Variational Inference with Normalizing Flows

Danilo Jimenez Rezende
Shakir Mohamed
Google DeepMind, London

DANILOR@GOOGLE.COM
SHAKIR@GOOGLE.COM

Abstract

The choice of approximate posterior distribution is one of the core problems in variational inference. Most applications of variational inference employ simple families of posterior approximations in order to allow for efficient inference, focusing on mean-field or other simple structured approximations. This restriction has a significant impact on the quality of inferences made using variational methods. We introduce a new approach for specifying flexible, arbitrarily complex and scalable approximate posterior distributions. Our approximations are distributions constructed through a normalizing flow, whereby a simple initial density is transformed into a more complex one by applying a sequence of invertible transformations until a desired level of complexity is attained. We use this view of normalizing flows to develop categories of finite and infinitesimal flows and provide a unified view of approaches for constructing rich posterior approximations. We demonstrate that the theoretical advantages of having posteriors that better match the true posterior, combined with the scalability of amortized variational approaches, provides a clear improvement in performance and applicability of variational inference.

1. Introduction

There has been a great deal of renewed interest in variational inference as a means of scaling probabilistic modeling to increasingly complex problems on increasingly larger data sets. Variational inference now lies at the core of large-scale topic models of text (Hoffman et al., 2013), provides the state-of-the-art in semi-supervised classification (Kingma et al., 2014), drives the models that currently produce the most realistic generative models of images (Gregor et al., 2014; 2015; Rezende et al., 2014; Kingma & Welling, 2014), and are a default tool for the understanding

of many physical and chemical systems. Despite these successes and ongoing advances, there are a number of disadvantages of variational methods that limit their power and hamper their wider adoption as a default method for statistical inference. It is one of these limitations, the choice of posterior approximation, that we address in this paper.

Variational inference requires that intractable posterior distributions be approximated by a class of known probability distributions, over which we search for the best approximation to the true posterior. The class of approximations used is often limited, e.g., mean-field approximations, implying that no solution is ever able to resemble the true posterior distribution. This is a widely raised objection to variational methods, in that unlike other inferential methods such as MCMC, even in the asymptotic regime we are unable recover the true posterior distribution.

There is much evidence that richer, more faithful posterior approximations do result in better performance. For example, when compared to sigmoid belief networks that make use of mean-field approximations, deep auto-regressive networks use a posterior approximation with an auto-regressive dependency structure that provides a clear improvement in performance (Mnih & Gregor, 2014). There is also a large body of evidence that describes the detrimental effect of limited posterior approximations. Turner & Sahani (2011) provide an exposition of two commonly experienced problems. The first is the widely-observed problem of under-estimation of the variance of the posterior distribution, which can result in poor predictions and unreliable decisions based on the chosen posterior approximation. The second is that the limited capacity of the posterior approximation can also result in biases in the MAP estimates of any model parameters (and this is the case e.g., in time-series models).

A number of proposals for rich posterior approximations have been explored, typically based on structured mean-field approximations that incorporate some basic form of dependency within the approximate posterior. Another potentially powerful alternative would be to specify the approximate posterior as a mixture model, such as those developed by Jaakkola & Jordan (1998); Jordan et al. (1999); Gershman et al. (2012). But the mixture approach limits

the potential scalability of variational inference since it requires evaluation of the log-likelihood and its gradients for each mixture component per parameter update, which is typically computationally expensive.

This paper presents a new approach for specifying approximate posterior distributions for variational inference. We begin by reviewing the current best practice for inference in general directed graphical models, based on amortized variational inference and efficient Monte Carlo gradient estimation, in section 2. We then make the following contributions:

- We propose the specification of approximate posterior distributions using normalizing flows, a tool for constructing complex distributions by transforming a probability density through a series of invertible mappings (sect. 3). Inference with normalizing flows provides a tighter, modified variational lower bound with additional terms that only add terms with linear time complexity (sect 4).

- We show that normalizing flows admit infinitesimal flows that allow us to specify a class of posterior approximations that in the asymptotic regime is able to recover the true posterior distribution, overcoming one oft-quoted limitation of variational inference.

- We present a unified view of related approaches for improved posterior approximation as the application of special types of normalizing flows (sect 5).

- We show experimentally that the use of general normalizing flows systematically outperforms other competing approaches for posterior approximation.

2. Amortized Variational Inference

To perform inference it is sufficient to reason using the marginal likelihood of a probabilistic model, and requires the marginalization of any missing or latent variables in the model. This integration is typically intractable, and instead, we optimize a lower bound on the marginal likelihood. Consider a general probabilistic model with observations $\mathbf{x}$, latent variables $\mathbf{z}$ over which we must integrate, and model parameters $\boldsymbol{\theta}$. We introduce an approximate posterior distribution for the latent variables $q_\phi(\mathbf{z}|\mathbf{x})$ and follow the variational principle (Jordan et al., 1999) to obtain a bound on the marginal likelihood:

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \tag{1}$$

$$= \log \int \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \tag{2}$$

$$\geq -\mathbb{D}_{\mathrm{KL}}[q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})] + \mathbb{E}_q[\log p_\theta(\mathbf{x}|\mathbf{z})] = -\mathcal{F}(\mathbf{x}), \tag{3}$$

where we used Jensen's inequality to obtain the final equation, $p_\theta(\mathbf{x}|\mathbf{z})$ is a likelihood function and $p(\mathbf{z})$ is a prior over the latent variables. We can easily extend this formulation to posterior inference over the parameters $\boldsymbol{\theta}$, but

we will focus on inference over the latent variables only. This bound is often referred to as the negative free energy $\mathcal{F}$ or as the evidence lower bound (ELBO). It consists of two terms: the first is the KL divergence between the approximate posterior and the prior distribution (which acts as a regularizer), and the second is a reconstruction error. This bound (3) provides a unified objective function for optimization of both the parameters $\boldsymbol{\theta}$ and ϕ of the model and variational approximation, respectively.

Current best practice in variational inference performs this optimization using mini-batches and stochastic gradient descent, which is what allows variational inference to be scaled to problems with very large data sets. There are two problems that must be addressed to successfully use the variational approach: 1) efficient computation of the derivatives of the expected log-likelihood $\nabla_\phi \mathbb{E}_{q_\phi(z)}[\log p_\theta(\mathbf{x}|\mathbf{z})]$, and 2) choosing the richest, computationally-feasible approximate posterior distribution $q(\cdot)$. The second problem is the focus of this paper. To address the first problem, we make use of two tools: Monte Carlo gradient estimation and inference networks, which when used together is what we refer to as *amortized variational inference*.

2.1. Stochastic Backpropagation

The bulk of research in variational inference over the years has been on ways in which to compute the gradient of the expected log-likelihood $\nabla_\phi \mathbb{E}_{q_\phi(z)}[\log p(\mathbf{x}|\mathbf{z})]$. Whereas we would have previously resorted to local variational methods (Bishop, 2006), in general we now always compute such expectations using Monte Carlo approximations (including the KL term in the bound, if it is not analytically known). This forms what has been aptly named doubly-stochastic estimation (Titsias & Lazaro-Gredilla, 2014), since we have one source of stochasticity from the mini-batch and a second from the Monte Carlo approximation of the expectation.

We focus on models with continuous latent variables, and the approach we take computes the required gradients using a non-centered reparameterization of the expectation (Papaspiliopoulos et al., 2003; Williams, 1992), combined with Monte Carlo approximation — referred to as *stochastic backpropagation* (Rezende et al., 2014). This approach has also been referred to or as *stochastic gradient variational Bayes* (SGVB) (Kingma & Welling, 2014) or as affine variational inference (Challis & Barber, 2012).

Stochastic backpropagation involves two steps:

- **Reparameterization**. We reparameterize the latent variable in terms of a known base distribution and a differentiable transformation (such as a location-scale transformation or cumulative distribution function). For example, if $q_\phi(z)$ is a Gaussian distribution $\mathcal{N}(z|\mu, \sigma^2)$, with $\phi = \{\mu, \sigma^2\}$, then the location-scale

transformation using the standard Normal as a base distribution allows us to reparameterize $\mathbf{z}$ as:

$$z \sim \mathcal{N}(z|\mu, \sigma^2) \Leftrightarrow z = \mu + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1)$$

- **Backpropagation with Monte Carlo**. We can now differentiate (backpropagation) w.r.t. the parameters ϕ of the variational distribution using a Monte Carlo approximation with draws from the base distribution:

$$\nabla_\phi \mathbb{E}_{q_\phi(z)}[f_\theta(z)] \Leftrightarrow \mathbb{E}_{\mathcal{N}(\epsilon|0,1)}[\nabla_\phi f_\theta(\mu + \sigma\epsilon)].$$

A number of general purpose approaches based on Monte Carlo control variate (MCCV) estimators exist as an alternative to stochastic backpropagation, and allow for gradient computation with latent variables that may be continuous *or* discrete (Williams, 1992; Mnih & Gregor, 2014; Ranganath et al., 2013; Wingate & Weber, 2013). An important advantage of stochastic backpropagation is that, for models with continuous latent variables, it has the lowest variance among competing estimators.

2.2. Inference Networks

A second important practice is that the approximate posterior distribution $q_\phi(\cdot)$ is represented using a recognition model or inference network (Rezende et al., 2014; Dayan, 2000; Gershman & Goodman, 2014; Kingma & Welling, 2014). An inference network is a model that learns an inverse map from observations to latent variables. Using an inference network, we avoid the need to compute per data point variational parameters, but can instead compute a set of global variational parameters ϕ valid for inference at both training and test time. This allows us to amortize the cost of inference by generalizing between the posterior estimates for all latent variables through the parameters of the inference network. The simplest inference models that we can use are diagonal Gaussian densities, $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}_\phi^2(\mathbf{x})))$, where the mean function $\boldsymbol{\mu}_\phi(\mathbf{x})$ and the standard-deviation function $\boldsymbol{\sigma}_\phi(\mathbf{x})$ are specified using deep neural networks.

2.3. Deep Latent Gaussian Models

In this paper, we study deep latent Gaussian models (DLGM), which are a general class of deep directed graphical models that consist of a hierarchy of L layers of Gaussian latent variables $\mathbf{z}_l$ for layer l. Each layer of latent variables is dependent on the layer above in a non-linear way, and for DLGMs, this non-linear dependency is specified by deep neural networks. The joint probability model is:

$$p(\mathbf{x}, \mathbf{z}_1, \ldots, \mathbf{z}_L) = p(\mathbf{x}|f_0(\mathbf{z}_1)) \prod_{l=1}^{L} p(\mathbf{z}_l|f_l(\mathbf{z}_{l+1})) \quad (4)$$

where the Lth Gaussian distribution is not dependent on any other random variables. The prior over latent variables is a unit Gaussian $p(\mathbf{z}_l) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the observation likelihood $p_\theta(\mathbf{x}|\mathbf{z})$ is any appropriate distribution that

is conditioned on $\mathbf{z}_1$ and is also parameterized by a deep neural network (figure 2). This model class is very general and includes other models such as factor analysis and PCA, non-linear factor analysis, and non-linear Gaussian belief networks as special cases (Rezende et al., 2014).

DLGMs use continuous latent variables and is a model class perfectly suited to fast amortized variational inference using the lower bound (3) and stochastic backpropagation. The end-to-end system of DLGM and inference network can be viewed as an encoder-decoder architecture, and this is the perspective taken by Kingma & Welling (2014) who present this combination of model and inference strategy as a variational auto-encoder. The inference networks used in Kingma & Welling (2014); Rezende et al. (2014) are simple diagonal or diagonal-plus-low rank Gaussian distributions. The true posterior distribution will be more complex than this assumption allows for, and defining multimodal and constrained posterior approximations in a scalable manner remains a significant open problem in variational inference.

3. Normalizing Flows

By examining the bound (3), we can see that the optimal variational distribution that allows $\mathbb{D}_{\mathrm{KL}}[q\|p] = 0$ is one for which $q_\phi(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z}|\mathbf{x})$, i.e. q matches the true posterior distribution. This possibility is obviously not realizable given the typically used $q(\cdot)$ distributions, such as independent Gaussians or other mean-field approximations. Indeed, one limitation of the variational methodology due to the available choices of approximating families, is that even in an asymptotic regime we can not obtain the true posterior. Thus, an ideal family of variational distributions $q_\phi(\mathbf{z}|\mathbf{x})$ is one that is highly flexible, preferably flexible enough to contain the true posterior as one solution. One path towards this ideal is based on the principle of normalizing flows (Tabak & Turner, 2013; Tabak & Vanden-Eijnden, 2010).

A *normalizing flow* describes the transformation of a probability density through a sequence of invertible mappings. By repeatedly applying the rule for change of variables, the initial density 'flows' through the sequence of invertible mappings. At the end of this sequence we obtain a valid probability distribution and hence this type of flow is referred to as a normalizing flow.

3.1. Finite Flows

The basic rule for transformation of densities considers an invertible, smooth mapping $f : \mathbb{R}^d \to \mathbb{R}^d$ with inverse $f^{-1} = g$, i.e. the composition $g \circ f(\mathbf{z}) = \mathbf{z}$. If we use this mapping to transform a random variable $\mathbf{z}$ with distribution $q(\mathbf{z})$, the resulting random variable $\mathbf{z}' = f(\mathbf{z})$ has a

distribution :

$$q(\mathbf{z}') = q(\mathbf{z}) \left| \det \frac{\partial f^{-1}}{\partial \mathbf{z}'} \right| = q(\mathbf{z}) \left| \det \frac{\partial f}{\partial \mathbf{z}} \right|^{-1}, \quad (5)$$

where the last equality can be seen by applying the chain rule (inverse function theorem) and is a property of Jacobians of invertible functions. We can construct arbitrarily complex densities by composing several simple maps and successively applying (5). The density $q_K(\mathbf{z})$ obtained by successively transforming a random variable $\mathbf{z}_0$ with distribution q_0 through a chain of K transformations f_k is:

$$\mathbf{z}_K = f_K \circ \ldots \circ f_2 \circ f_1(\mathbf{z}_0) \quad (6)$$

$$\ln q_K(\mathbf{z}_K) = \ln q_0(\mathbf{z}_0) - \sum_{k=1}^{K} \ln \left| \det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}} \right|, \quad (7)$$

where equation (6) will be used throughout the paper as a shorthand for the composition $f_K(f_{K-1}(\ldots f_1(x)))$. The path traversed by the random variables $\mathbf{z}_k = f_k(\mathbf{z}_{k-1})$ with initial distribution $q_0(\mathbf{z}_0)$ is called the *flow* and the path formed by the successive distributions q_k is a *normalizing flow*. A property of such transformations, often referred to as the law of the unconscious statistician (LOTUS), is that expectations w.r.t. the transformed density q_K can be computed without explicitly knowing q_K. Any expectation $\mathbb{E}_{q_K}[h(\mathbf{z})]$ can be written as an expectation under q_0 as:

$$\mathbb{E}_{q_K}[h(\mathbf{z})] = \mathbb{E}_{q_0}[h(f_K \circ f_{K-1} \circ \ldots \circ f_1(\mathbf{z}_0))], \quad (8)$$

which does not require computation of the the logdet-Jacobian terms when $h(\mathbf{z})$ does not depend on q_K.

We can understand the effect of invertible flows as a sequence of expansions or contractions on the initial density. For an expansion, the map $\mathbf{z}' = f(\mathbf{z})$ pulls the points $\mathbf{z}$ away from a region in $\mathbb{R}^d$, reducing the density in that region while increasing the density outside the region. Conversely, for a contraction, the map pushes points towards the interior of a region, increasing the density in its interior while reducing the density outside.

The formalism of normalizing flows now gives us a systematic way of specifying the approximate posterior distributions $q(\mathbf{z}|\mathbf{x})$ required for variational inference. With an appropriate choice of transformations f_K, we can initially use simple factorized distributions such as an independent Gaussian, and apply normalizing flows of different lengths to obtain increasingly complex and multi-modal distributions.

3.2. Infinitesimal Flows

It is natural to consider the case in which the length of the normalizing flow tends to infinity. In this case, we obtain an *infinitesimal flow*, that is described not in terms of a finite sequence of transformations — a finite flow, but as a

partial differential equation describing how the initial density $q_0(\mathbf{z})$ evolves over 'time': $\frac{\partial}{\partial t} q_t(\mathbf{z}) = \mathcal{T}_t[q_t(\mathbf{z})]$, where $\mathcal{T}$ describes the continuous-time dynamics.

Langevin Flow. One important family of flows is given by the Langevin stochastic differential equation (SDE):

$$d\mathbf{z}(t) = \mathbf{F}(\mathbf{z}(t), t)dt + \mathbf{G}(\mathbf{z}(t), t)d\boldsymbol{\xi}(t), \quad (9)$$

where $d\boldsymbol{\xi}(t)$ is a Wiener process with $\mathbb{E}[\boldsymbol{\xi}_i(t)] = 0$ and $\mathbb{E}[\boldsymbol{\xi}_i(t)\boldsymbol{\xi}_j(t')] = \delta_{i,j}\delta(t - t')$, $\mathbf{F}$ is the drift vector and $\mathbf{D} = \mathbf{G}\mathbf{G}^\top$ is the diffusion matrix. If we transform a random variable $\mathbf{z}$ with initial density $q_0(\mathbf{z})$ through the Langevin flow (9), then the rules for the transformation of densities is given by the Fokker-Planck equation (or Kolmogorov equations in probability theory). The density $q_t(\mathbf{z})$ of the transformed samples at time t will evolve as:

$$\frac{\partial}{\partial t} q_t(\mathbf{z}) = -\sum_i \frac{\partial}{\partial z_i}[F_i(\mathbf{z}, t)q_t] + \frac{1}{2}\sum_{i,j} \frac{\partial^2}{\partial z_i \partial z_j}[D_{ij}(\mathbf{z}, t)q_t].$$

In machine learning, we most often use the Langevin flow with $F(\mathbf{z}, t) = -\nabla_z \mathcal{L}(\mathbf{z})$ and $G(\mathbf{z}, t) = \sqrt{2}\delta_{ij}$, where $\mathcal{L}(\mathbf{z})$ is an unnormalised log-density of our model.

Importantly, in this case the stationary solution for $q_t(\mathbf{z})$ is given by the Boltzmann distribution: $q_\infty(\mathbf{z}) \propto e^{-\mathcal{L}(\mathbf{z})}$. That is, if we start from an initial density $q_0(\mathbf{z})$ and evolve its samples $\mathbf{z}_0$ through the Langevin SDE, the resulting points $\mathbf{z}_\infty$ will be distributed according to $q_\infty(\mathbf{z}) \propto e^{-\mathcal{L}(\mathbf{z})}$, i.e. the true posterior. This approach has been explored for sampling from complex densities by Welling & Teh (2011); Ahn et al. (2012); Suykens et al. (1998).

Hamiltonian Flow. Hamiltonian Monte Carlo can also be described in terms of a normalizing flow on an augmented space $\tilde{\mathbf{z}} = (\mathbf{z}, \boldsymbol{\omega})$ with dynamics resulting from the Hamiltonian $\mathcal{H}(\mathbf{z}, \boldsymbol{\omega}) = -\mathcal{L}(\mathbf{z}) - \frac{1}{2}\boldsymbol{\omega}^\top \mathbf{M}\boldsymbol{\omega}$; HMC is also widely used in machine learning, e.g., Neal (2011). We will use the Hamiltonian flow to make a connection to the recently introduced Hamiltonian variational approach from Salimans et al. (2015) in section 5.

4. Inference with Normalizing Flows

To allow for scalable inference using finite normalizing flows, we must specify a class of invertible transformations that can be used and an efficient mechanism for computing the determinant of the Jacobian. While it is straightforward to build invertible parametric functions for use in equation (5), e.g., invertible neural networks (Baird et al., 2005; Rippel & Adams, 2013), such approaches typically have a complexity for computing the Jacobian determinant that scales as $O(LD^3)$, where D is the dimension of the hidden layers and L is the number of hidden layers used. Furthermore, computing the gradients of the Jacobian determinant involves several additional operations that are also $O(LD^3)$

and involve matrix inverses that can be numerically unstable. We therefore require normalizing flows that allow for low-cost computation of the determinant, or where the Jacobian is not needed at all.

4.1. Invertible Linear-time Transformations

We consider a family of transformations of the form:

$$f(\mathbf{z}) = \mathbf{z} + \mathbf{u}h(\mathbf{w}^\top\mathbf{z} + b), \qquad (10)$$

where $\lambda = \{\mathbf{w} \in \mathbb{R}^D, \mathbf{u} \in \mathbb{R}^D, b \in \mathbb{R}\}$ are free parameters and $h(\cdot)$ is a smooth element-wise non-linearity, with derivative $h'(\cdot)$. For this mapping we can compute the logdet-Jacobian term in $O(D)$ time (using the matrix determinant lemma):

$$\psi(\mathbf{z}) = h'(\mathbf{w}^\top\mathbf{z} + b)\mathbf{w} \qquad (11)$$

$$\left|\det \frac{\partial f}{\partial \mathbf{z}}\right| = |\det(\mathbf{I} + \mathbf{u}\psi(\mathbf{z})^\top)| = |1 + \mathbf{u}^\top\psi(\mathbf{z})|. \qquad (12)$$

From (7) we conclude that the density $q_K(\mathbf{z})$ obtained by transforming an arbitrary initial density $q_0(\mathbf{z})$ through the sequence of maps f_k of the form (10) is implicitly given by:

$$\mathbf{z}_K = f_K \circ f_{K-1} \circ \ldots \circ f_1(\mathbf{z})$$

$$\ln q_K(\mathbf{z}_K) = \ln q_0(\mathbf{z}) - \sum_{k=1}^{K} \ln|1 + \mathbf{u}_k^\top\psi_k(\mathbf{z}_{k-1})|. \qquad (13)$$

The flow defined by the transformation (13) modifies the initial density q_0 by applying a series of contractions and expansions in the direction perpendicular to the hyperplane $\mathbf{w}^\top\mathbf{z} + b = 0$, hence we refer to these maps as planar flows.

As an alternative, we can consider a family of transformations that modify an initial density q_0 around a reference point $\mathbf{z}_0$. The transformation family is:

$$f(\mathbf{z}) = \mathbf{z} + \beta h(\alpha, r)(\mathbf{z} - \mathbf{z}_0), \qquad (14)$$

$$\left|\det \frac{\partial f}{\partial \mathbf{z}}\right| = [1 + \beta h(\alpha, r)]^{d-1}[1 + \beta h(\alpha, r) + \beta h'(\alpha, r)r)],$$

where $r = |\mathbf{z} - \mathbf{z}_0|$, $h(\alpha, r) = 1/(\alpha + r)$, and the parameters of the map are $\lambda = \{\mathbf{z}_0 \in \mathbb{R}^D, \alpha \in \mathbb{R}^+, \beta \in \mathbb{R}\}$. This family also allows for linear-time computation of the determinant. It applies radial contractions and expansions around the reference point and are thus referred to as radial flows. We show the effect of expansions and contractions on a uniform and Gaussian initial density using the flows (10) and (14) in figure 1. This visualization shows that we can transform a spherical Gaussian distribution into a bimodal distribution by applying two successive transformations.

Not all functions of the form (10) or (14) will be invertible. We discuss the conditions for invertibility and how to satisfy them in a numerically stable way in the appendix.
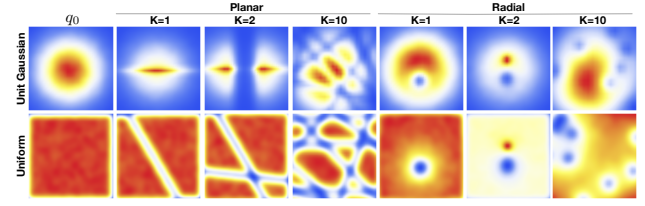

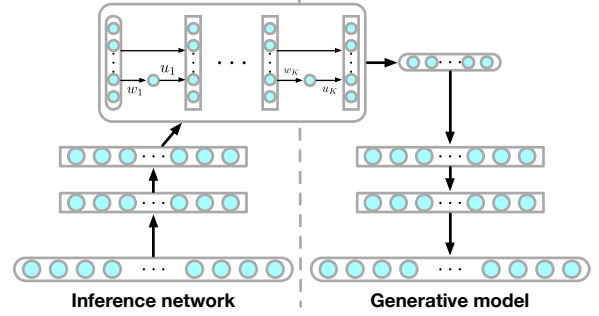*Figure 1.* Effect of normalizing flow on two distributions.

Figure 2. Inference and generative models. Left: Inference network maps the observations to the parameters of the flow; Right: generative model which receives the posterior samples from the inference network during training time. Round containers represent layers of stochastic variables whereas square containers represent deterministic layers.

4.2. Flow-Based Free Energy Bound

If we parameterize the approximate posterior distribution with a flow of length K, $q_\phi(\mathbf{z}|\mathbf{x}) := q_K(\mathbf{z}_K)$, the free energy (3) can be written as an expectation over the initial distribution $q_0(\mathbf{z})$:

$$\mathcal{F}(\mathbf{x}) = \mathbb{E}_{q_\phi(z|x)}[\log q_\phi(\mathbf{z}|\mathbf{x}) - \log p(\mathbf{x}, \mathbf{z})]$$

$$= \mathbb{E}_{q_0(z_0)}[\ln q_K(\mathbf{z}_K) - \log p(\mathbf{x}, \mathbf{z}_K)]$$

$$= \mathbb{E}_{q_0(z_0)}[\ln q_0(\mathbf{z}_0)] - \mathbb{E}_{q_0(z_0)}[\log p(\mathbf{x}, \mathbf{z}_K)]$$

$$- \mathbb{E}_{q_0(z_0)}\left[\sum_{k=1}^{K} \ln|1 + \mathbf{u}_k^\top\psi_k(\mathbf{z}_{k-1})|\right]. \qquad (15)$$

Normalizing flows and this free energy bound can be used with any variational optimization scheme, including generalized variational EM. For amortized variational inference, we construct an inference model using a deep neural network to build a mapping from the observations $\mathbf{x}$ to the parameters of the initial density $q_0 = \mathcal{N}(\mu, \sigma)$ ($\mu \in \mathbb{R}^D$ and $\sigma \in \mathbb{R}^D$) as well as the parameters of the flow λ.

4.3. Algorithm Summary and Complexity

The resulting algorithm is a simple modification of the amortized inference algorithm for DLGMs described by (Kingma & Welling, 2014; Rezende et al., 2014), which we summarize in algorithm 1. By using an inference net-

---

Algorithm 1 Variational Inf. with Normalizing Flows

Parameters: ϕ variational, θ generative
while not converged **do**
    $\mathbf{x} \leftarrow \{\text{Get mini-batch}\}$
    $\mathbf{z}_0 \sim q_0(\bullet|\mathbf{x})$
    $\mathbf{z}_K \leftarrow f_K \circ f_{K-1} \circ \ldots \circ f_1(\mathbf{z}_0)$
    $\mathcal{F}(\mathbf{x}) \approx \mathcal{F}(\mathbf{x}, \mathbf{z}_K)$
    $\Delta\boldsymbol{\theta} \propto -\nabla_\theta \mathcal{F}(\mathbf{x})$
    $\Delta\boldsymbol{\phi} \propto -\nabla_\phi \mathcal{F}(\mathbf{x})$
end while

---

work we are able to form a single computational graph which allows for easy computation of all the gradients of the parameters of the inference network and the generative model. The estimated gradients are used in conjunction with preconditioned stochastic gradient-based optimization methods such as RMSprop or AdaGrad (Duchi et al., 2010), where we use parameter updates of the form: $(\boldsymbol{\theta}^{t+1}, \boldsymbol{\phi}^{t+1}) \leftarrow (\boldsymbol{\theta}^t, \boldsymbol{\phi}^t) + \boldsymbol{\Gamma}^t(\mathbf{g}_\theta^t, \mathbf{g}_\phi^t)$, with $\boldsymbol{\Gamma}$ is a diagonal preconditioning matrix that adaptively scales the gradients for faster minimization.

The algorithmic complexity of jointly sampling and computing the log-det-Jacobian terms of the inference model scales as $O(LN^2) + O(KD)$, where L is the number of deterministic layers used to map the data to the parameters of the flow, N is the average hidden layer size, K is the flow-length and D is the dimension of the latent variables. Thus the overall algorithm is at most quadratic making the overall approach competitive with other large-scale systems used in practice.

5. Alternative Flow-based Posteriors

Using the framework of normalizing flows, we can provide a unified view of recent proposals for designing more flexible posterior approximations. At the outset, we distinguish between two types of flow mechanisms that differ in how the Jacobian is handled. The work in this paper considers *general normalizing flows* and presents a method for linear-time computation of the Jacobian. In contrast, *volume-preserving flows* design the flow such that its Jacobian-determinant is equal to one while still allowing for rich posterior distributions. Both these categories allow for flows that may be finite or infinitesimal.

The Non-linear Independent Components Estimation (NICE) developed by Dinh et al. (2014) is an instance of a *finite volume-preserving flow*. The transformations used are neural networks $f(\cdot)$ with easy to compute inverses $g(\cdot)$ of the form:

$$f(\mathbf{z}) = (\mathbf{z}_A, \mathbf{z}_B + h_\lambda(\mathbf{z}_A)), \qquad (16)$$
$$g(\mathbf{z}') = (\mathbf{z}'_A, \mathbf{z}'_B - h_\lambda(\mathbf{z}'_A)). \qquad (17)$$

where $\mathbf{z} = (\mathbf{z}_A, \mathbf{z}_B)$ is an arbitrary partitioning of the vec-

tor $\mathbf{z}$ and h_λ is a neural network with parameters λ. This form results in a Jacobian that has a zero upper triangular part, resulting in a determinant of 1. In order to build a transformation capable of mixing all components of the initial random variable $\mathbf{z}_0$, such flows must alternate between different partitionings of $\mathbf{z}_k$. The resulting density using the forward and inverse transformations is given by:

$$\ln q_K(f_K \circ f_{K-1} \circ \ldots \circ f_1(\mathbf{z}_0)) = \ln q_0(\mathbf{z}_0), \quad (18)$$
$$\ln q_K(\mathbf{z}') = q_0(g_1 \circ g_2 \circ \ldots \circ g_K(\mathbf{z}')). \qquad (19)$$

We will compare NICE to the general transformation approach described in section 2.1. Dinh et al. (2014) assume the partitioning is of the form $\mathbf{z} = [\mathbf{z}_A = \mathbf{z}_{1:d}, \mathbf{z}_B = \mathbf{z}_{d+1:D}]$. To enhance mixing of the components in the flow, we introduce two mechanisms for mixing the components of $\mathbf{z}$ before separating them in the disjoint subgroups $\mathbf{z}_A$ and $\mathbf{z}_B$. The first mechanism applies a random permutation (NICE-perm) and the second applies a random orthogonal transformation (NICE-orth)[1].

The Hamiltonian variational approximation (HVI) developed by Salimans et al. (2015) is an instance of an *infinitesimal volume-preserving flow*. For HVI, we consider posterior approximations $q(\mathbf{z}, \boldsymbol{\omega}|\mathbf{x})$ that make use of additional auxiliary variables $\boldsymbol{\omega}$. The latent variables $\mathbf{z}$ are independent of the auxiliary variables $\boldsymbol{\omega}$ and using the change of variables rule, the resulting distribution is: $q(\mathbf{z}', \boldsymbol{\omega}') = |\mathbf{J}|q(\mathbf{z})q(\boldsymbol{\omega})$, where $\mathbf{z}', \boldsymbol{\omega}' = f(\mathbf{z}, \boldsymbol{\omega})$ using a transformation f. Salimans et al. (2015) obtain a volume-preserving invertible transformation by exploiting the use of such transition operators in the MCMC literature, in particular the methods of Langevin and Hybrid Monte Carlo. This is an extremely elegant approach, since we now know that as the number of iterations of the transition function tends to infinity, the distribution $q(\mathbf{z}')$ will tend to the true distribution $p(\mathbf{z}|\mathbf{x})$. This is an alternative way to make use of the Hamiltonian infinitesimal flow described in section 3.2. A disadvantage of using the Langevin or Hamiltonian flow is that they require one or more evaluations of the likelihood and its gradients (depending in the number of leapfrog steps) per iteration during both training and test time.

6. Results

Throughout this section we evaluate the effect of using normalizing flow-based posterior approximations for inference in deep latent Gaussian models (DLGMs). Training was performed by following a Monte Carlo estimate of the gradient of an annealed version of the free energy (20), with respect the model parameters $\boldsymbol{\theta}$ and the variational parameters $\boldsymbol{\phi}$ using stochastic backpropoagation. The Monte

---

[1] Random orthogonal transformations can be generated by sampling a matrix with independent unit-Gaussian entries $A_{i,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and then performing a QR-factorization. The resulting Q-matrix will be a random orthogonal matrix (Genz, 1998).

Table 1. Test energy functions.

Potential $U(\mathbf{z})$
1: $\frac{1}{2}\left(\frac{\|\mathbf{z}\|-2}{0.4}\right)^2 - \ln\left(e^{-\frac{1}{2}\left[\frac{\mathbf{Z}_1-2}{0.6}\right]^2} + e^{-\frac{1}{2}\left[\frac{\mathbf{Z}_1+2}{0.6}\right]^2}\right)$
2: $\frac{1}{2}\left[\frac{\mathbf{Z}_2 - w_1(\mathbf{z})}{0.4}\right]^2$
3: $-\ln\left(e^{-\frac{1}{2}\left[\frac{\mathbf{Z}_2-w_1(\mathbf{Z})}{0.35}\right]^2} + e^{-\frac{1}{2}\left[\frac{\mathbf{Z}_2-w_1(\mathbf{Z})+w_2(\mathbf{Z})}{0.35}\right]^2}\right)$
4: $-\ln\left(e^{-\frac{1}{2}\left[\frac{\mathbf{Z}_2-w_1(\mathbf{Z})}{0.4}\right]^2} + e^{-\frac{1}{2}\left[\frac{\mathbf{Z}_2-w_1(\mathbf{Z})+w_3(\mathbf{Z})}{0.35}\right]^2}\right)$
with $w_1(\mathbf{z}) = \sin\left(\frac{2\pi\mathbf{Z}_1}{4}\right)$, $w_2(\mathbf{z}) = 3e^{-\frac{1}{2}\left[\frac{\mathbf{Z}_1-1}{0.6}\right]^2}$, $w_3(\mathbf{z}) = 3\sigma\left(\frac{\mathbf{Z}_1-1}{0.3}\right)$ and $\sigma(x) = 1/(1+e^{-x})$.

Carlo estimate is computed using a single sample of the latent variables per data-point per parameter update.

A simple annealed version of the free energy is used since this was found to provide better results. The modified bound is:

$$\mathbf{z}_K = f_K \circ f_{K-1} \circ \ldots \circ f_1(\mathbf{z})$$
$$\mathcal{F}^{\beta_t}(\mathbf{x}) = \mathbb{E}_{q^0(\mathbf{z}_0)}\left[\ln p^K(\mathbf{z}_K) - \log p(\mathbf{x}, \mathbf{z}_K)\right]$$
$$= \mathbb{E}_{q_0(\mathbf{z}_0)}\left[\ln q_0(\mathbf{z}_0)\right] - \beta_t \mathbb{E}_{q_0(\mathbf{z}_0)}\left[\log p(\mathbf{x}, \mathbf{z}_K)\right]$$
$$- \mathbb{E}_{q_0(\mathbf{z}_0)}\left[\sum_{k=1}^{K} \ln|1 + u_k^T \psi_k(\mathbf{z}_{k-1})|\right] \quad (20)$$

where $\beta_t \in [0,1]$ is an inverse temperature that follows a schedule $\beta_t = \min(1, 0.01 + t/10000)$, going from 0.01 to 1 after 10000 iterations.
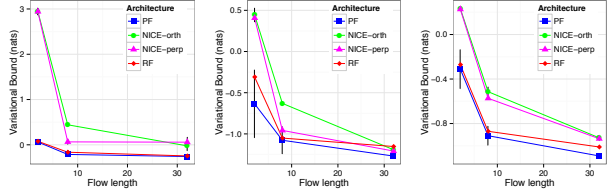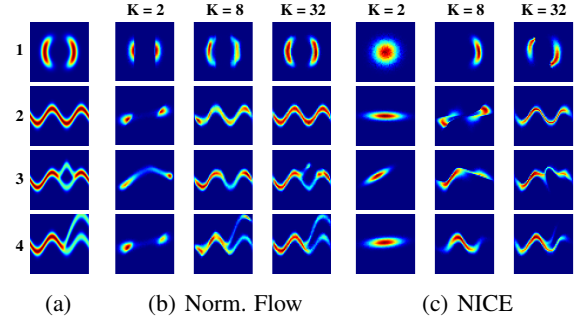
The deep neural networks that form the conditional probability between random variables consist of deterministic layers with 400 hidden units using the Maxout non-linearity on windows of 4 variables (Goodfellow et al., 2013). Briefly, the Maxout non-linearity with window-size Δ takes an input vector $\mathbf{x} \in \mathbb{R}^d$ and computes: $\text{Maxout}(\mathbf{x})_k = \max_{i \in \{\Delta k, \Delta(k+1)\}} \mathbf{x}_i$ for $k = 0 \ldots d/\Delta$.

We use mini-batches of 100 data points and RMSprop optimization (with learning rate $= 1 \times 10^{-5}$ and momentum $= 0.9$) (Kingma & Welling, 2014; Rezende et al., 2014). Results were collected after $500,000$ parameter updates. Each experiment was repeated 100 times with different random seeds and we report the averaged scores and standard errors. The true marginal likelihood is estimated by importance sampling using 200 samples from the inference network as in (Rezende et al., 2014, App. E).

6.1. Representative Power of Normalizing Flows

To provide an insight into the representative power of density approximations based on normalizing flows, we parameterize a set of unnormalized 2D densities $p(\mathbf{z}) \propto \exp[-U(\mathbf{z})]$ which are listed in table 1.

In figure 3(a) we show the true distribution for four cases,

(a) (b) Norm. Flow (c) NICE

(d) Comparison of KL-divergences.

Figure 3. Approximating four non-Gaussian 2D distributions. The images represent densities for each energy function in table 1 in the range $(-4, 4)^2$. (a) True posterior; (b) Approx posterior using the normalizing flow (13); (c) Approx posterior using NICE (19); (d) Summary results comparing KL-divergences between the true and approximated densities for the first 3 cases.

which show distributions that have characteristics such as multi-modality and periodicity that cannot be captured with typically-used posterior approximations.

Figure 3(b) shows the performance of normalizing flow approximations for these densities using flow lengths of 2, 8 and 32 transformations. The non-linearity $h(\mathbf{z}) = \tanh(\mathbf{z})$ in equation (10) was used for the mapping and the initial distribution was a diagonal Gaussian, $q_0(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \sigma^2\mathbf{I})$. We see a substantial improvement in the approximation quality as we increase the flow length. Figure 3(c) shows the same approximation using the volume-preserving transformation used in NICE (Dinh et al., 2014) for the same number of transformations. We show summary statistics for the planar flow (13), and NICE (18) for random orthogonal matrices and with random permutation matrices in 3(d). We found that NICE and the planar flow (13) may achieve the same asymptotic performance as we grow the flow-length, but the planar flow (13) requires far fewer parameters. Presumably because all parameters of the flow (13) are learned, in contrast to NICE which requires an extra mechanism for mixing the components that is not learned but randomly initialized. We did not observe a substantial difference between using random orthogonal matrices or random permutation matrices in NICE.

6.2. MNIST and CIFAR-10 Images

The MNIST digit dataset (LeCun & Cortes, 1998) contains 60,000 training and 10,000 test images of ten handwritten
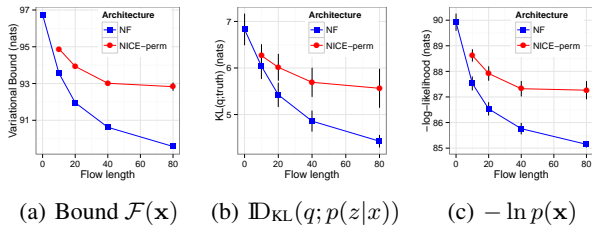
Figure 4. Effect of the flow-length on MNIST.

Table 2. Comparison of negative log-probabilities on the test set for the binarised MNIST data.

Model	$-\ln p(\mathbf{x})$
DLGM diagonal covariance	≤ 89.9
DLGM+NF (k = 10)	≤ 87.5
DLGM+NF (k = 20)	≤ 86.5
DLGM+NF (k = 40)	≤ 85.7
DLGM+NF (k = 80)	≤ 85.1
DLGM+NICE (k = 10)	≤ 88.6
DLGM+NICE (k = 20)	≤ 87.9
DLGM+NICE (k = 40)	≤ 87.3
DLGM+NICE (k = 80)	≤ 87.2
Results below from (Salimans et al., 2015)	
DLGM + HVI (1 leapfrog step)	88.08
DLGM + HVI (4 leapfrog steps)	86.40
DLGM + HVI (8 leapfrog steps)	85.51
Results below from (Gregor et al., 2014)	
DARN $n_h = 500$	84.71
DARN $n_h = 500$, adaNoise	84.13

digits (0 to 9) that are 28×28 pixels in size. We used the binarized dataset as in (Uria et al., 2014). We trained different DLGMs with 40 latent variables for $500,000$ parameter updates.

The performance of a DLGM using the (planar) normalizing flow (DLGM+NF) approximation is compared to the volume-preserving approaches using NICE (DLGM+NICE) on exactly the same model for different flow-lengths K, and we summarize the performance in figure 4. This graph shows that an increase in the flow-length systematically improves the bound $\mathcal{F}$, as shown in figure 4(a), and reduces the KL-divergence between the approximate posterior $q(\mathbf{z}|\mathbf{x})$ and the true posterior distribution $p(\mathbf{z}|\mathbf{x})$ (figure 4(b)). It also shows that the approach using general normalizing flows outperforms that of NICE. We also show a wider comparison in table 2. Results are included for the Hamiltonian variational approach as well, but the model specification is different and thus gives an indication of attainable performance for this approach on this data set.

The CIFAR-10 natural images dataset (Krizhevsky & Hinton, 2010) consists of 50,000 training and 10,000 test RGB images that are of size 3x32x32 pixels from which we extract 3x8x8 random patches. The color levels were converted to the range $[\epsilon, 1 - \epsilon]$ with $\epsilon = 0.0001$. Here we used similar DLGMs as used for the MNIST experiment,

Table 3. Test set performance on the CIFAR-10 data.

	$K=0$	$K=2$	$K=5$	$K=10$
$-\ln p(\mathbf{x})$	-293.7	-308.6	-317.9	-320.7

but with 30 latent variables. Since this data is non-binary, we use a logit-normal observation likelihood, $p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\alpha}) = \prod_i \frac{\mathcal{N}(\text{logit}(\mathbf{X}_i)|\boldsymbol{\mu}_i, \boldsymbol{\alpha}_i)}{\mathbf{X}_i(1-\mathbf{X}_i)}$, where $\text{logit}(x) = \log \frac{x}{1-x}$. We summarize the results in table 3 where we are again able to show that an increase in the flow length K systematically improves the test log-likelihoods, resulting in better posterior approximations.

7. Conclusion and Discussion

In this work we developed a simple approach for learning highly non-Gaussian posterior densities by learning transformations of simple densities to more complex ones through a normalizing flow. When combined with an amortized approach for variational inference using inference networks and efficient Monte Carlo gradient estimation, we are able to show clear improvements over simple approximations on different problems. Using this view of normalizing flows, we are able to provide a unified perspective of other closely related methods for flexible posterior estimation that points to a wide spectrum of approaches for designing more powerful posterior approximations with different statistical and computational tradeoffs.

An important conclusion from the discussion in section 3 is that there exist classes of normalizing flows that allow us to create extremely rich posterior approximations for variational inference. With normalizing flows, we are able to show that in the asymptotic regime, the space of solutions is rich enough to contain the true posterior distribution. If we combine this with the local convergence and consistency results for maximum likelihood parameter estimation in certain classes of latent variables models (Wang & Titterington, 2004), we see that we are now able overcome the objections to using variational inference as a competitive and default approach for statistical inference. Making such statements rigorous is an important line of future research.

Normalizing flows allow us to control the complexity of the posterior at run-time by simply increasing the flow length of the sequence. The approach we presented considered normalizing flows based on simple transformations of the form (10) and (14). These are just two of the many maps that can be used, and alternative transforms can be designed for posterior approximations that may require other constraints, e.g., a restricted support. An important avenue of future research lies in describing the classes of transformations that allow for different characteristics of the posterior and that still allow for efficient, linear-time computation.

References

Ahn, S., Korattikara, A., and Welling, M. Bayesian posterior sampling via stochastic gradient Fisher scoring. In *ICML*, 2012.

Baird, L., Smalenberger, D., and Ingkiriwang, S. One-step neural network inversion with PDF learning and emulation. In *IJCNN*, volume 2, pp. 966–971. IEEE, 2005.

Bishop, C. M. *Pattern recognition and machine learning*. springer New York, 2006.

Challis, E. and Barber, D. Affine independent variational inference. In *NIPS*, 2012.

Dayan, P. Helmholtz machines and wake-sleep learning. *Handbook of Brain Theory and Neural Network. MIT Press, Cambridge, MA*, 44(0), 2000.

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

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159, 2010.

Genz, A. Methods for generating random orthogonal matrices. *Monte Carlo and Quasi-Monte Carlo Methods*, 1998.

Gershman, S., Hoffman, M., and Blei, D. Nonparametric variational inference. In *ICML*, 2012.

Gershman, S. J. and Goodman, N. D. Amortized inference in probabilistic reasoning. In *Annual Conference of the Cognitive Science Society*, 2014.

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. Maxout networks. *ICML*, 2013.

Gregor, K., Danihelka, I., Mnih, A., Blundell, C., and Wierstra, D. Deep autoregressive networks. In *ICML*, 2014.

Gregor, Karol, Danihelka, Ivo, Graves, Alex, Jimenez Rezende, Danilo, and Wierstra, Daan. Draw: A recurrent neural network for image generation. In *ICML*, 2015.

Hoffman, M. D., Blei, D. M, Wang, C., and Paisley, J. Stochastic variational inference. *JMLR*, 14(1):1303–1347, 2013.

Jaakkola, T. S. and Jordan, M. I. Improving the mean field approximation via the use of mixture distributions. In *Learning in graphical models*, pp. 163–173. 1998.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

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 *NIPS*, pp. 3581–3589, 2014.

Krizhevsky, A. and Hinton, G. Convolutional deep belief networks on CIFAR-10. *Unpublished manuscript*, 2010.

LeCun, Y. and Cortes, C. The MNIST database of handwritten digits, 1998.

Mnih, A. and Gregor, K. Neural variational inference and learning in belief networks. In *ICML*, 2014.

Neal, R. M. MCMC using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2011.

Papaspiliopoulos, O., Roberts, G. O., and Sköld, M. Non-centered parameterisations for hierarchical models and data augmentation. In *Bayesian Statistics 7*, 2003.

Ranganath, R., Gerrish, S., and Blei, D. M. Black box variational inference. In *AISTATS*, 2013.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

Rippel, O. and Adams, R. P. High-dimensional probability estimation with deep density models. *arXiv:1302.5125*, 2013.

Salimans, T., Kingma, D. P., and Welling, M. Markov chain Monte Carlo and variational inference: Bridging the gap. In *ICML*, 2015.

Suykens, J. A. K., Verrelst, H., and Vandewalle, J. Online learning Fokker-Planck machine. *Neural processing letters*, 7(2):81–89, 1998.

Tabak, E. G. and Turner, C. V. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

Tabak, E. G and Vanden-Eijnden, E. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.

Titsias, M. and Lazaro-Gredilla, M. Doubly stochastic variational Bayes for non-conjugate inference. In *ICML*, 2014.

Turner, R. E. and Sahani, M. Two problems with variational expectation maximisation for time-series models. In Barber, D., Cemgil, T., and Chiappa, S. (eds.), *Bayesian Time series models*, chapter 5, pp. 109–130. Cambridge University Press, 2011.

Uria, B., Murray, I., and Larochelle, H. A deep and tractable density estimator. In *ICML*, 2014.

Wang, B. and Titterington, D. M. Convergence and asymptotic normality of variational Bayesian approximations for exponential family models with missing values. In *UAI*, 2004.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient Langevin dynamics. In *ICML*, 2011.

Williams, Ronald J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Wingate, D. and Weber, T. Automated variational inference in probabilistic programming. In *NIPS Workshop on Probabilistic Programming*, 2013.

A. Invertibility conditions

We describe the constraints required to have invertible maps for the planar and radial normalizing flows described in section 3.

A.1. Planar flows

Functions of the form (10) are not always invertible depending on the non-linearity and parameters chosen. When using $h(x) = \tanh(x)$, a sufficient condition for $f(\mathbf{z})$ to be invertible is that $\mathbf{w}^\top \mathbf{u} \geq -1$.

This can be seen by splitting $\mathbf{z}$ as a sum of a vector $\mathbf{z}_\perp$ perpendicular to $\mathbf{w}$ and a vector $\mathbf{z}_\parallel$, parallel to $\mathbf{w}$. Substituting $\mathbf{z} = \mathbf{z}_\perp + \mathbf{z}_\parallel$ into (10) gives

$$f(\mathbf{z}) = \mathbf{z}_\perp + \mathbf{z}_\parallel + \mathbf{u}h(\mathbf{w}^\top \mathbf{z}_\parallel + b). \tag{21}$$

This equation can be solved for $\mathbf{z}_\perp$ given $\mathbf{z}_\parallel$ and $\mathbf{y} = f(\mathbf{z})$, having a unique solution

$$\mathbf{z}_\perp = y - \mathbf{z}_\parallel - \mathbf{u}h(\mathbf{w}^\top \mathbf{z}_\parallel + b). \tag{22}$$

The parallel component can be further expanded as $\mathbf{z}_\parallel = \alpha \frac{\mathbf{w}}{||\mathbf{w}||^2}$, where $\alpha \in \mathbb{R}$. The equation that must be solved for α is derived by taking the dot product of (21) with $\mathbf{w}$, yielding the scalar equation

$$\mathbf{w}^T f(\mathbf{z}) = \alpha + \mathbf{w}^T \mathbf{u}h(\alpha + b). \tag{23}$$

A sufficient condition for (23) to be invertible w.r.t α is that its r.h.s $\alpha + \mathbf{w}^T \mathbf{u}h(\alpha + b)$ to be a non-decreasing function. This corresponds to the condition $1 + \mathbf{w}^T \mathbf{u}h'(\alpha + b) \geq 0 \equiv \mathbf{w}^T \mathbf{u} \geq -\frac{1}{h'(\alpha + b)}$. Since $0 \leq h'(\alpha + b) \leq 1$, it suffices to have $\mathbf{w}^T \mathbf{u} \geq -1$.

We enforce this constraint by taking an arbitrary vector $\mathbf{u}$ and modifying its component parallel to $\mathbf{w}$, producing a new vector $\hat{\mathbf{u}}$ such that $\mathbf{w}^\top \hat{\mathbf{u}} > -1$. The modified vector can be compactly written as $\hat{\mathbf{u}}(\mathbf{w}, \mathbf{u}) = \mathbf{u} + \left[m(\mathbf{w}^\top \mathbf{u}) - (\mathbf{w}^\top \mathbf{u}) \right] \frac{\mathbf{w}}{||\mathbf{w}||^2}$, where the scalar function $m(x)$ is given by $m(x) = -1 + \log(1 + e^x)$.

A.2. Radial flows

Functions of the form (14) are not always invertible depending on the values of α and β. This can be seen by splitting the vector $\mathbf{z}$ as $\mathbf{z} = \mathbf{z}_0 + r\hat{\mathbf{z}}$, where $r = |\mathbf{z} - \mathbf{z}_0|$. Replacing this into (14) gives

$$f(\mathbf{z}) = \mathbf{z}_0 + r\hat{\mathbf{z}} + \beta \frac{r\hat{\mathbf{z}}}{\alpha + r}. \tag{24}$$

This equation can be uniquely solved for $\hat{\mathbf{z}}$ given r and $\mathbf{y} = f(\mathbf{z})$,

$$\hat{\mathbf{z}} = \frac{\mathbf{y} - \mathbf{z}_0}{r \left(1 + \frac{\beta}{\alpha + r} \right)}. \tag{25}$$

To obtain a scalar equation for the norm r, we can subtract both sides of (24) and take the norm of both sides. This gives

$$|y - \mathbf{z}_0| = r \left(1 + \frac{\beta}{\alpha + r} \right). \tag{26}$$

A sufficient condition for (26) to be invertible is for its r.h.s. $r \left(1 + \frac{\beta}{\alpha + r} \right)$ to be a non-decreasing function, which implies $\beta \geq -\frac{(r + \alpha)^2}{\alpha}$. Since $r \geq 0$, it suffices to impose $\beta \geq -\alpha$. This constraint is imposed by reparametrizing β as $\hat{\beta} = -\alpha + m(\beta)$, where $m(x) = \log(1 + e^x)$.