximators. In *Advances in Neural Information Processing Systems*, 2020.

Varshamov, R. R. Estimate of the number of signals in error correcting codes. *Docklady Akad. Nauk, SSSR*, 117: 739–741, 1957.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

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

Zhang, F. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

Zhang, H., Gao, X., Unterman, J., and Arodz, T. Approximation capabilities of neural odes and invertible residual networks.