Relaxing Bijectivity Constraints with Continuously Indexed Normalising Flows

Rob Cornish [1] Anthony Caterini [1] George Deligiannidis [1 2] Arnaud Doucet [1]

Abstract

We show that normalising flows become pathological when used to model targets whose supports have complicated topologies. In this scenario, we prove that a flow must become arbitrarily numerically noninvertible in order to approximate the target closely. This result has implications for all flow-based models, and especially *residual flows* (ResFlows), which explicitly control the Lipschitz constant of the bijection used. To address this, we propose *continuously indexed flows* (CIFs), which replace the single bijection used by normalising flows with a continuously indexed family of bijections, and which can intuitively "clean up" mass that would otherwise be misplaced by a single bijection. We show theoretically that CIFs are not subject to the same topological limitations as normalising flows, and obtain better empirical performance on a variety of models and benchmarks.

1 Introduction

Normalising flows (Rezende & Mohamed, 2015) have become popular methods for density estimation (Dinh et al., 2017; Papamakarios et al., 2017; Kingma & Dhariwal, 2018; Chen et al., 2019). These methods model an unknown target distribution $P_X^\star$ on a data space $\mathcal{X} \subseteq \mathbb{R}^d$ as the marginal of X obtained by the generative process

$$Z \sim P_Z, \quad X := f(Z), \tag{1}$$

where P_Z is a *prior* distribution on a space $\mathcal{Z} \subseteq \mathbb{R}^d$, and $f : \mathcal{X} \to \mathcal{Z}$ is a bijection. The use of a bijection means the density of X can be computed analytically by the change-of-variables formula, and the parameters of f can be learned by maximum likelihood using i.i.d. samples from $P_X^\star$.

To be effective, a normalising flow model must specify an expressive family of bijections with tractable Jacobians.

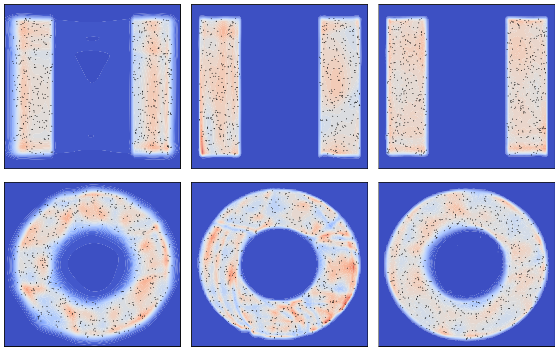

Figure 1: Densities learned by a 10-layer ResFlow (left), 100-layer ResFlow (middle), and 10-layer CIF-ResFlow (right) for two datasets (samples shown in black) that are not homeomorphic to the Gaussian prior. The 10-layer ResFlow visibly leaks mass outside of the support of the target due to its small bi-Lipschitz constant. The larger ResFlow improves on this, but still achieves smaller average log probability than the CIF-ResFlow, as is apparent from the greater homogeneity of the right-hand densities.

Affine coupling layers (Dinh et al., 2015; 2017), autoregressive maps (Germain et al., 2015; Papamakarios et al., 2017), invertible linear transformations (Kingma & Dhariwal, 2018), ODE-based maps (Grathwohl et al., 2019), and invertible ResNet blocks (Behrmann et al., 2019; Chen et al., 2019) are all examples of such bijections that can be composed to produce expressive flows. These models have demonstrated significant promise in their ability to model complex datasets and to synthesise realistic data.

In all these cases, f and f^{-1} are both continuous. It follows that f is a *homeomorphism*, and therefore preserves the topology of its domain (Runde, 2007, Definition 3.3.10). As Dupont et al. (2019) and Dinh et al. (2019) mention, this seems intuitively problematic when P_Z and $P_X^\star$ are supported on domains with distinct topologies, which occurs for example when the supports differ in their number of connected components or "holes", or when they are "knotted" differently. This seems inevitable in practice, as P_Z is usually quite simple (e.g. a Gaussian) while $P_X^\star$ is very complicated (e.g. a distribution over images).

As our first contribution, we make precise the consequences

[1] University of Oxford, Oxford, United Kingdom [2] The Alan Turing Institute, London, United Kingdom. Correspondence to: Rob Cornish <rcornish@robots.ox.ac.uk>.

Proceedings of the 37th International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

of using a topologically misspecified prior. We confirm that in this case it is indeed impossible to recover the target perfectly if f is a homeomorphism. Moreover, in Theorem 2.1 we prove that, in order to *approximate* such a target arbitrarily well, we must have $\operatorname{BiLip} f \to \infty$, where $\operatorname{BiLip} f$ denotes the *bi-Lipschitz constant* of f defined as the infimum over $M \in [1, \infty]$ such that

$$M^{-1}\|z - z'\| \leq \|f(z) - f(z')\| \leq M\|z - z'\| \qquad (2)$$

for all $z, z' \in \mathcal{Z}$. Theorem 2.1 applies essentially regardless of the training objective, and has implications for the case that P_Z and $P_X^\star$ both have full support but are heavily concentrated on regions that are not homeomorphic. Since $\operatorname{BiLip} f$ is a natural measure of the "invertibility" of f (Behrmann et al., 2020), this result shows that the goal of designing neural networks with well-conditioned inverses is fundamentally at odds with the goal of designing neural networks that can approximate complicated densities.

Theorem 2.1 also has immediate implications for *residual flows* (ResFlows) (Behrmann et al., 2019; Chen et al., 2019), which have recently achieved state-of-the-art performance on several large-scale density estimation tasks. Unlike models based on triangular maps (Jaini et al., 2019), ResFlows have the attractive feature that the structure of their Jacobians is unconstrained, which may explain their greater expressiveness. However, as part of the construction, the bi-Lipschitz constant of f is bounded, and so these models must be composed many times in order to achieve overall the large bi-Lipschitz constant required for a complex $P_X^\star$.[1]

To address this problem we introduce *continuously indexed flows* (CIFs), which generalise (1) by replacing the single bijection f with an indexed family of bijections $\{F(\cdot; u)\}_{u \in \mathcal{U}}$, where the index set $\mathcal{U}$ is continuous. Intuitively, CIFs allow mass that would be erroneously placed by a single bijection to be rerouted into a more optimal location. We show that CIFs can learn the support of a given $P_X^\star$ exactly regardless of the topology of the prior, and without the bi-Lipschitz constant of any $F(\cdot; u)$ necessarily becoming infinite. CIFs do not specify the form of F, and can be used in conjunction with any standard normalising flow architecture directly.

Our use of a continuous index overcomes several limitations associated with alternative approaches based on a discrete index (Dinh et al., 2019; Duan, 2019), which suffer either from a discontinuous loss landscape or an intractable computational complexity. However, as a consequence, we sacrifice the ability to compute the likelihood of our model analytically. To address this, we propose a variational approximation that exploits the bijective structure of the model and is suitable for training large-scale models in practice. We empirically evaluate CIFs applied to ResFlows, neural spline flows (NSFs) (Durkan et al., 2019), masked autoregressive flows (MAFs) (Papamakarios et al., 2017), and RealNVPs (Dinh et al., 2017), obtaining improved performance in all cases. We observe a particular benefit for ResFlows: with a 10-layer CIF-ResFlow we surpass the performance of a 100-layer baseline ResFlow and achieve state-of-the-art results on several benchmark datasets.

2 Bi-Lipschitz Constraints on Pushforwards

Normalising flows fall into a larger class of density estimators based on *pushforwards*. Given a prior measure P_Z on $\mathcal{Z}$ and a mapping $f : \mathcal{Z} \to \mathcal{X}$, these models are defined as

$$P_X := f\#P_Z,$$

where the right-hand side denotes a distribution with $f\#P_Z(B) := P_Z(f^{-1}(B))$ for Borel $B \subseteq \mathcal{X}$. Normalising flows take f to be bijective, which under sufficient regularity yields a closed-form expression for the density[2] of P_X (Billingsley, 2008, Theorem 17.2).

Intuitively, the pushforward map f *transports* the mass allocated by P_Z into $\mathcal{X}$-space, thereby defining P_X based on where each unit of mass ends up. This imposes a global constraint on f if P_X is to match perfectly a given target $P_X^\star$. In particular, denote by $\operatorname{supp} P_Z$ the *support* of P_Z. While the precise definition of the support involves topological formalities (see Section B.1 in the Supplement), intuitively this set defines the region of $\mathcal{Z}$ to which P_Z assigns mass. It is then straightforward to show that $P_X = P_X^\star$ only if

$$\operatorname{supp} P_X^\star = \overline{f(\operatorname{supp} P_Z)}, \qquad (3)$$

where $\overline{A}$ denotes the closure of A in $\mathcal{X}$.[3]

The constraint (3) is especially onerous for normalising flows because of their bijectivity. In practice, f and f^{-1} are invariably both continuous, and so f is a *homeomorphism*. Consequently, for these models (3) entails[4]

$$\operatorname{supp} P_X = \operatorname{supp} P_X^\star \text{ only if } \operatorname{supp} P_Z \cong \operatorname{supp} P_X^\star, \qquad (4)$$

where $\mathcal{A} \cong \mathcal{B}$ means that $\mathcal{A}$ and $\mathcal{B}$ are *homeomorphic*, i.e. isomorphic as topological spaces (Runde, 2007, Definition 3.3.10). This means that $\operatorname{supp} P_Z$ and $\operatorname{supp} P_X^\star$ must exactly share *all* topological properties, including number of connected components, number of "holes", the way they are "knotted", etc., in order to learn the target perfectly. Condition (4) therefore suggests that normalising flows are not optimally suited to the task of learning complex real-world densities, where such topological mismatch seems inevitable.

[1] Chen et al. (2019) report using 100-200 layers to learn even simple 2D densities.

[2] Throughout, by "density" we mean Lebesgue density. We will write densities using lowercase, e.g. p_X for the measure P_X.

[3] See Proposition B.3 in the Supplement for a proof.

[4] Note that $\overline{f(\operatorname{supp} P_Z)} = f(\operatorname{supp} P_Z)$ here since $\operatorname{supp} P_Z$ is closed by Proposition B.2 in the Supplement.

However, (4) only rules out the limiting case $P_X = P_X^\star$. In practice it is likely enough to have $P_X \approx P_X^\star$, and it is therefore relevant to consider the implications of a topologically misspecified prior in this case also. Intuitively, this seems to require f become *almost* nonbijective as P_X approaches $P_X^\star$, but it is not immediately clear what this means, or whether this must occur for all models. Likewise, in practice it might be reasonable to assume the density of $P_X^\star$ is everywhere strictly positive. In this case, even if $P_X^\star$ is *concentrated* on some very complicated set, the constraint (4) would trivially be met if P_Z is Gaussian, for example. Nevertheless, it seems that infinitesimal regions of mass should not significantly change the behaviour required of f, and we would therefore like to extend (4) to apply here also.

The bi-Lipschitz constant (2) naturally quantifies the "invertibility" of f. Behrmann et al. (2020) recently showed a relationship between the bi-Lipschitz constant and the *numerical* invertibility of f. If f is injective and differentiable,

$$\operatorname{BiLip} f = \max\left(\sup_{z\in\mathcal{Z}} \|\mathrm{D}f(z)\|_{\mathrm{op}}, \sup_{x\in f(\mathcal{Z})} \|\mathrm{D}f^{-1}(x)\|_{\mathrm{op}} \right),$$

where $\mathrm{D}g(y)$ is the Jacobian of g at y and $\|\cdot\|_{\mathrm{op}}$ is the operator norm. A large bi-Lipschitz constant thus means f or f^{-1} "jumps" somewhere in its domain. More generally, if f is not injective, then $\operatorname{BiLip} f = \infty$, while if $\operatorname{BiLip} f < \infty$, then f is a homeomorphism from $\mathcal{Z}$ to $f(\mathcal{Z})$.[5]

The following theorem shows that if the supports of P_Z and $P_X^\star$ are not homeomorphic, then the bi-Lipschitz constant of f must grow arbitrarily large in order to approximate $P_X^\star$. Here $\xrightarrow{\mathcal{D}}$ denotes weak convergence.

Theorem 2.1. *Suppose P_Z and $P_X^\star$ are probability measures on $\mathbb{R}^{d_\mathcal{Z}}$ and $\mathbb{R}^{d_\mathcal{X}}$ respectively, and that $\operatorname{supp} P_Z \not\simeq \operatorname{supp} P_X^\star$. Then for any sequence of measurable $f_n : \mathbb{R}^{d_\mathcal{Z}} \to \mathbb{R}^{d_\mathcal{X}}$, we can have $f_n \# P_Z \xrightarrow{\mathcal{D}} P_X^\star$ only if*

$$\lim_{n\to\infty} \operatorname{BiLip} f_n = \infty.$$

Weak convergence is implied by the minimisation of all standard statistical divergences used to train generative models, including the KL and Jensen-Shannon divergences and the Wasserstein metric (Arjovsky et al., 2017, Theorem 2). Thus, Theorem 2.1 states that these quantities can vanish only if the bi-Lipschitz constant of the learned mapping becomes arbitrarily large. Likewise, note that we do not assume $d_\mathcal{Z} = d_\mathcal{X}$ so that this result also applies to injective flow models (Kumar et al., 2019), as well as other pushforward-based models such as GANs (Goodfellow et al., 2014).[6]

Theorem 2.1 also applies when $\operatorname{supp} P_Z$ is *almost* not homeomorphic to $\operatorname{supp} P_X^\star$, as is made precise by the following corollary. Here ρ denotes any metric for the weak topology; see Chapter 6 of Villani (2008) for standard examples.

Corollary 2.2. *Suppose P_Z and P_X^0 are probability measures on $\mathbb{R}^{d_\mathcal{Z}}$ and $\mathbb{R}^{d_\mathcal{X}}$ respectively with $\operatorname{supp} P_Z \not\simeq \operatorname{supp} P_X^0$. Then there exists nonincreasing $M : [0,\infty) \to [1,\infty]$ with $M(\epsilon) \to \infty$ as $\epsilon \to 0$ such that, for any probability measure $P_X^\star$ on $\mathbb{R}^{d_\mathcal{X}}$, we have $\operatorname{BiLip} f \geq M(\epsilon)$ whenever $\rho(P_X^\star, P_X^0) \leq \epsilon$ and $\rho(f\#P_Z, P_X^\star) \leq \epsilon$.*

In other words, if the target is close to a probability measure with non-homeomorphic support to that of the prior (i.e. $\rho(P_X^\star, P_X^0)$ is small), and if the model is a good approximation of the target (i.e. $\rho(f\#P_Z, P_X^\star)$ is small), then the Bi-Lipschitz constant of f must be large.

Proofs of these results are in Section B.3 of the Supplement.

2.1 Practical Implications

The results of this section indicate a limitation of existing flow-based density models. This is most direct for *residual flows* (ResFlows) (Behrmann et al., 2019; Chen et al., 2019), which take $f = f_L \circ \cdots \circ f_1$ with each layer of the form

$$f_\ell^{-1}(x) = x + g_\ell(x), \qquad \operatorname{Lip} g_\ell \leq \kappa < 1. \tag{5}$$

Here Lip denotes the Lipschitz constant, which is bounded by a fixed constant κ throughout training. The Lipschitz constraint is enforced by spectral normalisation (Miyato et al., 2018; Gouk et al., 2018) and ensures each f_ℓ is bijective. However, it also follows (Behrmann et al., 2019, Lemma 2) that

$$\operatorname{BiLip} f \leq \max(1+\kappa, (1-\kappa)^{-1})^L < \infty, \tag{6}$$

and Theorem 2.1 thus restricts how well a ResFlow can approximate $P_X^\star$ with non-homeomorphic support to P_Z. Figure 1 illustrates this in practice for simple 2-D examples.

It is possible to relax (6) by taking $\kappa \to 1$. However, this can have a detrimental effect on the variance of the Russian roulette estimator (Kahn, 1955) used by Chen et al. (2019) to compute the Jacobian, and in Section B.4 of the Supplement we give a simple example in which the variance is in fact infinite. Alternatively, we can also loosen the bound (6) by taking $L \to \infty$, and Figure 1 shows that this does indeed lead to better performance. However, greater depth means greater computational cost. In the next section we describe an alternative approach that allows relaxing the bi-Lipschitz constraint of Theorem 2.1 without modifying either κ or L, and thus avoids these potential issues.

Unlike ResFlows, most normalising flows used in practice have an unconstrained bi-Lipschitz constant (Behrmann et al., 2020). As as result, Theorem 2.1 does not prevent

[5] See Section B.2 in the Supplement for proofs.

[6] However, the implications for GANs seem less problematic since a GAN generator is not usually assumed to be bijective.

these models from approximating non-homeomorphic targets arbitrarily well, and indeed several architectures have been proposed that can in principle do so (Huang et al., 2018; Jaini et al., 2019). Nevertheless, the constraint (4) shows that these models still face an underlying limitation in practice, and suggests we may improve performance more generally by relaxing the requirement of bijectivity. We verify empirically in Section 5 that, in addition to ResFlows, our proposed method also yields benefits for flows without an explicit bi-Lipschitz constraint.

Finally, Theorem 2.1 has implications for the numerical stability of normalising flows. It was recently pointed out by Behrmann et al. (2020) that, while having a well-defined mathematical inverse, many common flows can become *numerically* noninvertible over the course of training, leading to low-quality reconstructions and calling into question the accuracy of density values output by the change-of-variables formula. Behrmann et al. (2020) suggest explicitly constraining $\mathrm{BiLip}\, f$ in order to avoid this problem. Theorem 2.1 shows that this involves a fundamental tradeoff against expressivity: if greater numerical stability is required of our normalising flow, then we must necessarily reduce the set of targets we can represent arbitrarily well.

3 Continuously Indexed Flows

In this section we propose *continuously indexed flows* (CIFs) for relaxing the bijectivity of standard normalising flows. We begin by defining the model we consider, and then detail our suggested training and inference procedures. In the next section we discuss advantages over related approaches.

3.1 Model Specification

CIFs are obtained by replacing the single bijection f used by normalising flows with an indexed family $\{F(\cdot; u)\}_{u \in \mathcal{U}}$, where $\mathcal{U} \subseteq \mathbb{R}^{d_\mathcal{U}}$ is our index set and each $F(\cdot; u) : \mathcal{Z} \to \mathcal{X}$ is a bijection. We then define the model P_X as the marginal of X obtained from the following generative process:

$$Z \sim P_Z, \quad U \sim P_{U|Z}(\cdot|Z), \quad X := F(Z; U). \tag{7}$$

Like (1), we assume a prior P_Z on $\mathcal{Z}$, but now also require conditional distributions $P_{U|Z}(\cdot|z)$ on $\mathcal{U}$ for each $z \in \mathcal{Z}$.

We can increase the complexity of (7) by taking P_Z itself to have the same form. This is directly analogous to the standard practice of composing simple bijections to obtain a richer class of normalising flows. In our context, stacking L layers of (7) corresponds to the generative process

$$Z_0 \sim P_{Z_0},\; U_\ell \sim P_{U_\ell|Z_{\ell-1}}(\cdot|Z_{\ell-1}),\; Z_\ell := F_\ell(Z_{\ell-1}; U_\ell), \tag{8}$$

where $\ell \in \{1, \ldots, L\}$. We then take P_X to be the marginal of $X := Z_L$. We have found this construction to improve significantly the expressiveness of our models and make extensive use of it in our experiments below. Note that this corresponds to an instance of (7) where, defining $F^\ell(\cdot; u_1, \ldots, u_\ell) := F_\ell(\cdot; u_\ell) \circ \cdots \circ F_1(\cdot; u_1)$, we take $Z = Z_0$, $U = (U_1, \ldots, U_L)$, $P_{U|Z}(\mathrm{d}u|z) = \prod_\ell P_{U_\ell|Z_{\ell-1}}(\mathrm{d}u_\ell | F^\ell(z; u_1, \ldots, u_\ell))$, and $F = F^L$. We use this to streamline some of the discussion below.

Previous works, most notably RAD (Dinh et al., 2019), have considered related models with a discrete index set $\mathcal{U}$. We instead consider a *continuous* index. In particular, our $\mathcal{U}$ will be an open subset of $\mathbb{R}^{d_\mathcal{U}}$, with each $P_{U|Z}(\cdot|z)$ having a density $p_{U|Z}(\cdot|z)$. A continuous index confers various advantages that we describe in Section 4. The choice also requires a distinct approach to training and inference that we describe in Section 3.2.

We require choices of $p_{U|Z}$ and F for each layer of our model. Straightforward possibilities are

$$F(z; u) = f\left(e^{-s(u)} \odot z - t(u)\right) \tag{9}$$

$$p_{U|Z}(\cdot|z) = \mathrm{Normal}(\mu^p(z), \Sigma^p(z)) \tag{10}$$

for any bijection f (e.g. a ResFlow step) and appropriately defined neural networks s, t, μ^p, and Σ^p.[7] Here the exponential of a vector is meant elementwise, and $\odot$ denotes elementwise multiplication. Note that (9) may be used with all existing normalising flow implementations out-of-the-box. These choices yielded strong empirical results despite their simplicity, but more sophisticated alternatives are certainly possible and may bring improvements in some applications.

3.2 Training and Inference

Heuristically,[8] (7) yields the joint "density"

$$p_{X,U,Z}(x, u, z) := p_Z(z)\, p_{U|Z}(u|z)\, \delta(x - F(z; u)),$$

where p_Z is the density of P_Z and δ is the Dirac delta. If F is sufficiently regular, we can marginalise out the dependence on z by making the change of variable $x' := F(z; u)$, which means $\mathrm{d}z = |\det \mathrm{D}F^{-1}(x'; u)|\, \mathrm{d}x'$.[9] This yields a proper density for (X, U) by integrating over x':

$$\begin{aligned} p_{X,U}(x, u) &:= p_Z(F^{-1}(x; u)) \\ &\times p_{U|Z}(u|F^{-1}(x; u))\, |\det \mathrm{D}F^{-1}(x; u)|. \end{aligned} \tag{11}$$

For an L-layered model, an extension of this argument also gives the following joint density for each $(Z_\ell, U_{1:\ell})$:

$$\begin{aligned} p_{Z_\ell, U_{1:\ell}}(z_\ell, u_{1:\ell}) &:= p_{Z_{\ell-1}, U_{1:\ell-1}}(F_\ell^{-1}(z_\ell; u_\ell), u_{1:\ell-1}) \\ &\times p_{U_\ell|Z_{\ell-1}}(u_\ell | F_\ell^{-1}(z_\ell; u_\ell))|\det \mathrm{D}F_\ell^{-1}(z_\ell; u_\ell)|. \end{aligned} \tag{12}$$

[7] Note this requires $\mathcal{Z} = \mathcal{X} = \mathbb{R}^d$ and $\mathcal{U} = \mathbb{R}^{d_\mathcal{U}}$, i.e. these domains are not strict subsets. We assume this in all our experiments.

[8] We make this rigorous in Section B.5 of the Supplement.

[9] Here $\mathrm{D}F(z; u)$ denotes the Jacobian with respect to z only.

Taking $X := Z_L$ as before we obtain $p_{X,U_{1:L}}$ and hence a density for P_X via

$$p_X(x) := \int p_{X,U_{1:L}}(x, u_{1:L}) \, \mathrm{d}u_{1:L}. \tag{13}$$

Since $\mathcal{U}$ is continuous, this is not analytically tractable. To facilitate likelihood-based training and inference, we make use of a variational scheme that we describe now.

Assuming an L-layered model (8), we introduce an approximate posterior density $q_{U_{1:L}|X} \approx p_{U_{1:L}|X}$ and consider the evidence lower bound (ELBO) of $\log p_X(x)$:

$$\mathcal{L}(x) := \mathbb{E}_{u_{1:L} \sim q_{U_{1:L}|X}(\cdot|x)} \left[\log \frac{p_{X,U_{1:L}}(x, u_{1:L})}{q_{U_{1:L}|X}(u_{1:L}|x)} \right]. \tag{14}$$

It is a standard result that $\mathcal{L}(x) \leq \log p_X(x)$ with equality if and only if $q_{U_{1:L}|X}$ is the exact posterior $p_{U_{1:L}|X}$. This allows learning an approximation to $P_X^\star$ by maximising $\sum_{i=1}^n \mathcal{L}(x_i)$ jointly in $p_{X,U_{1:L}}$ and $q_{U_{1:L}|X}$, where we assume a dataset of n i.i.d. samples $x_i \sim P_X^\star$.

We now consider how to parametrise an effective $q_{U_{1:L}|X}$. Standard approaches to designing inference networks for variational autoencoders (VAEs) (Kingma & Welling, 2014; Rezende et al., 2014; Rezende & Mohamed, 2015; Kingma et al., 2016), while mathematically valid, would not exploit the conditional independencies induced by the bijective structure of (8). We therefore propose a novel inference network that is specifically targeted towards our model, which we compare with existing VAE approaches in Section 4.3. In particular, our $q_{U_{1:L}|X}$ has the following form:

$$q_{U_{1:L}|X}(u_{1:L}|x) := \prod_{\ell=1}^{L} q_{U_\ell|Z_\ell}(u_\ell|z_\ell), \tag{15}$$

with $z_L := x$ and $z_\ell := F_{\ell+1}^{-1}(z_{\ell+1}; u_{\ell+1})$ for $\ell \in \{1, \ldots, L-1\}$, and $q_{U_\ell|Z_\ell}$ can be any parameterised conditional density. We show in Section B.6 of the Supplement that the posterior $p_{U_{1:L}|X}$ factors in the same way as (15), so that we do not lose any generality. Observe also that this scheme shares parameters between $q_{U_{1:L}|X}$ and $p_{X,U_{1:L}}$ in a natural way, since the same F_ℓ are used in both.

We assume each $q_{U_\ell|Z_\ell}$ can be suitably reparametrised (Kingma & Welling, 2014; Rezende et al., 2014) so that, for some function H_ℓ and some density η_ℓ that does not depend on the parameters of $q_{U_{1:L}|Z_\ell}$ and $p_{X,U_{1:L}}$, we have $H_\ell(\epsilon_\ell, z_\ell) \sim q_{U_\ell|Z_\ell}(\cdot|z_\ell)$ when $\epsilon_\ell \sim \eta_\ell$. We can then obtain unbiased estimates of $\mathcal{L}(x)$ using Algorithm 1, which corresponds to a single-sample approximation to the expectation in (14). It is straightforward to see that Algorithm 1 has $\Theta(L)$ complexity. Differentiating through this procedure allows maximising $\sum_{i=1}^n \mathcal{L}(x_i)$ via stochastic gradient descent. At test time, we can also estimate $\log p_X(x)$ directly using importance sampling as described by Rezende et al. (2014, (40)). In particular, letting $\hat{\mathcal{L}}^{(1)}, \ldots, \hat{\mathcal{L}}^{(m)}$ denote the result of separate calls to ELBO(x), we have

$$m^{-1} \operatorname{LogSumExp}(\hat{\mathcal{L}}^{(1)}, \ldots, \hat{\mathcal{L}}^{(m)}) \to \log p_X(x) \tag{16}$$

almost surely as $m \to \infty$.

Algorithm 1 Unbiased estimation of $\mathcal{L}(x)$

```
function ELBO($x$)
    $z_L \leftarrow x$
    $\Delta \leftarrow 0$
    for $\ell = L, \ldots, 1$ do
        $\epsilon \sim \eta_\ell$
        $u \leftarrow H_\ell(\epsilon, z_\ell)$
        $z_{\ell-1} \leftarrow F_\ell^{-1}(z_\ell; u)$
        $\Delta \leftarrow \Delta + \log p_{U_\ell|Z_{\ell-1}}(u|z_{\ell-1}) - \log q_{U_\ell|Z_\ell}(u|z_\ell)$
            $+ \log|\det \mathrm{D}F_\ell^{-1}(z_\ell; u)|$
    end for
    return $\Delta + \log p_{Z_0}(z_0)$
end function
```

In all our experiments we used

$$q_{U_\ell|Z_\ell}(\cdot|z_\ell) = \operatorname{Normal}(\mu_\ell^q(z_\ell), \Sigma_\ell^q(z_\ell)) \tag{17}$$

for appropriate neural networks μ_ℓ^q and Σ_ℓ^q, which is immediately reparameterisable as described e.g. by Kingma & Welling (2014). We found this gave good enough performance that we did not require alternatives such as IAF (Kingma et al., 2016), but such options may also be useful.

Finally, Algorithm 1 requires an expression for $\log|\det \mathrm{D}F_\ell^{-1}(z_\ell; u_\ell)|$. For (9) this is

$$\log\left|\det \mathrm{D}f_\ell^{-1}\left(e^{s_\ell(u_\ell)} \odot (z_\ell + t_\ell(u_\ell))\right)\right| + \sum_{i=1}^{d} [s_\ell(u_\ell)]_i,$$

where $[x]_i$ denotes the i^{th} dimension of x.

4 Comparison with Related Models

4.1 Comparison with Normalising Flows

We now compare CIFs with normalising flows, and in particular describe how CIFs relax the constraints of bijectivity identified in Section 2.

4.1.1 Advantages

Observe that (7) generalises normalising flows: if $F(\cdot; u)$ does not depend on u, then we obtain (1). Moreover, training with the ELBO in this case does not reduce performance compared with training a flow directly, as the following result shows. Here the components of our model F_θ, $p_{U|Z}^\theta$, and $q_{U|X}^\theta$ are parameterised by $\theta \in \Theta$, and for a given choice of parameters θ we will denote by P_X^θ and $\mathcal{L}^\theta$ the corresponding distribution and ELBO (14) respectively.

Proposition 4.1. *Suppose there exists $\phi \in \Theta$ such that, for some bijection $f : \mathcal{Z} \to \mathcal{X}$, $F_\phi(\cdot; u) = f(\cdot)$ for all $u \in \mathcal{U}$. Likewise, suppose $p^\phi_{U|Z}$ and $q^\phi_{U|X}$ are such that, for some density r on $\mathcal{U}$, $p^\phi_{U|Z}(\cdot|z) = q^\phi_{U|X}(\cdot|x) = r(\cdot)$ for all $z \in \mathcal{Z}$ and $x \in \mathcal{X}$. If $\mathbb{E}_{x\sim P^\star_X}[\mathcal{L}^\theta(x)] \geq \mathbb{E}_{x\sim P^\star_X}[\mathcal{L}^\phi(x)]$, then*

$$D_{\mathrm{KL}}\big(P^\star_X \,\big\|\, P^\theta_X\big) \leq D_{\mathrm{KL}}(P^\star_X \,\|\, f\#P_Z).$$

Simply stated, in the limit of infinite data, optimising the ELBO will yield at least as performant a model (as measured by the KL) as any normalising flow our model family can express. The proof is in Section B.7 of the Supplement. In practice, our choices (9), (10), and (17) can easily realise the conditions of Proposition 4.1 by zeroing out the output weights of the neural networks (other than f) involved. Thus, for a given f, we have reason to expect a comparative or better performing model (as measured by average log-likelihood) when trained as a CIF rather than as a normalising flow.

We expect this will in fact lead to *improved* performance because, intuitively, $P_{U|Z}$ can reroute z that would otherwise map outside of $\operatorname{supp} P^\star_X$. To illustrate, fix f in (9) and choose some $z \in \mathcal{Z}$. If $f(z) \in \operatorname{supp} P^\star_X$, then setting $F(z; u) = f(z)$ for all $u \in \mathcal{U}$ as described above ensures $F(z; U) \in \operatorname{supp} P^\star_X$ when $U \sim P_{U|Z}(\cdot|z)$. If conversely $f(z) \notin \operatorname{supp} P^\star_X$, then we *still* have $F(z; U) \in \operatorname{supp} P^\star_X$ almost surely if $P_{U|Z}(\cdot|z)$ is supported on $\{u \in \mathcal{U} : F(z; u) \in \operatorname{supp} P^\star_X\}$. Of course, if f is too simple, then $P_{U|Z}$ must heuristically become very complex in order to obtain this behaviour. This would seem to make inference harder, leading to a looser ELBO (14) and thus overall worse performance after training. We therefore expect CIFs to work well for f that, like the 10-layer ResFlow in Figure 1, can learn a close approximation to the support of the target but "leak" some mass outside of it due to (4) or Theorem 2.1. A CIF can then use $P_{U|Z}$ to "clean up" these small extraneous regions of mass.

We provide empirical support for this argument in Section 5. We also summarise our discussion above with the following precise result. Here ∂A denotes the boundary of a set A.

Proposition 4.2. *If $P^\star_X(\partial \operatorname{supp} P^\star_X) = 0$ and $(z, u) \mapsto F(z; u)$ is jointly continuous with*

$$\overline{F(\operatorname{supp} P_Z \times \mathcal{U})} \supseteq \operatorname{supp} P^\star_X, \tag{18}$$

then there exists $P_{U|Z}$ such that $\operatorname{supp} P_X = \operatorname{supp} P^\star_X$ if and only if, for all $z \in \operatorname{supp} P_Z$, there exists $u \in \mathcal{U}$ with

$$F(z; u) \in \operatorname{supp} P^\star_X. \tag{19}$$

The assumptions here are fairly minimal: the boundary condition ensures $P^\star_X$ is not pathological, and if (18) does not hold, then $D_{\mathrm{KL}}(P^\star_X \,\|\, P_X) = \infty$ for *every* $P_{U|Z}$.[10]

Additionally, the following result gives a sufficient condition under which it is possible to learn the target exactly.

Proposition 4.3. *If $F(z; \cdot) : \mathcal{U} \to \mathcal{X}$ is surjective for each $z \in \mathcal{Z}$, then there exists $P_{U|Z}$ such that $P_X = P^\star_X$.*

See Section B.8 of the Supplement for proofs. These results do not require $\operatorname{supp} P_Z \cong \operatorname{supp} P^\star_X$, thereby showing CIFs relax the constraint (4) for standard normalising flows.

Of course, in practice, our parameterisation (9) does not necessarily ensure that F will satisfy these conditions, and our parameterisation (10) may not be expressive enough to instantiate the $P_{U|Z}$ that is required. However, these results show that CIFs provide at least a *mechanism* for correcting a topologically misspecified prior. When F and $P_{U|Z}$ are sufficiently expressive, we can expect that they will learn to approximate these conditions over the course of training if doing so produces a better density estimate. We therefore anticipate CIFs will improve performance for ResFlows, where Theorem 2.1 applies, and may have benefits more generally, since all flows are ultimately constrained by (4).

4.1.2 Disadvantages

On the other hand, CIFs introduce additional overhead compared with regular normalising flows. It therefore remains to show we obtain better performance on a fixed computational budget, which requires using a smaller model. Empirically this holds for the models and datasets we consider in Section 5, but there are likely cases where it does not, particularly if the topologies of the target and prior are similar.

Likewise, CIFs sacrifice the exactness of normalising flows. We do not see this as a significant problem for the task of density estimation, since the importance sampling estimator (16) means that at test time we can obtain arbitrary accuracy by taking m to be large. However, the lack of a closed-form density does limit the use of CIFs in some downstream tasks. In particular, CIFs cannot immediately be plugged in to a variational approximation in the manner of Rezende & Mohamed (2015), since this requires exact likelihoods. However, it may be possible to use CIFs in the context of an extended-space variational framework along the lines of Agakov & Barber (2004), and we leave this for future work.

4.2 Comparison with Discretely Indexed Models

Similar models to CIFs have been proposed that use a discrete index space. In the context of Bayesian inference, Duan (2019) proposes a single-layer ($L = 1$) model consisting of (7) with $\mathcal{U} = \{1, \ldots, I\}$ and $F(\cdot; i) = f_i$ for separate normalising flows $f_1, \ldots, f_I$. A special case of this framework is given by *deep Gaussian mixture models* (Van den Oord & Schrauwen, 2014; van den Oord & Dambre, 2015), which corresponds to using invertible linear transformations for each f_i. In this case, (13) becomes a summation that can

[10] See Proposition B.1 and Proposition B.3 in the Supplement.

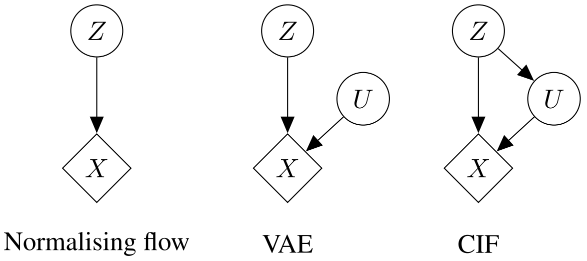

Figure 2: Comparison of related generative models. Circular nodes are random and diamond nodes are deterministic. CIFs generalise both normalising flows and VAEs as shown.

be computed analytically. However, this quickly becomes intractable as L grows larger, since the cost to compute this is seen to be $\Theta(I^L)$. Unlike for a continuous u, this cannot easily be reduced to $\Theta(L)$ using a variational approximation as in Section 3.2, since a discrete $q_{U|X}$ is not amenable to the reparameterisation trick. In addition, the use of separate bijections also means that the number of parameters of the model grows as I increases. In contrast, a continuous index allows a natural mechanism for sharing parameters across different $F(\cdot; u)$ as in (9).

Prior to Duan (2019), Dinh et al. (2019) proposed RAD as a means to mitigate the $\Theta(I^L)$ cost of naïvely stacking discrete layers. RAD partitions $\mathcal{X}$ into I disjoint subsets $B_1, \ldots, B_I$ and defines bijections $f_i : \mathcal{Z} \to B_i$ for each i. The model is then taken to be the marginal of X in

$$Z \sim P_Z, \quad U \sim P_{U|Z}(\cdot|Z), \quad X := f_U(Z),$$

where each $P_{U|Z}(\cdot|z)$ is a discrete distribution on $\{1, \ldots, I\}$. Note that this is not an instance of our model (7), since we require each $F(\cdot; u)$ to be surjective onto $\mathcal{X}$. The use of partitioning means that (13) is a summation with only a single term, which reduces the cost for L layers to $\Theta(L)$. However, partitioning also makes p_X discontinuous. This leads to a very difficult optimisation problem and Dinh et al. (2019) only report results for simple 2-D densities. Additionally, partitioning requires ad-hoc architectural changes to existing normalising flows, and does not directly address the increasing parameter cost as I grows large.

4.3 Comparison with Variational Autoencoders

CIFs also generalise a broad family of variational autoencoders (VAEs) (Kingma & Welling, 2014; Rezende et al., 2014). Recall that VAEs take

$$p_X(x) := \int p_U(u) p_{X|U}(x|u) \,\mathrm{d}u \tag{20}$$

for some choices of densities p_U and $p_{X|U}$.[11] For instance, a mean-field Gaussian observation density has

$$p_{X|U}(\cdot|u) := \text{Normal}\left(t(u), \text{diag}\left(e^{s(u)}\right)\right),$$

where $t, s : \mathcal{U} \to \mathcal{X}$, and $\text{diag}(v)$ denotes the matrix with diagonal $v \in \mathbb{R}^d$ and zeros elsewhere. If P_Z is a standard Gaussian, if each $P_{U|Z}(\cdot|z)$ has independent density p_U, and if F is (9) with f the identity, then it follows that (7) has marginal density (20) (modulo the signs of s and t).[12]

More generally, every VAE model (20) with each $p_{X|U}(\cdot|u)$ strictly positive corresponds to an instance of (7) where U is sampled independently of Z. To see this, let p_Z be any strictly positive density on $\mathcal{Z}$, and let each $F(\cdot; u)$ be the Knothe-Rosenblatt coupling (Villani, 2008) of p_Z and $p_{X|U}(\cdot|u)$. By construction each $F(\cdot; u)$ is invertible and gives $F(Z; u) \sim p_{X|U}(\cdot|u)$ when $Z \sim p_Z$. As a result, (7) again yields X with a marginal density defined by (20). Consequently, CIFs generalise the VAE framework by adding an additional edge in the graphical model as shown in Figure 2.

On the other hand, CIFs differ from VAEs in the way they are composed. Whereas CIFs stack by taking p_Z to be a CIF, VAEs are typically stacked by taking p_U to be a VAE (Rezende et al., 2014; Kingma et al., 2014; Burda et al., 2016; Sønderby et al., 2016). This has implications for the design of the inference network $q_{U_{1:L}|X}$. In particular, a hierarchical VAE obtained in this way is *Markovian*, so that

$$p_{U_{1:L}|X}(x, u_{1:L}) = p_{U_L|X}(u_L|x) \prod_{\ell=1}^{L} p_{U_\ell|U_{\ell-1}}(u_\ell|u_{\ell-1})$$

where L is the number of layers. This directly allows specifying $q_{U_{1:L}|X}$ to be of the same form without any loss of generality (Kingma et al., 2014; Burda et al., 2016; Sønderby et al., 2016). Conversely, CIFs do not factor in this way, which motivates our alternative approach in Section 3.2.

Note finally that CIFs should not be conflated with the large class of methods that use normalising flows to improve the *inference* procedure in VAEs (Rezende & Mohamed, 2015; Kingma et al., 2016; van den Berg et al., 2018). These approaches are orthogonal to ours and indeed may be useful for improving our own inference procedure by replacing (17) with a more expressive model.

4.4 Other Related Work

Additional related methods have been proposed. Within a classification context, Dupont et al. (2019) identify topological problems related to ODE-based mappings (Chen et al.,

[11] Note that this notation is nonstandard for VAEs in order to align with the rest of the paper. Here our U corresponds to z as used by Kingma & Welling (2014).

[12] Here Z corresponds to ϵ as used by Kingma & Welling (2014).

2018), which like normalising flows are homeomorphisms and hence preserve the topology of their input. To avoid this, Dupont et al. (2019) propose augmenting the data by appending auxiliary dimensions and learning a new mapping on this space. In contrast, CIFs may be understood as augmenting not the data but instead the *model* by considering a family of individual bijections on the *original* space.

In addition, Ho et al. (2019) use a variational scheme to improve on the standard dequantisation method proposed by Theis et al. (2016) for modelling image datasets with normalising flows. This approach is potentially complementary to CIFs, but we do not make use of it in our experiments.

5 Experiments

We evaluated the performance of CIFs on several problems of varying difficulty, including synthetic 2-D data, several tabular datasets, and three image datasets. In all cases we took $\mathcal{Z} = \mathcal{X} = \mathbb{R}^d$ with d the dimension of the dataset. We used the stacked architecture (8) with the prior P_{Z_0} a Gaussian. At each layer, F had form (9) with f a primitive flow step from a baseline architecture (e.g. a single residual block for ResFlow). Each $p_{U|Z}$ and $q_{U|X}$ had form (10) and (17) respectively. We provide an overview of our results for the tabular and image datasets here. Full experimental details, including additional 2-D figures along the lines of Figure 1, are in Section C of the Supplement. See `github.com/jrmcornish/cif` for our code.

5.1 Tabular Datasets

We tested the performance of CIFs on the tabular datasets used by Papamakarios et al. (2017). For each dataset, we trained 10 and 100-layer baseline fully connected ResFlows, and corresponding 10-layer CIF-ResFlows. The CIF-ResFlows had roughly 1.5-4.5% more parameters (depending on the dimension of the dataset) than the otherwise identical 10-layer ResFlows, and roughly 10% of the parameters of the 100-layer ResFlows. Table 1 reports the average log-probability of the test set that we obtained for each model. Observe that in all cases CIF-ResFlows significantly outperform both baseline models. Moreover, for all but GAS, the CIF-ResFlows achieve state-of-the-art performance based on the results reported by Durkan et al. (2019, Table 1). This is particularly noticeable for POWER and BSDS300, where CIF-ResFlow improves on the best results of Durkan et al. (2019) by 0.94 and 2.77 nats respectively.

We additionally tried using *masked autoregressive flows* (MAFs) (Papamakarios et al., 2017) and *neural spline flows* (NSFs) (Durkan et al., 2019) for f. In each case, we closely match the experimental settings of the baselines and augment using CIFs, controlling for the number of parameters used by the CIF extensions. Table 1 reports the average log-probability across the test set for each experiment. Here, CIF-NSF-1 is a CIF with the same number of parameters as the baseline, and CIF-NSF-2 is a model using a baseline configuration for f (but having more parameters overall). We see that CIF-MAFs consistently outperform MAFs across datasets; CIF-NSFs do not improve upon NSFs as dramatically, although we still notice improvements and would expect to improve further with more hyperparameter tuning. Lastly it is important to notice that MAFs and NSFs do not restrict the Lipschitz constant of f. These results show that CIFs can yield benefits for normalising flows even if Theorem 2.1 is not directly a limitation.

Finally, for ablation purposes we tried taking f to be the identity. We obtained consistently worse performance than for CIF-ResFlows and CIF-MAF in this case, which aligns with our conjecture in Section 4.1.1 that a performant CIF requires an expressive base flow f. Details and results are given in Section C.1.4 of the Supplement.

5.2 Image Datasets

We also considered CIFs applied to the MNIST (LeCun, 1998), Fashion-MNIST (Xiao et al., 2017), and CIFAR-10 (Krizhevsky & Hinton, 2009) datasets. Following our tabular experiments, we trained a multi-scale convolutional ResFlow and a corresponding CIF-ResFlow, as well as a larger baseline ResFlow to account for the additional parameters and depth introduced by our method. Note that these models were significantly smaller than those used by Chen et al. (2019): e.g. for CIFAR10, the ResFlow used by Chen et al. (2019) had 25M parameters, while our two baseline ResFlows and our CIF-ResFlow had 2.4M, 6.2M, and 5.6M parameters respectively. We likewise considered RealNVPs with the same multi-scale convolutional architecture used by Dinh et al. (2017) for their CIFAR-10 experiments. For these runs we trained baseline RealNVPs, corresponding CIF-RealNVPs, and larger baseline RealNVPs with more depth and parameters.

The results are given in Table 2 and Table 3. Observe CIFs outperformed the baseline models for all datasets, which shows that our approach can scale to high dimensions. For the CIF-ResFlows, we also obtained better performance than Chen et al. (2019) on MNIST and better performance than Glow (Kingma & Dhariwal, 2018) on CIFAR10, despite using a much smaller model. Samples from all models are shown in Section C.2 of the Supplement.

6 Conclusion and Future Work

The constraint (4) shows that normalising flows are unable to exactly model targets whose topology differs from that

[13] Only one seed was used per run due to computational limitations. However, the results were not cherry-picked.

Table 1: Mean ± standard error (over 3 seeds) of average test set log-likelihood (in nats). Higher is better. Best performing runs for each group are shown in bold. A $\star$ indicates state-of-the-art performance according to Durkan et al. (2019, Table 1).

	Power	Gas	Hepmass	Miniboone	BSDS300
ResFlow ($L = 10$)	-2.73 ± 0.03	4.16 ± 0.08	-20.68 ± 0.02	-14.2 ± 0.10	123.51 ± 0.09
ResFlow ($L = 100$)	0.48 ± 0.00	10.57 ± 0.17	-16.67 ± 0.05	-11.16 ± 0.04	148.05 ± 0.61
CIF-ResFlow ($L = 10$)	$\mathbf{1.60 \pm 0.21}^\star$	$\mathbf{12.12 \pm 0.10}$	$\mathbf{-13.74 \pm 0.03}^\star$	$\mathbf{-8.10 \pm 0.04}^\star$	$\mathbf{160.50 \pm 0.08}^\star$
MAF	0.19 ± 0.02	9.23 ± 0.07	-18.33 ± 0.10	-10.98 ± 0.03	156.13 ± 0.00
CIF-MAF	$\mathbf{0.48 \pm 0.01}$	$\mathbf{12.02 \pm 0.10}$	$\mathbf{-16.63 \pm 0.09}$	$\mathbf{-9.93 \pm 0.04}$	$\mathbf{156.67 \pm 0.02}$
NSF	$\mathbf{0.69 \pm 0.00}$	13.01 ± 0.02	-14.30 ± 0.05	-10.68 ± 0.06	$\mathbf{157.59 \pm 0.02}$
CIF-NSF-1	$\mathbf{0.68 \pm 0.01}$	12.94 ± 0.01	$\mathbf{-13.83 \pm 0.10}$	$\mathbf{-9.93 \pm 0.06}$	$\mathbf{157.60 \pm 0.02}$
CIF-NSF-2	$\mathbf{0.69 \pm 0.00}$	$\mathbf{13.08 \pm 0.00}$	-14.18 ± 0.09	-10.80 ± 0.01	157.56 ± 0.02

Table 2: Average test bits per dimension.[13] Lower is better.

	MNIST	CIFAR-10
ResFlow (small)	1.074	3.474
ResFlow (big)	1.018	3.422
CIF-ResFlow	**0.922**	**3.334**

Table 3: Mean ± standard error of average test set bits per dimension over 3 random seeds. Lower is better.

	Fashion-MNIST	CIFAR-10
RealNVP (small)	2.944 ± 0.003	3.565 ± 0.001
RealNVP (big)	2.946 ± 0.002	3.554 ± 0.001
CIF-RealNVP	$\mathbf{2.823 \pm 0.003}$	$\mathbf{3.477 \pm 0.019}$

of the prior. Moreover, in order to approximate such targets closely, Theorem 2.1 shows that the bi-Lipschitz constant of a flow must become arbitrarily large. To address these problems, we have proposed CIFs, which can "clean up" regions of mass that are placed outside the support of the target by a standard flow. CIFs perform well in practice and outperform baseline flows on several benchmark datasets.

While we have focussed on the use of CIFs for density estimation in this paper, it would also be interesting to apply CIFs in other contexts where normalising flows have been used successfully. As CIFs do not have an analytically available density, this would likely require the modification of existing numerical frameworks, but the expressiveness benefits provided by CIFs might make this additional effort worthwhile. We leave this direction for future work.

Acknowledgements

Rob Cornish is supported by the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines & Systems (EP/L015897/1) and NVIDIA. Anthony Caterini is a Commonwealth Scholar supported by the U.K. Government. Arnaud Doucet is partially supported by the U.S. Army Research Laboratory, the U.S. Army Research Office, and by the U.K. Ministry of Defence (MoD) grant EP/R013616/1 and the U.K. EPSRC under grant numbers EP/R034710/1 and EP/R018561/1.

References

Agakov, F. V. and Barber, D. An auxiliary variational method. In *International Conference on Neural Information Processing*, pp. 561–566. Springer, 2004.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223, 2017.

Bache, K. and Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Beatson, A. and Adams, R. P. Efficient optimization of loops and limits with randomized telescoping sums. In *International Conference on Machine Learning*, pp. 534–543, 2019.

Behrmann, J., Grathwohl, W., Chen, R. T., Duvenaud, D., and Jacobsen, J.-H. Invertible residual networks. In *International Conference on Machine Learning*, pp. 573–582, 2019.

Behrmann, J., Vicol, P., Wang, K.-C., Grosse, R. B., and Jacobsen, J.-H. On the invertibility of invertible neural networks, 2020. URL https://openreview.net/forum?id=BJlVeyHFwH.

Billingsley, P. *Probability and Measure*. John Wiley & Sons, 2008.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. In *ICLR*, 2016.

Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pp. 6571–6583, 2018.

Chen, T. Q., Behrmann, J., Duvenaud, D. K., and Jacobsen, J.-H. Residual flows for invertible generative modeling. In *Advances in Neural Information Processing Systems*, pp. 9913–9923, 2019.

Dinh, L., Krueger, D., and Bengio, Y. Nice: Non-linear independent components estimation. In *ICLR Workshop*, 2015.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *ICLR*, 2017.

Dinh, L., Sohl-Dickstein, J., Pascanu, R., and Larochelle, H. A RAD approach to deep mixture models. In *ICLR Workshop*, 2019.

Duan, L. L. Transport Monte Carlo. *arXiv preprint arXiv:1907.10448*, 2019.

Dudley, R. M. *Real Analysis and Probability*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2 edition, 2002.

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

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. In *Advances in Neural Information Processing Systems*, pp. 7509–7520, 2019.

Germain, M., Gregor, K., Murray, I., and Larochelle, H. Made: Masked autoencoder for distribution estimation. In *International Conference on Machine Learning*, pp. 881–889, 2015.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Gouk, H., Frank, E., Pfahringer, B., and Cree, M. Regularisation of neural networks by enforcing lipschitz continuity. *arXiv preprint arXiv:1804.04368*, 2018.

Grathwohl, W., Chen, R. T., Betterncourt, J., Sutskever, I., and Duvenaud, D. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *ICLR*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pp. 630–645. Springer, 2016b.

Ho, J., Chen, X., Srinivas, A., Duan, Y., and Abbeel, P. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, pp. 2722–2730, 2019.

Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. Neural autoregressive flows. In *International Conference on Machine Learning*, pp. 2083–2092, 2018.

Hutchinson, M. F. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 19(2):433–450, 1990.

Jaini, P., Selby, K. A., and Yu, Y. Sum-of-squares polynomial flow. In *International Conference on Machine Learning*, pp. 3009–3018, 2019.

Kahn, H. Use of different Monte Carlo sampling techniques. Technical report, Rand Corporation, 1955.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

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

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *ICLR*, 2014.

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pp. 3581–3589, 2014.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pp. 4743–4751, 2016.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Kumar, A., Poole, B., and Murphy, K. Learning generative samplers using relaxed injective flow. In *ICML Workshop on Invertible Neural Nets and Normalizing Flows*, 2019.

LeCun, Y. The MNIST database of handwritten digits, 1998. URL http://yann.lecun.com/exdb/mnist/.

Lyne, A.-M., Girolami, M., Atchadé, Y., Strathmann, H., Simpson, D., et al. On Russian roulette estimates for Bayesian inference with doubly-intractable likelihoods. *Statistical science*, 30(4):443–467, 2015.

Martin, D., Fowlkes, C., Tal, D., and Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pp. 416–423. IEEE, 2001.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

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

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538, 2015.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pp. 1278–1286, 2014.

Rhee, C.-h. and Glynn, P. W. Unbiased estimation with square root convergence for SDE models. *Operations Research*, 63(5):1026–1043, 2015.

Rudin, W. *Principles of Mathematical Analysis*, volume 3. McGraw-hill New York, 1964.

Rudin, W. *Real and Complex Analysis*. Tata McGraw-hill education, 2006.

Runde, V. *A Taste of Topology*. Springer, 2007.

Skilling, J. The eigenvalues of mega-dimensional matrices. In *Maximum Entropy and Bayesian Methods*, pp. 455–466. Springer, 1989.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. Ladder variational autoencoders. In *Advances in Neural Information Processing Systems*, pp. 3738–3746, 2016.

Theis, L., Oord, A. v. d., and Bethge, M. A note on the evaluation of generative models. In *ICLR*, 2016.

van den Berg, R., Hasenclever, L., Tomczak, J. M., and Welling, M. Sylvester normalizing flows for variational inference. In *UAI 2018: The Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 393–402, 2018.

van den Oord, A. and Dambre, J. Locally-connected transformations for deep GMMs. In *International Conference on Machine Learning (ICML): Deep learning Workshop*, pp. 1–8, 2015.

Van den Oord, A. and Schrauwen, B. Factoring variations in natural images with deep Gaussian mixture models. In *Advances in Neural Information Processing Systems*, pp. 3518–3526, 2014.

Villani, C. *Optimal Transport: Old and New*, volume 338. Springer Science & Business Media, 2008.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.